

**UNIVERSIDAD NACIONAL DE SAN CRISTÓBAL DE
HUAMANGA**

FACULTAD DE INGENIERÍA DE MINAS, GEOLOGÍA Y CIVIL

ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS



**Sistema de información de recaudación tributaria de la
Municipalidad Distrital de Santillana bajo el marco Jira Scrum
Agile Software, Huanta, 2019**

Tesis para optar el título profesional de:

INGENIERO DE SISTEMAS

Presentado por:

Bach. Yonny HUARANCCA ÑAUPARI

Asesor:

Msc. Ing. Hubner JANAMPA PATILLA

Ayacucho - Perú

2021

DEDICATORIA

Mi tesis lo dedico con todo mi amor y cariño a mi amado madre María Ñaupari Araujo, padre Jesús Huarancca Curo y a mis hermanos Simona, Jhosep L. y Yamir Huarancca Ñaupari, por darme una carrera para nuestro futuro y por creer en mi capacidad, aunque hemos pasado momentos difíciles siempre ha estado brindándome su compañía, comprensión, aprecio, cariño y amor.

A mi abuela Teresa Araujo Condori y mi tía Hilda Ñaupari Araujo, por ser mi fuente de motivación e inspiración para poder superarme cada día más y así poder luchar para que la vida me prepare un futuro mejor.

AGRADECIMIENTO

A la primera instancia agradezco a mis formadores por tener la gran vocación de transmitir sus conocimientos. A la Escuela de Formación Profesional de Ingeniería de Sistemas y a la Universidad Nacional de san Cristóbal de Huamanga: por haber sido mi segunda casa y lugar de grandes conocimientos.

Gracias a la vida por esta oportunidad nueva de triunfo, gracias a todas las personas que me apoyaron y creyeron en la realización de esta tesis.

A Dios, por permitirme vivir y seguir el camino correcto, con objetivos claros, luchando por obtener lo que más quiero y por darme la oportunidad de que mi familia se sienta orgulloso de mis triunfos.

CONTENIDO

DEDICATORIA	i
AGRADECIMIENTO	ii
CONTENIDO	iii
RESUMEN	vi
INTRODUCCIÓN	i
CAPÍTULO I	1
PLANTEAMIENTO DE LA INVESTIGACIÓN	1
1.1. DIAGNÓSTICO Y ENUNCIADO DEL PROBLEMA	1
1.2. FORMULACIÓN DEL PROBLEMA DE INVESTIGACIÓN	4
PROBLEMA PRINCIPAL.....	4
PROBLEMAS SECUNDARIOS.....	4
CAPÍTULO II	6
OBJETIVOS DE LA INVESTIGACIÓN	6
2.1. OBJETIVO GENERAL.....	6
2.2. OBJETIVOS ESPECÍFICO	6
CAPÍTULO III.....	7
JUSTIFICACIÓN Y DELIMITACIÓN DE LA INVESTIGACIÓN	7
3.1. IMPORTANCIA DEL TEMA.....	7
IMPORTANCIA TÉCNICA.....	7
IMPORTANCIA ECONÓMICA.....	7
3.2. JUSTIFICACIÓN	7
3.3. DELIMITACIÓN	8
CAPÍTULO IV.....	9
REVISIÓN DE LA LITERATURA	9
4.1. ANTECEDENTES DE LA INVESTIGACIÓN	9
4.2. MARCO TEÓRICO.....	11
4.2.1. SISTEMA DE INFORMACIÓN DE RECAUDACIÓN TRIBUTARIA	11
A. IMPUESTO PREDIO URBANO	15
B. IMPUESTO PREDIO RURAL	15
4.2.2. JIRA SCRUM AGILE SOFTWARE.....	15
A. GESTIÓN REGULAR DE LAS EXPECTATIVAS DEL CLIENTE	16
B. EQUIPOS AUTO DIRIGIDOS Y AUTO ORGANIZADOS.....	16
4.3. MÉTODOS Y TÉCNICAS	16

4.3.1.	SOFTWARE JIRA	16
4.3.2.	MARCO DE TRABAJO SCRUM	22
4.4.	HERRAMIENTAS	40
4.4.1.	APLICACIÓN DE ESCRITORIO	40
4.4.2.	LENGUAJE DE PROGRAMACIÓN JAVA	41
4.4.3.	PROGRAMACIÓN ORIENTADA A OBJETOS (POO).....	42
4.4.4.	SISTEMA GESTOR DE BASE DE DATOS MYSQL	43
4.4.5.	PATRÓN DE ARQUITECTURA DE SOFTWARE MVC	44
4.4.6.	JAVA FX.....	46
4.4.7.	SPRING BOOT.....	46
4.4.8.	MAVEN	47
4.5.	POBLACIÓN	47
4.6.	MUESTRA	48
CAPÍTULO V		49
HIPÓTESIS Y VARIABLES DE INVESTIGACIÓN		49
5.1.	HIPÓTESIS.....	49
5.2.	DEFINICIÓN CONCEPTUAL DE LAS VARIABLES	49
5.3.	DEFINICIÓN OPERACIONAL DE LAS VARIABLES DE ESTUDIO	51
CAPÍTULO VI.....		52
METODOLOGÍA DE LA INVESTIGACIÓN		52
6.1.	TIPO Y NIVEL DE INVESTIGACIÓN.....	52
6.2.	DISEÑO DE LA INVESTIGACIÓN	52
6.3.	POBLACIÓN Y MUESTRA	53
6.4.	TÉCNICAS E INSTRUMENTOS PARA RECOLECTAR INFORMACIÓN....	53
6.5.	HERRAMIENTAS PARA EL DESARROLLO DE SISTEMA INFORMÁTICO54	
6.6.	TÉCNICAS PARA APLICAR JIRA SCRUM.....	56
CAPÍTULO VII		58
ANÁLISIS Y RESULTADOS DE LA INVESTIGACIÓN.....		58
7.1.	FASE Nº 1: PILA DE PRODUCTO (PRODUCT BACKLOG)	58
7.1.1.	DESCRIPCIÓN DEL PROYECTO.....	59
7.1.2.	CARTERA DE PRODUCTOS	65
7.1.2.1.	TABLA AGÍL SCRUM	65
7.1.2.2.	PANEL DE EPICS	66
7.1.2.3.	VERSIONES DE BACKLOG	67

7.1.3.	LISTADO DE LA PILA DE PRODUCTOS	69
7.1.4.	RELACIÓN DE HISTORIAL DE USUARIO ENTRE EPICS Y VERSIONES..	70
7.2.	FASE Nº 2: PLANIFICACIÓN DE SPRINT (SPRINT PLANNING).....	71
7.2.1.	RESERVA DE SPRINT (SPRINT BACKLOG)	72
7.2.2.	REUNIONES DE PLANIFICACIÓN DE SPRINT.....	99
7.3.	FASE Nº 3: SCRUM DIARIO (DAILY SCRUM)	111
A.	COMUNICACIÓN DE SPRINT BACKLOG	111
B.	GRÁFICO DE TRABAJO PENDIENTE (BURNDOWN CHARTS).....	111
7.4.	FASE Nº 4: REVISIÓN DE SPRINT (SPRINT REVIEW).....	122
A.	PLANIFICACIÓN DE ENTREGA	122
7.5.	FASE Nº 5: RETROSPECTIVA DE SPRINT (SPRINT RETROSPECTIVE).	141
7.6.	RESULTADOS	146
7.7.	ANÁLISIS DE RESULTADOS	150
CAPÍTULO VIII.....		151
CONCLUSIONES Y RECOMENDACIONES		151
8.1.	CONCLUSIONES	151
8.2.	RECOMENDACIONES.....	152
BIBLIOGRAFÍA		154
ANEXOS		158

RESUMEN

La investigación titulada “Sistema de Información de recaudación Tributaria de la Municipalidad Distrital de Santillana Bajo el Marco JIRA SCRUM AGILE SOFTWARE”, de la provincia de Huanta departamento de Ayacucho, en el área de Sub Gerencia de Tesorería y Abastecimiento. El objetivo de esta investigación es Aplicar el marco Jira Scrum Ágile Software que nos permite el desarrollo del Sistema Información de Recaudación Tributario del impuesto predial de la Municipalidad Distrital de Santillana, Huanta, Perú, 2019, para la realización de la gestión regular de las expectativas del cliente y para la gestión de los equipos auto dirigidos y auto organizados. El estudio es de nivel descriptiva por que se utiliza técnicas de entrevista en las reuniones de planificación que se da con el representante de la municipalidad y el equipo del proyecto. La investigación es observacional, retrospectivo y transversal, en la investigación se busca abstraer los conceptos relevantes del marco Jira Scrum Ágile Software, para presentar modelo hibrido que permitirá desarrollar el Sistema Información de Recaudación Tributaria del Impuesto Predial. Se afirma que la gestión regular de las expectativas del cliente es un beneficio en las reuniones de las iteraciones de Sprint en Jira Scrum Ágile y se afirma que la gestión de los equipos de auto dirigidos y auto organizados es un beneficio para el Producto Owner, Scrum Master y los Scrum Team.

En la investigación se busca abstraer los conceptos relevantes del marco Jira SCRUM Agile Software, para presentar modelo hibrido que permitirá desarrollar el Sistema Información de Recaudación Tributaria del Impuesto Predio Urbano e Impuesto Predio Rural.

Palabras claves: Jira Scrum, Tributaria, Expectativas del cliente, Auto dirigidos, Auto organizado.

INTRODUCCIÓN

La Municipalidad Distrital de Santillana son órganos del Gobierno, que tienen la obligación del desarrollo local, parte de esto es la recaudación tributaria Municipal. Una de sus funciones de área de tesorería es la recaudación tributaria para el beneficio de la Municipalidad, pero cuyo área tiene un deficiente proceso de recaudación debido a no contar con un Sistema de Información de Recaudación Tributaria del Impuesto Predio Urbano e Impuesto Predio Rural, a consecuencia no tenía ingreso durante mucho tiempo, estas deficiencias motivaron a llevar la investigación aplicando Jira Scrum Agile Software para la entrega de un producto software de alta calidad que reúna los principios de gestión y prácticas de programación, cumpliendo gestión regular de las expectativas del cliente, Equipos auto dirigidos y auto organizados.

Jira Scrum Agile Software es una herramienta de gestión de proyectos ágil muy efectiva tanto para Scrum como para Kanban. Ahora es muy poderoso en ese sentido. el marco jira Scrum agile ha escalado utilizando herramientas y procesos estándar para brindar visibilidad al éxito en toda la empresa, no solo para software.

Los principales objetivos son: a) Aplicar el marco Jira Scrum Agile Software para la realización de la gestión regular de las expectativas del cliente que nos permita el desarrollo del Sistema Información de Recaudación Tributaria del impuesto predial de la Municipalidad distrital de Santillana, Huanta, 2019. b) Aplicar el marco Jira Scrum Agile Software para la gestión de los equipos auto dirigidos y auto organizados que nos permitan el desarrollo del Sistema Información de Recaudación Tributaria del impuesto predial de la Municipalidad distrital de Santillana, Huanta, 2019.

CAPÍTULO I

PLANTEAMIENTO DE LA INVESTIGACIÓN

1.1. DIAGNÓSTICO Y ENUNCIADO DEL PROBLEMA

La Municipalidad Distrital de Santillana que compone en total de 115 Centros Poblados, creada mediante Ley N° 3000 el 21 de diciembre de 1918, en el gobierno del presidente José Pardo y Barreda, ubicado a 3,239 msnm en la provincia de Huanta y departamento de Ayacucho.

En la actualidad las Municipalidades aledañas en mayoría como la Municipalidad Distrital de Santillana tiene un déficit de recaudación tributaria por que no cuenta con un sistema de información recaudadora o el sistema que tenían anteriormente que fueron usados dejaron de ser mantenidos por los proveedores, porque sus alcaldes no se preocuparon en gestionar, por ello son obsoletos al momento de usarlo ya que ocurren muchas fallas.

Al no tener un sistema de autoevaluó la población y la Municipalidad Distrital de Santillana no se benefician, por ejemplo, la población no puede acceder a un techo propio porque se necesita un requisito de la evaluación del predio, para tal fin muchos propietarios deben ir hasta la ciudad de Huanta para gestionar la evaluación de sus predios, a esto se suma la demora de la burocracia, en tanto pierden muchos de ellos la oportunidad solo por no saber a tiempo la exactitud del valor total de su terreno y el predio a pagar. Según censo 2017 en la siguiente tabla se puede ver cantidad de viviendas que pudrían encontrarse en perjudicación ver Tabla 1.

Tabla 1*Vivienda particular censado con ocupación presente, por tipos de vivienda particular, según distrito, 2017*

PERÚ: VIVIENDAS PARTICULARES CENSADAS CON OCUPANTES PRESENTES, POR TIPO DE VIVIENDA PARTICULAR, SEGÚN DISTRITO, 2017

(Cifras absolutas)

Departamento	Provincia	Distrito	Total de viviendas particulares	Casa independiente	Departamento en edificio	Vivienda en quinta	Casa en vecindad	Choza	Vivienda improvisada	Local no destinado para habitación humana	Otro tipo de vivienda particular
Ayacucho	Huanta	Huanta	9967	9550	27	185	100	90	2	13	0
Ayacucho	Huanta	Ayahuanco	421	390	0	0	0	30	0	1	0
Ayacucho	Huanta	Huamanguilla	1289	1282	0	0	0	1	0	6	0
Ayacucho	Huanta	Iguain	883	873	0	0	0	10	0	0	0
Ayacucho	Huanta	Luricocha	1762	1757	0	2	0	0	1	2	0
Ayacucho	Huanta	Santillana	1110	1055	0	5	33	15	0	2	0
Ayacucho	Huanta	Sivia	3199	3040	0	19	25	111	2	2	0
Ayacucho	Huanta	Llochegua	2894	2887	0	0	2	0	1	5	0
Ayacucho	Huanta	Canayre	1041	1027	0	1	0	8	0	5	0
Ayacucho	Huanta	Uchuraccay	1053	983	0	0	0	70	0	0	0
Ayacucho	Huanta	Pucacolpa	708	573	0	4	2	128	0	1	0
Ayacucho	Huanta	Chaca	628	607	0	0	2	19	0	0	0

Nota. Datos obtenidos Censos Nacionales 2017: XII de Población, VII de Vivienda y III de Comunidades Indígenas. INEI.

La Municipalidad es perjudicado en la recaudación tributaria, por no contar con un sistema recaudador de previo urbano y rural. Y eso que está dentro de su jurisdicción y con los límites que señala la ley, leyes que está en su facultad para poder crear, modificar y suprimir contribuciones y tasas, dentro del Decreto Legislativo N° 776.

El sistema de Autovaluo de la Municipalidad Distrital de Santillana del año 2008, estaba desarrollado en visual basic y un gestor de base de datos como SQL server 2006. El problema ocurrió cuando empezó subir el misma contribuyente dos veces, esto género que los datos se repitiera, algo que ese entonces el base de datos no tenía sus llaves principales y llaves foráneos, otro factor del error del sistema fue el trasladar a otro máquina de una persona no autorizada, porque solamente podía realizar la persona que había desarrollado el sistema y no así el trabajador de la municipalidad y como último el sistema llenaba datos en el campo del recibo que no lo correspondía.

Se evalúa el sistema de autovaluo creado en el año 2008 llamado Sysautovaluo en el siguiente Tabla 2.

Tabla 2

Evaluación del SysAutovaluo

EVALUACIÓN DEL "SYSAUTOVALUO"	ESTANDAR
CODIGO FUENTE	
el código fuente se tiene	NO
el código fuente se puede mejorar	NO
el código fuente cuenta con estándares de seguridad	NO
tiene documentación o manuales	NO
BASE DE DATOS	
la Base de Datos se tiene	SI
se puede mejorar la Base de Datos	NO
cumple con las estándares de Desarrollo	NO
se tiene Usuario y contraseña al acceso al Base de Datos	NO
se puede recuperar la base de datos de todo los registros y pagos de la población	NO
el Base de Datos está desarrollado en una versión actual de un gestor de Base de datos	NO
ESTANDARES DE NORMA	
Resolución Ministerial N° 370-2018-Vivienda	NO
Decreto supremo N° 153-2004-EF Valores totales de predio	SI

resoluciones ministeriales N° 126-2007-Vivienda y N° 266-2012-Vivienda de la tabla de depreciación por antigüedad y estado de conservación según el material estructural predominante	SI
IMPUESTO PREDIAL	
Registro de Declaraciones Juradas	SI
Predios Urbanos y Rústicos	SI
Cálculo de Base Imponible	SI
Mantenimiento de Tablas para el cálculo	NO
Cálculo y recalcado automático	NO
Información confiable de los contribuyentes y su Predios	NO
ARBITRIOS MUNICIPALES	
Determinación de Arbitrios Municipales	SI
Calculo por fecha de declaración	SI
Mantenimiento de Tablas para el cálculo	NO
Cálculo y recalcado automático	NO
FISCALIZACIÓN	
Registro de la Base de Datos de la Fiscalización ejecutada sobre determinados contribuyentes	NO
Cálculo automático obteniendo las diferencias en las Cuentas Corrientes	NO
Resoluciones de Determinación	SI

Nota. Fuente elaboración propia, que nos representa la evaluación del sysautotvaluo del año 2008.

1.2. FORMULACIÓN DEL PROBLEMA DE INVESTIGACIÓN

PROBLEMA PRINCIPAL

¿Cómo el marco **Jira Scrum Agile Software** permite el desarrollo del **Sistema Información de Recaudación Tributaria del impuesto predial** de la Municipalidad distrital de Santillana, Huanta, 2019?

PROBLEMAS SECUNDARIOS

- a. ¿Cómo el marco Jira Scrum Agile Software realiza la gestión regular de las expectativas del cliente en el desarrollo del Sistema Información de Recaudación Tributaria del impuesto predial de la Municipalidad distrital de Santillana, Huanta, 2019?
- b. ¿Cómo el marco Jira Scrum Agile Software gestiona los equipos auto dirigidos y auto organizados en el desarrollo del Sistema Información de

Recaudación Tributaria del impuesto predial de la Municipalidad distrital de Santillana, Huanta, 2019?

CAPÍTULO II

OBJETIVOS DE LA INVESTIGACIÓN

2.1. OBJETIVO GENERAL

Aplicar el marco **Jira Scrum Agile Software** que nos permite el desarrollo del **Sistema Información de Recaudación Tributaria del impuesto predial** de la Municipalidad distrital de Santillana, Huanta, 2019

2.2. OBJETIVOS ESPECÍFICO

- a) Aplicar el marco **Jira Scrum Agile Software para la realización de la gestión regular de las expectativas del cliente** que nos permita el desarrollo del Sistema Información de Recaudación Tributaria del impuesto predial de la Municipalidad distrital de Santillana, Huanta, 2019.
- b) Aplicar el marco **Jira Scrum Agile Software para la gestión de los equipos auto dirigidos y auto organizados** que nos permitan el desarrollo del **Sistema Información de Recaudación Tributaria del impuesto predial** de la Municipalidad distrital de Santillana, Huanta, 2019.

CAPÍTULO III

JUSTIFICACIÓN Y DELIMITACIÓN DE LA INVESTIGACIÓN

3.1. IMPORTANCIA DEL TEMA

IMPORTANCIA TÉCNICA

El Jira Software que permite cualquier metodología ágil, ya sea el Scrum o Kanban que puede gestionar proyectos ágiles hasta informes, puede planificar, rastrear y administrar todos los proyectos de desarrollo de software que contribuye un desarrollador, eligiendo un marco de ayuda a su equipo a lanzar software de calidades de estándar y mayor velocidad.

Jira SCRUM Agile Software gestiona versiones, fácil preparación del trabajo atrasado, planificación de sprint (iteración de trabajo de duración fija) y punto de historia. Con esto los desarrolladores con su interacción con su equipo, adoptan mejores prácticas de programación de la metodología SCRUM de Jira Software.

IMPORTANCIA ECONÓMICA

La adecuada gestión de proyecto con las herramientas ágiles para SCRUM con Jira Software, permitirá el flujo de trabajo que arme sus propias estados y transacciones de cada tipo de problema como errores, historias, etc. que aprovecha para actualizar automáticamente en función al evento. Con esta información podemos estimar el costo correspondiente, asimismo la metodología de desarrollo de SCRUM sobre Jira Software por lo que reducirá el costo de los cambios generados en etapas posteriores en el desarrollo del proyecto.

3.2. JUSTIFICACIÓN

La investigación se lleva a cabo por los diversos motivos como:

- a) La utilización de nuevas tecnologías como Jira Scrum Agile Software según Xavier M. (2016), afirma que “Las nuevas tecnologías es fundamental para la optimización de los procesos de negocios y aumentar la eficiencia. Por esto, la mayoría de las compañías necesitan disponer de

un software adecuado que cubra todas las necesidades empresariales” (p.2).

- b) Una de las preocupaciones centrales es que la mayoría de la población como de San José de Secce capital del Distrito de Santillana y sus 115 centros poblados, no pueden acceder al programa del techo propio de forma inmediata porque no cuentan con un sistema que automatiza su Autovaluo de su predio rural o urbano, como uno de los requisitos fundamentales fondo mivivienda.
- c) El alcance de este proyecto de investigación es que Distrito de Santillana contara con un sistema de Autovaluo que puede valorizar el valor de su propiedad anualmente, por lo que la población se beneficiara para calcular con exactitud su costo de su propiedad para respectiva venta, acceder a programa de techo propio o para los fines que crea conveniente.
- d) Un problema real que tiene la Municipalidad de Santillana como ente del gobierno es de no optimizar la recaudación tributaria municipal, porque no cuenta con sistema automatizado que puede calcular el predio con los estándares y normal legales como Resolución Ministerial N° 370-2018-Vivienda de los cuadros de valores unitarios oficiales de edificación, Decreto supremo N° 153-2004-EF Valores totales de predio y resoluciones ministeriales N° 126-2007-Vivienda y N° 266-2012-Vivienda de la tabla de depreciación por antigüedad y estado de conservación según el material estructural predominante.
- e) Con la investigación realizada se puede utilizar el sistema en otros Municipales cercanos que cumplen con las necesidades de la población y aranceles iguales como la Municipalidad de Chaca y Municipalidad de Uchuraccay de la provincia de Huanta y departamento de Ayacucho.

3.3. DELIMITACIÓN

En la investigación nos limitaremos al análisis de la metodología de SCRUM y el marco de trabajo Jira Software, ya que el primero será control y planificar proyectos con un gran volumen de cambio de última hora y trabajo en equipo a partir de iteraciones o Sprint y el segundo admite cualquier metodología ágil de gestión de proyecto para el desarrollo de software.

CAPÍTULO IV

REVISIÓN DE LA LITERATURA

4.1. ANTECEDENTES DE LA INVESTIGACIÓN

Según Javier E. (2017), en su proyecto de grado denominado “Kanban y Scrumbar Orientados a Proyectos de Tecnología de la Información”, del Instituto Universitario Aeronáutico, define que Jira es un producto cuyo uso está extendido en todo tipo de compañías a lo largo del mundo, tanto tecnológicas como cualquier otra que requiera de una gestión de proyectos, procesos, incidencias, defectos y/o actividades. En este sentido, Jira también es un facilitador de comunicaciones y de seguimiento de trabajo fundamental para agilizar los procesos en un flujo de trabajo. La herramienta permite definir y hacer seguimiento a proyectos basados en los marcos de trabajo Scrum, Kanban y Scrumban, proporcionando funcionalidades como: Creación de historias de usuario, estimación de las historias en la planificación de la iteración, visualización de la actividad del equipo, reporte de progreso, entre otros.

Pilar R. y Juan G González (2008), en su Tesis de Máster "Estudio de la Aplicación de Metodologías Ágiles para la Evolución de Productos Software", en la Universidad Politécnica de Madrid, concluye que la metodología SRCUM, es un proceso para la gestión y control del producto que trata de eliminar la complejidad en estas áreas para centrarse en la construcción de software que satisfaga las necesidades del negocio. Es simple y escalable, ya que no establece prácticas de ingeniería del software, sino que se aplica o combina, fácilmente, con otras prácticas ingenieriles, metodologías de desarrollo o estándares ya existentes en la organización.

Ríos (2017), en sus Tesis “Determinación de la recaudación de los impuestos municipales” de la Pontificia Universidad Católica del Perú concluye que, en los últimos años, a pesar de que se han realizado esfuerzos por incrementar los niveles de recaudación de impuestos de las municipalidades con una mejor normatividad e instrumentos, la gran mayoría de municipios aún dependen de manera importante de las transferencias que reciben. En nuestro país, la recaudación de impuestos municipales no ha sido estudiada de manera extensiva. Según los resultados obtenidos a partir de la aplicación de los modelos econométricos de datos

de panel se observa que la recaudación de los Impuestos municipales está determinada por factores macroeconómicos y socioeconómicos propios de cada distrito, así como por variables asociadas a la gestión municipal.

Según Llacctahuman, J. (2015), en su Tesis nominado “Sistema integral para mejorar la calidad de información en la recaudación tributaria de la municipalidad distrital de el tambo” de la Universidad Nacional del Centro del Perú, define que después de la implementación del Sistema Integral, se tiene una adecuada administración y procesamiento de datos, haciendo más confiable la información que se maneja, permitiendo que las notificaciones o información que se entregue a los contribuyentes sean más exactos, conllevando que el contribuyente tenga mayor conciencia tributaria y se apersonen a la municipalidad ya sea para actualizar sus declaraciones juradas, o en muchos de los casos para hacer efectivo el pago del autoevaluó.

Yucra, B. (2015), en su Tesis nominado “La gestión tributaria municipal y propuesta para el mejoramiento de la recaudación impuesto predial en Municipalidad Distrital de Cayma 2015” de la Universidad Nacional de San Agustín de Arequipa, concluye que el impuesto predial es un tributo municipal de trato directo entre la administración tributaria y el ciudadano, ya que la municipalidad es la encargada de la Recaudación, Administración y fiscalización de dicho impuesto. El tema de la recaudación tributaria esta generalmente percibido como un trabajo poco grato, pues las personas evitan con frecuencia el pago de este impuesto, razón por la que ha generado un monto altísimo en morosidad en la municipalidad de Cayma y en respuesta a esto la administración tributaria otorga campañas de Amnistías que no hace más que aumentar y fomentar la inconciencia tributaria, sumado a esto no se actualizado las formas de pago mediante la red bancaria, internet, etc.

4.2. MARCO TEÓRICO

4.2.1. SISTEMA DE INFORMACIÓN DE RECAUDACIÓN TRIBUTARIA

Verónica, C. (2015), En su tesis “Implementación de un sistema de información para la recaudación del impuesto de alcabala ‘sipria’ de la municipalidad provincial de puno - 2015”, define:

Es un conjunto de elementos que interactúan entre sí con el fin de apoyar las actividades de una empresa o negocio. Teniendo muy en cuenta el equipo computacional necesario para que el sistema de información pueda operar y el recurso humano que interactúa con el Sistema de Información, el cual está formado por las personas que utilizan el sistema. (p.9)

Alicia, J. (2016), estará basado en “un modelo dimensional que, por un lado, permite describir los datos en un lenguaje más próximo al de los usuarios y, por otro, está orientado a la ejecución de consultas sobre grandes volúmenes de datos” (p.7).

a) TRIBUTO MUNICIPAL

Marco normativa. (2015). Define que “tributario municipal es el conjunto de tributos (impuestos, tasas y contribuciones) cuya administración se encuentra a cargo de las municipalidades, sean éstas de ámbito provincial o distrital. Asimismo, forman parte de este sistema las normas tributarias municipales y las entidades municipales encargadas de administrar tales tributos” (P.17). Entendemos que todo tributo municipal recae en todos los municipales provinciales y distritales del estado peruano, entonces cada entidad percibe sus ingresos como del impuesto predial, impuesto de alcabala, impuesto al patrimonio automotriz, impuesta a las apuestas, impuesta a los juegos e impuesto a los espectáculos públicos.

Todas las municipalidades están facultadas a promover el desarrollo en su comunidad dentro de su jurisdicción, porque es muy importante la recaudación tributaria municipal a su población. La recaudación principalmente está basada en los impuestos predial y recae sobre el valor de los predios urbanos y rústicos en base a su autovaluo.

b) AUTOVALUO

Según Mendoza (2019). Un autoavalúo, también conocido como autoavalúo, es un proceso formal mediante el cual se estima el valor de una propiedad. Ello se hace con la única finalidad de establecer la base de cálculo de algunos impuestos municipales. A dicha base impositiva se le denomina Valor Total de Autoavalúo, o también base imponible. Las municipalidades peruanas asignan a cada uno de tus bienes inmuebles un valor estimado del mismo. En la mayoría de las ocasiones, este valor resulta inferior al precio que aquel tiene en el mercado. Situación que te resulta favorable, ya que implica que cancelas un impuesto menor por ellas.

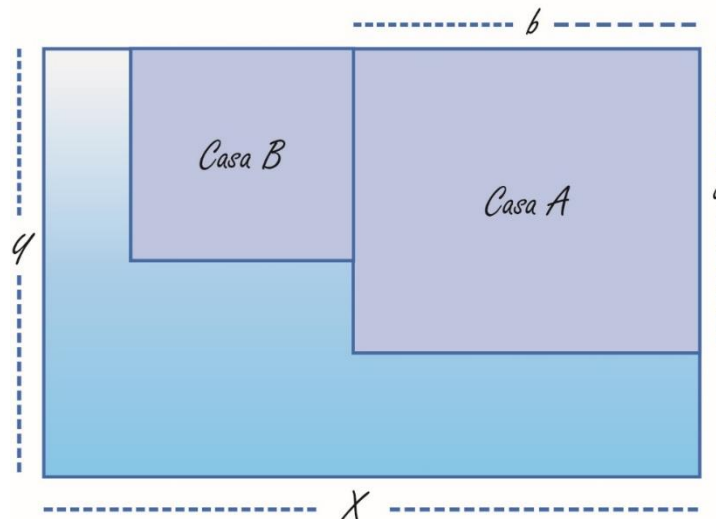
c) CALCULO DE AUTOVALUO

Para poder calcular el Autoavaluo o llamado Base Imponible es de la siguiente manera:

Pondremos un ejemplo de un terreno con su respectiva construcción de casa, a base a eso hallaremos el Base Imponible.

Figura 1

Terreno del contribuyente



Nota. Fuente elaboración propia, que nos representa un terreno con dos casas A y B.

- Área del terreno: $A_t \rightarrow A_t = x * y$
- Área de construcción de la casa: $A_c \rightarrow A_c = a * b$
- Valor unitario por m_2 :

$$V_{um_2} \rightarrow V_{um_2} = \sum \text{Valor Unitario de Edificación}$$

Entonces la suma de valor unitario de edificación es:

$$\begin{aligned} \sum \text{Valor Unitario de Edificación} \\ = \text{Suma}(\text{muros y columnas} + \text{techo} + \text{puerta y ventana} \\ + \text{revestimiento} + \text{baño} + \text{instalación eléctrico y sanitario}) \end{aligned}$$

Todo este valor de unitario de edificación sale de la Normas Legales de Resolución Ministerial N° 370-2018-VIVIENDA.

- Valor de depreciación: V_d

Entonces el valor de depreciación es:

$$\begin{aligned} V_d \\ = \text{es el porcentaje de (antigüedad, material estructural predominante,} \\ \text{estado de conservación)} \end{aligned}$$

El valor de la depreciación está en la tabla de depreciación del Ministerio de Vivienda Construcción y Saneamiento aprobado mediante resolución ministeriales N° 126-2007-VIVIENDA y N° 266-2012-VIVIENDA.

Ya sabiendo nuestras variables con sus respectivos datos ahora hallaremos el Valor de la Construcción (V_c) y Valor del Terreno (V_t).

$$V_c = (V_{um_2} - V_d) * A_c$$

$$V_t = A_t * \text{Arancel}$$

Teniendo el valor de la construcción y valor del terreno ya podemos saber autovaluo o Base Imponible (BI).

$$BI = V_t + V_c$$

Por lo que tendríamos como respuesta de la casa A del primer piso y si queremos hallar para el segundo piso o más pisos de la casa A el procedimiento sería el mismo, en tanto para la casa B también el procedimiento sería el mismo, en consecuente se suma por cada iteración BI, concluyamos que autovaluo es:

$$\text{Autovaluo} = \sum BI$$

d) IMPUESTO PREDIAL

Ley tributo municipal (1993), en su artículo 8, establece que “el Impuesto Predial es de periodicidad anual y grava el valor de los predios urbanos y rústicos.

Para efectos del Impuesto se considera predios a los terrenos, incluyendo los terrenos ganados al mar, a los ríos y a otros espejos de agua, así como las edificaciones e instalaciones fijas y permanentes que constituyan partes integrantes de dichos predios, que no pudieran ser separadas sin alterar, deteriorar o destruir la edificación”. Entonces una propiedad predial es un tipo de bien que se consideran como tales como a los terrenos, casas, edificios, etc. y grava a esto el valor periódicamente el impuesto predial.

Zevala (2019). Informe legal. Describe que “el Impuesto Predial es un tributo de periodicidad anual que grava el valor de los predios urbanos y rústicos, el cual se determina en base a la declaración jurada de autoavalúo que presenta el contribuyente. La recaudación, administración y fiscalización del impuesto corresponde a la municipalidad distrital donde se encuentre ubicado el predio” (p.12).

El predio como tal es una finca, tierra o una propiedad que esto está avalado con las normas del estado que se entiende al subsuelo y al sobresuelo. Pero según las leyes del estado peruano el subsuelo no debe comprender los recursos naturales, los yacimientos, los restos arqueológicos, ni otros bienes regidos por leyes especiales.

e) **CALCULO DE PREDIO**

Sabiendo nuestro valor de autoavaluo en el caso anterior podemos saber nuestro predio con lo que pagaremos nuestro impuesto anual o trimestral según la posibilidad de socioeconómico y sociocultural.

Veamos para poder a pagar total debemos colocar nuestro valor de autoavaluo o Base Imponible en el siguiente rango de la tabla de UIT y multiplicar con el porcentaje de alícuota correspondiente.

Tabla 3

Tramo de Autoavaluo Alícuota

Tramo de Autoavaluo	Alícuota
hasta 15 UIT	0.20%
más de 15 UIT y hasta 60	0.60%

UIT

Por lo tanto, tendríamos nuestro resultado de impuesto a pagar en:

Impuesto anual: $I_a = BI * alicuota$

Impuesto trimestral: $I_t = (BI * alicuota)/4$

A. IMPUESTO PREDIO URBANO

Reglamento nacional de tasación del Perú (2007), artículo II.a.02, considera al predio urbano al que “está situado en centro poblado y se destine a vivienda, comercio, industria o cualquier otro fin urbano; así como los terrenos sin edificar, siempre que cuenten con los servicios generales propios del centro poblado y los que tengan terminadas y recibidas sus obras de habilitación urbana, estén o no habilitadas legamente”.

B. IMPUESTO PREDIO RURAL

Reglamento nacional de tasación del Perú (2007), artículo III.a.01, considera al predio rural o rustico a los “terrenos ubicados en zona rural dedicados a uso agrícola, pecuaria, forestal y de protección y a los eriazos susceptibles de destinarse a dichos usos que no hayan habilitados como urbanos ni estén comprendidos dentro de los límites de expansión urbana. Forman parte del predio rustico, la tierra, el agua y su ecosistema, así como los construcciones e instalaciones fijas y permanentes que existan en él”.

4.2.2. JIRA SCRUM AGILE SOFTWARE

Ha estado presente por un tiempo; originalmente, era un sistema de emisión de tickets (un tipo de software de seguimiento de errores). La gestión de proyectos ha evolucionado a lo largo de los años, y los procesos ágiles se han vuelto cada vez más populares, por lo que JIRA se ha convertido en una herramienta de gestión de proyectos ágil muy efectiva tanto para Scrum como para Kanban. Ahora es muy poderoso en ese sentido (David H, 2018, p.22).

Según David, H. (2018), en su proyecto, implementa “un marco Jira Scrum Agile escalado utilizando herramientas y procesos estándar para brindar visibilidad

al éxito en toda la empresa, no solo para el software. Scrum se ha convertido en aún más” (p.14).

A. GESTIÓN REGULAR DE LAS EXPECTATIVAS DEL CLIENTE

Gestión regular de las expectativas del cliente, basada en resultados tangibles, es una de las ventajas o beneficios de Scrum, según Karenny, B. (2009), define:

El cliente establece sus expectativas indicando el valor que le aporta cada requisito del proyecto y cuando espera que esté completado; y comprueba de manera regular si se van cumpliendo sus expectativas, da feedback, ya desde el inicio del proyecto puede tomar decisiones informadas a partir de resultados objetivos y dirige estos resultados del proyecto, iteración a iteración, hacia su meta. (p.76)

B. EQUIPOS AUTO DIRIGIDOS Y AUTO ORGANIZADOS

También llamado equipo Scrum (Scrum Team), según Ken S. y Jeff S. (2013), define:

Los Equipos Scrum son auto organizados y multifuncionales. Los equipos auto organizados eligen la mejor forma de llevar a cabo su trabajo y no son dirigidos por personas externas al equipo. Los equipos multifuncionales tienen todas las competencias necesarias para llevar a cabo el trabajo sin depender de otras personas que no son parte del equipo. El modelo de equipo en Scrum está diseñado para optimizar la flexibilidad, la creatividad y la productividad. Los Equipos Scrum entregan productos de forma iterativa e incremental, maximizando las oportunidades de obtener retroalimentación. Las entregas incrementales de producto “Terminado” aseguran que siempre estará disponible una versión potencialmente útil y funcional del producto. (p. 6)

4.3. MÉTODOS Y TÉCNICAS

4.3.1. SOFTWARE JIRA

“JIRA es una herramienta de gestión de proyectos ágil que admite cualquier metodología ágil. Desde paneles ágiles hasta informes, puede planificar,

rastrear y administrar todos sus proyectos de desarrollo de software ágil con una sola herramienta” (David Harned, 2018, p.18).

Según Ravi, S. (2015), en su estudio que proporciona la información de su libro lo cual define que Jira Software como:

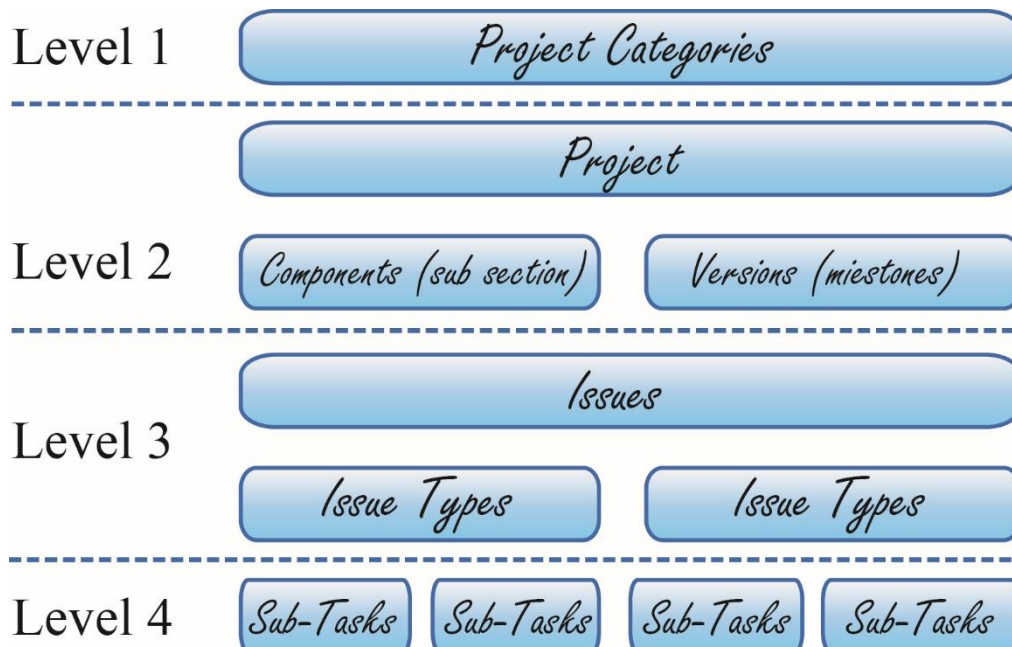
Un sistema de seguimiento de problemas patentado. Se utiliza para rastrear errores, resolver problemas y administrar funciones del proyecto. Existen muchas herramientas de este tipo disponibles en el mercado, pero lo mejor de JIRA es que se puede configurar fácilmente y ofrece una amplia gama de personalizaciones. Fuera de la caja, JIRA ofrece funcionalidades de seguimiento de defectos / errores, pero también se puede personalizar para actuar como un sistema de servicio de asistencia, un conjunto de gestión de pruebas simple o un sistema de gestión de proyectos con trazabilidad de extremo a extremo. (p.8)

a) **ARQUITECTURA DE JIRA**

Para entender el concepto básico nos enfocaremos en siguiente gráfico.

Figura 2

Arquitectura de JIRA



Nota. Ravi, S. (2016, p. 23). Mastering JIRA 7.

Categorías de proyectos: Es cuando tenemos muchos proyectos en JIRA y es importante separar todos los proyectos en varias categorías. Esto podría representar las unidades de negocio, cliente o equipos de una empresa.

Proyectos: todo proyecto JIRA es una colección de temas como por ejemplo puede ser para coordinar un servicio de un producto, realizar un seguimiento de un proyecto, administrar un servicio de asistencia, etc., depende de lo que pide el cliente.

Componentes: con subsecciones de un proyecto y se utiliza para agrupar en partes más pequeñas de un proyecto.

Versiones: son un punto en el tiempo para un proyecto. Te ayudan a programar y organizar tus lanzamientos.

Tipos de problemas: permitirá crear más de un tipo de problemas que son diferentes entre sí en términos de qué tipo de información almacenan. JIRA viene con tipos de problemas predeterminados, como errores, tareas y sub tareas, pero puede crear más tipos de problemas que puedan seguir su propio flujo de trabajo y tener Conjunto diferente de campos.

Sub tareas: están divididos en dos tareas en: estándar y sub tareas, que son elementos secundarios de una tarea estándar.

b) EQUIPOS DE JIRA SOFTWARE

“Jira Software está diseñado para que todos los miembros de tu equipo de software puedan planificar, supervisar y publicar un magnífico software” (Atlassian, 2017).

Definiendo los equipos de jira software en:

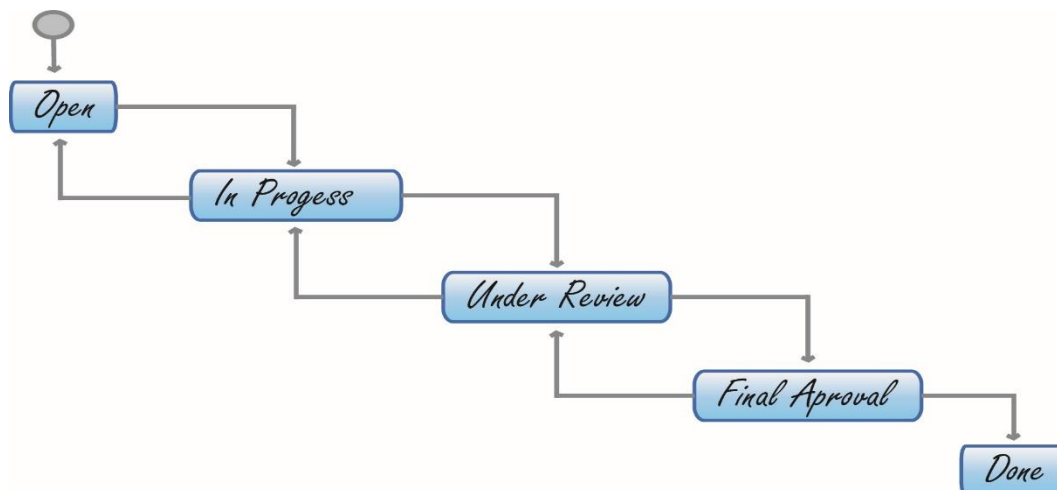
- **Planifica.** - Crea historias de usuario e incidencias, planifica sprints y distribuye tareas entre tu equipo de software.
- **Supervisa.** - Prioriza y analiza el trabajo de tu equipo en su contexto y con una completa visibilidad.
- **Lanza.** - Realiza lanzamientos con confianza y seguridad, sabiendo que la información que tienes es siempre la más actualizada.
- **Crea informes.** - Mejora el rendimiento del equipo con datos visuales en tiempo real que tu equipo puede emplear.

c) **FLUJO DE TRABAJO JIRA SOFTWARE**

“Cada equipo tiene su propio proceso para la entrega de software. Utiliza un flujo de trabajo predeterminado o crea uno que se adapte a la forma de trabajar de tu equipo” (Atlassian, 2017).

Figura 3

Flujo de Trabajo JIRA Software



Nota. Atlassian, (2017). Flujo de Trabajo JIRA Software. Recuperado de <https://www.atlassian.com/es/software/jira>

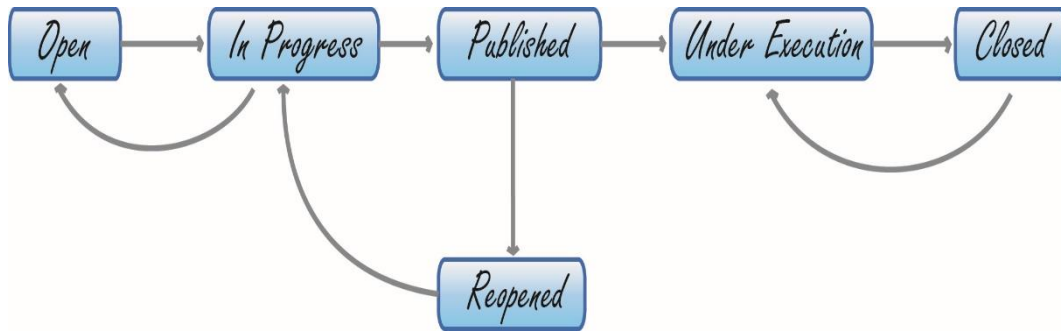
d) **FLUJO DE TRABAJO PARA LA CAMPAÑA DE PRUEBA**

Cada empresa tiene sus respectivos procesos y modelos de software, según Ravi, S. (2015), define el flujo de trabajo para las campañas de prueba como:

Una colección de casos de prueba. Ya hemos creado su tipo de problema. El usuario comenzará primero con la creación de un nuevo número de campaña de prueba de tipo y luego todos los casos de prueba que forman parte de ella se crearán como una sub tarea. La campaña de prueba podría asignarse a un líder de equipo específico, que podría asignar más casos de prueba entre los miembros del equipo. El siguiente diagrama representa un flujo de trabajo típico para una campaña de prueba. (p.43)

Figura 4

Flujo de trabajo para la campaña de prueba



Nota. Ravi, S. (2016, p. 22). Mastering JIRA 7.

Entonces podemos sacar la conclusión según la gráfica que la campaña de prueba moverá al estado en progreso y los casos de prueba se preparan y crean un sub tarea. Después de finalizar se publicará en la campaña de prueba. Bajo ejecución significa que se están ejecutando actualmente.

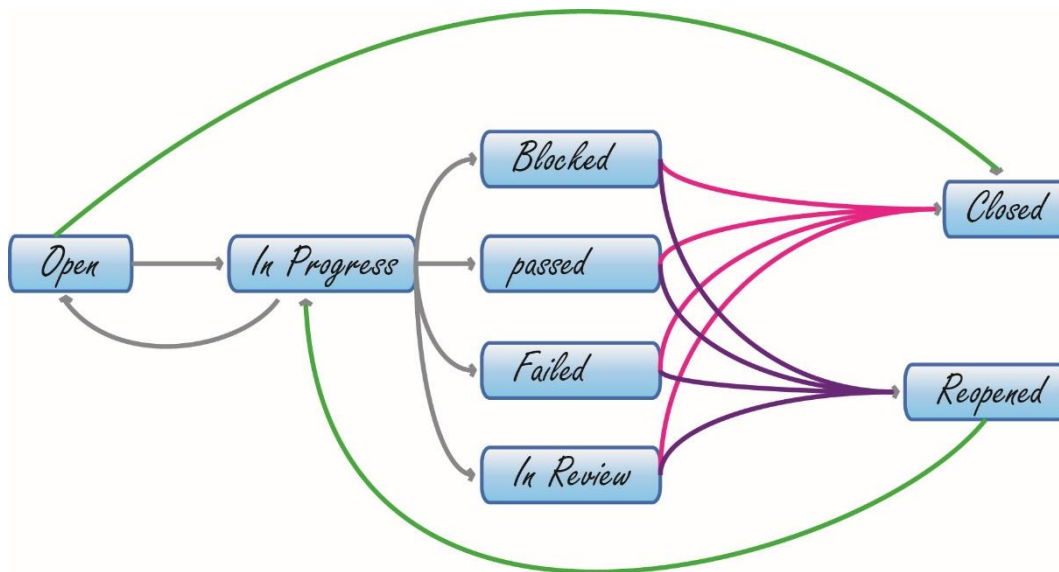
e) **FLUJO DE TRABAJO PARA CASO DE PRUEBA**

Según Ravi, S. (2015), en su libro describe el flujo de trabajo para caso de prueba que:

Caso de prueba contiene la información de la entrada, salida esperada, junto con un conjunto de acciones y detalles del entorno que define si se cumple un requisito según el plan original para verificar su funcionamiento. Configuraremos JIRA para pruebas manuales donde almacenaremos toda esta información como Tipos de problemas. (p.44)

Figura 5

Flujo de trabajo para caso de prueba



Nota. Ravi, S. (2016, p. 24). Mastering JIRA 7.

El caso de prueba será una sub tarea de una campaña de prueba en particular. Después de haberse creado, primero se moverá al estado en progreso cuando el probador comience a realizar las pruebas. Sobre la base de estas pruebas, se puede mover a uno de los siguientes estados:

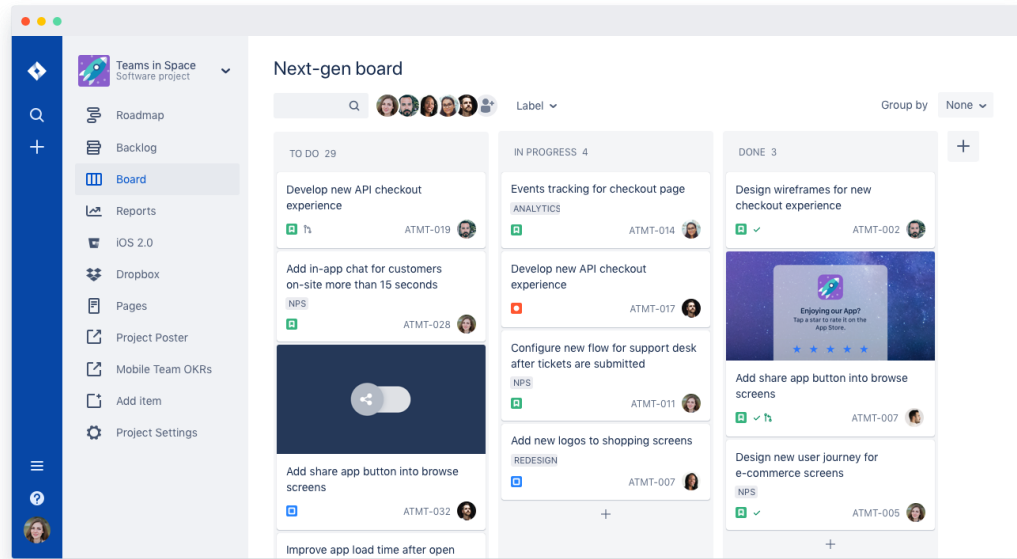
- **Bloqueado.** - si este caso de prueba es crítico para que otras funciones funcionen
- **Aprobado.** - si funciona correctamente
- **Falló.** - si no funciona
- **En revisión.** - si requiere más investigación

f) **TABLEROS DE JIRA SCRUM**

El tablero de Jira Scrum según Atlassian. (2017), es “la herramienta que une a los equipos en torno a objetivos únicos y promueve una entrega iterativa e incremental. el tablero de Jira Scrum es la visualización de su progreso durante el ciclo de desarrollo”.

Figura 6

Tableros de jira scrum



Nota. Atlassian, (2017). Tablero de JIRA Scrum. Recuperado de <https://www.atlassian.com/es/software/jira/features>

El tablero de Jira Scrum es mucho más que un panel de tareas y entre sus funciones destacan:

- Aumentar la comunicación y la transformación
- Promover la planificación de sprints y el desarrollo iterativo
- Mejorar el enfoque y la organización del equipo

4.3.2. MARCO DE TRABAJO SCRUM

“Scrum es un enfoque ágil para desarrollar productos y servicios innovadores” (Kenneth S., 2012, p.13)

Según Schwaber y Sutherland (2013) Es un marco de trabajo por el cual las personas pueden acometer problemas complejos adaptativos, a la vez que entregar productos del máximo valor posible productiva y creativamente.

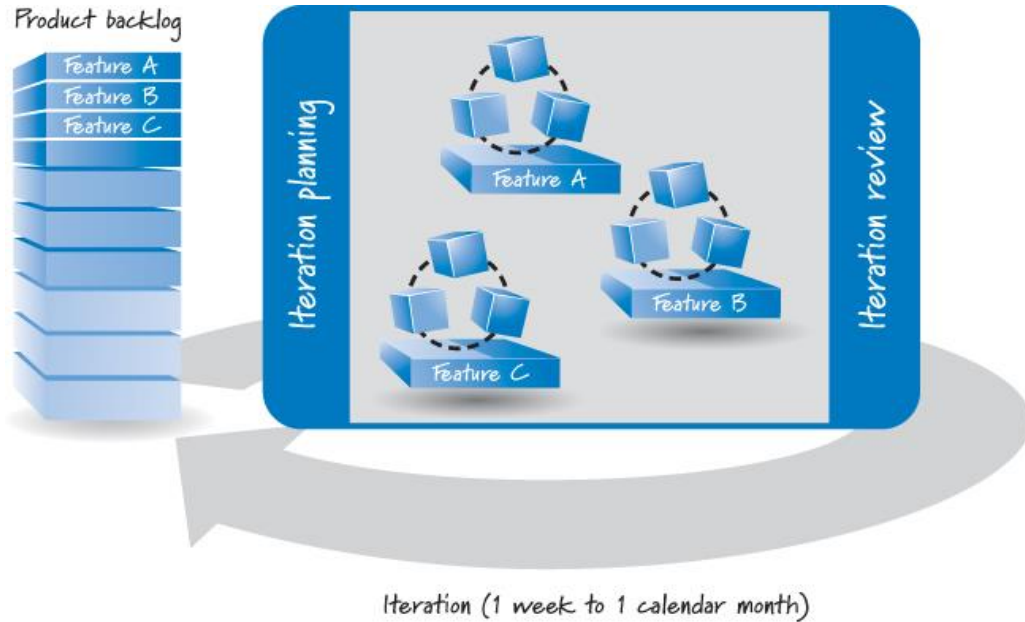
Según Díaz (2009) define a Scrum, como una colección de procesos para la gestión de proyectos, que permiten centrarse en la entrega de valor para el cliente y la potenciación del equipo para lograr su máxima eficiencia, dentro de un esquema de mejora continua.

Según Sommerville (2011) afirma que Scrum es un método ágil que ofrece un marco de referencia para la administración del proyecto. Se centra alrededor de un conjunto de Sprints, que son periodos fijos cuando se desarrolla un incremento

de sistema. La planeación se basa en priorizar un atraso de trabajo y seleccionar las tareas de importancia más alta para un Sprint.

Figura 7

Resumen de desarrollo ágil

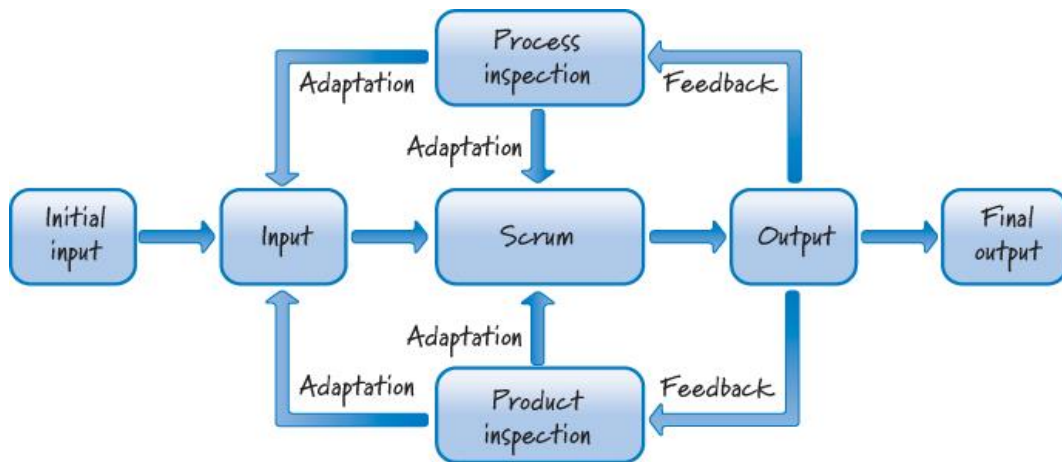


Nota. Kenneth, S. (2012, p. 2). Essential Scrum. A practical guide to the most popular agile process.

“En el corazón de Scrum se encuentran los principios de inspección, adaptación y transparencia” (Schwaber y Beedle, 2001)

Figura 8

Modelo de proceso Scrum



Nota. Kenneth, S. (2012, p. 36). Essential Scrum. A practical guide to the most popular agile process.

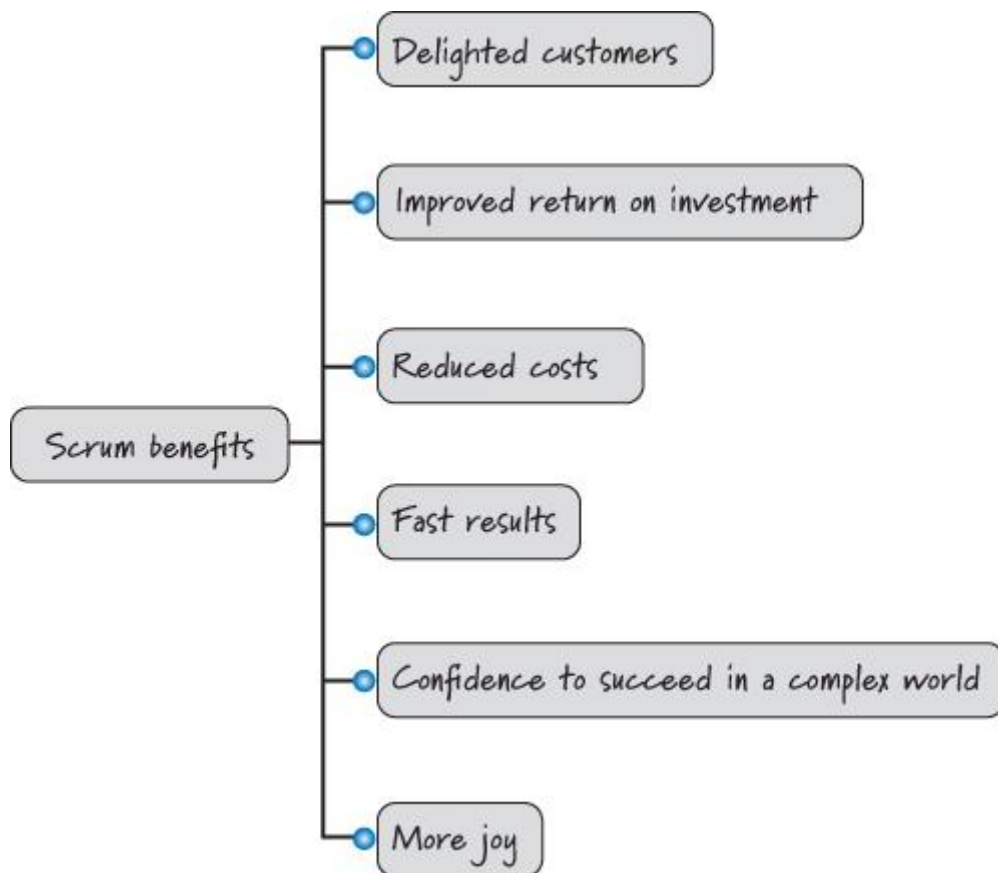
4.3.2.1. BENEFICIOS DE SCRUM

La organización sea pequeña o grande que hayan aplicado SCRUM están experimentando una realidad diferente y según Kenneth S. (2012) define:

Scrum en entregar características de trabajo, integradas, probadas y valiosas para el negocio cada iteración conduce a resultados rápidos. Scrum también es adecuado para ayudar a las organizaciones a tener éxito en un mundo complejo donde deben adaptarse rápidamente en función de las acciones interconectadas de competidores, clientes, usuarios, organismos reguladores y otras partes interesadas. Y Scrum brinda más alegría a todos los participantes. ¡No solo los clientes están encantados, sino que las personas que hacen el trabajo realmente lo disfrutan! Disfrutan de una colaboración frecuente y significativa, lo que lleva a mejores relaciones interpersonales y una mayor confianza mutua entre los miembros del equipo. (p.14)

Figura 9

Beneficios de Scrum



Nota. Kenneth, S. (2012, p. 6). Essential Scrum. A practical guide to the most popular agile process.

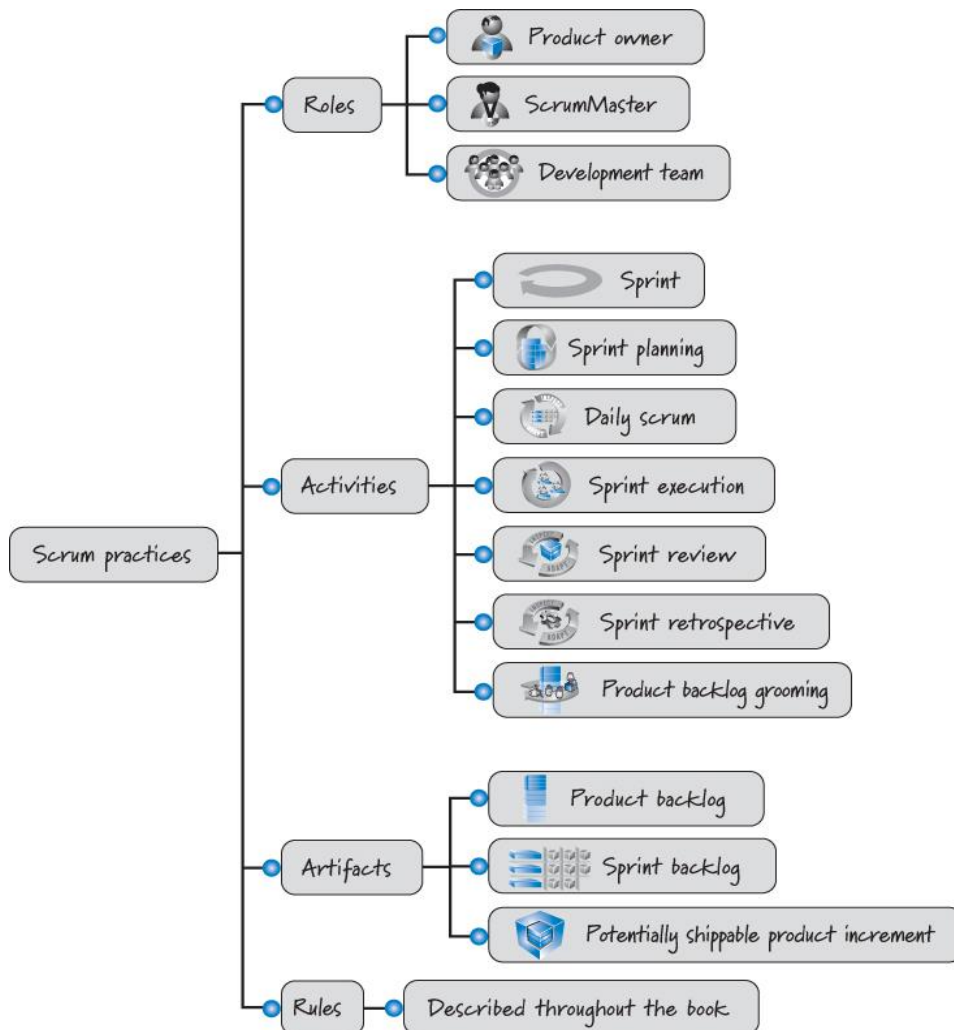
4.3.2.2. MARCO DE SCRUM

Scrum es un marco centrado en personas, en valores, honestidad, apertura, coraje, respeto, enfoque, confianza, empoderamiento y colaborador. Según Kenneth S. (2012) define el marco de Scrum:

Scrum es un marco para organizar y gestionar el trabajo. El marco de Scrum se basa en un conjunto de valores, principios y prácticas que proporcionan la base a la cual su organización agregará su implementación única de prácticas de ingeniería relevantes y sus enfoques específicos para realizar las prácticas de Scrum. El resultado será una versión de Scrum que es exclusivamente suya. (p.15)

Figura 10

Prácticas SCRUM



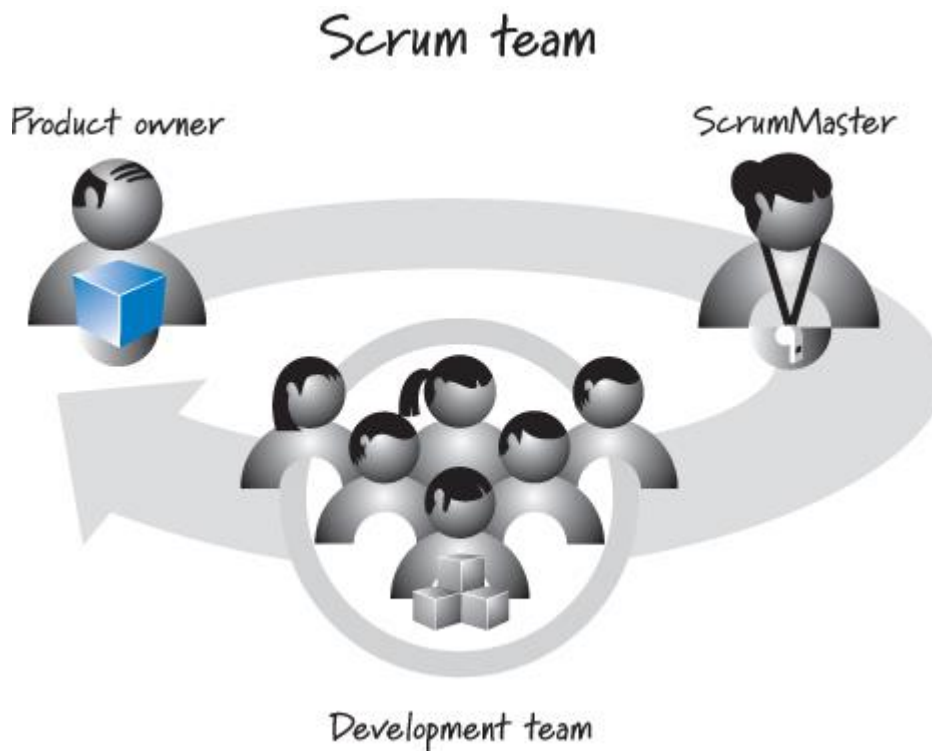
Nota. Kenneth, S. (2012, p. 14). Essential Scrum. A practical guide to the most popular agile process.

4.3.2.3. ROLES SCRUM

“El desarrollo de Scrum consisten en uno o más equipos de Scrum , cada uno compuesto por tres roles de Scrum: propietario del producto , ScrumMaster y el equipo de desarrollo” (Kenneth S., 2012, p.15).

Figura 11

Rol de Scrum



Nota. Kenneth, S. (2012, p. 15). Essential Scrum. A practical guide to the most popular agile process.

a. **El dueño de Producto (Product Owner)**

Según Adkins, L. (2010), define es la única autoridad responsable de decidir qué características y funcionalidades construir y el orden en el cual construirlas. El propietario del producto mantiene y comunica a todos los demás participantes una visión clara de lo que el equipo Scrum está tratando de lograr. Como tal, el propietario del producto es responsable del éxito general de la solución que se está desarrollando o manteniendo

El propietario del producto es el punto central empoderado del liderazgo del producto. El es la única autoridad responsable de decidir qué características y funcionalidades construir y el orden en el cual construirlos. El dueño del producto mantiene y comunica a todos los demás participantes una visión clara de lo que el equipo Scrum está tratando de lograr. Como tal, el propietario del producto es responsable del éxito general de la solución que está siendo desarrollado o mantenido (Rubín, 2013).

Schwaber y Sutherland (2013) afirman que el dueño de producto es el responsable de maximizar el valor del producto y del trabajo del equipo de desarrollo. El cómo se lleva a cabo esto podría variar ampliamente entre distintas organizaciones, Equipos Scrum e individuos. El Dueño de Producto es la única persona responsable de gestionar la Lista del Producto (Product Backlog).

b. El Scrum Master

Según Kenneth S. (2012), define que Scrum Master ayuda a todos los involucrados a comprender y adoptar los valores, principios y prácticas de Scrum. Actúa como entrenadora, brindando liderazgo en los procesos y ayudando al equipo Scrum y al resto de la organización a desarrollar su propio enfoque Scrum específico de alto rendimiento. Al mismo tiempo, Scrum Master ayuda a la organización a través del desafiante proceso de gestión de cambios que puede ocurrir durante una adopción de Scrum.

Según Schwaber y Sutherland (2013), es el responsable de asegurar que Scrum es entendido y adoptado. Los Scrum Masters hacen esto asegurándose de que el Equipo Scrum trabaja ajustándose a la teoría, prácticas y reglas de Scrum. El Scrum Master ayuda a las personas externas al equipo Scrum a entender qué interacciones con el equipo Scrum pueden ser de ayuda y cuáles no. El Scrum Master ayuda a todos a modificar estas interacciones para maximizar el valor creado por el equipo Scrum.

Es el responsable de ayudar a comprender y adoptar los valores, principios y prácticas de Scrum. Ellos actúan como entrenadores tanto para el equipo de desarrollo como para el propietario del producto. También proporciona liderazgo de procesos, ayudando al equipo Scrum y al resto de la organización (Rubín, 2013).

c. El Equipo de desarrollo (Development Team)

Según Schwaber y Sutherland (2013), consiste en los profesionales que desempeñan el trabajo de entregar un incremento de producto “Terminado”, que potencialmente se pueda poner en producción, al final de cada Sprint. Solo los miembros del Equipo de Desarrollo participan en la creación del incremento. Los Equipos de Desarrollo son estructurados y empoderados por la organización para organizar y gestionar su propio trabajo. La sinergia resultante optimiza la eficiencia y efectividad del Equipo de Desarrollo.

Kenneth S. (2012), consiste en que el equipo de desarrollo se auto organiza para determinar la mejor manera de lograr el objetivo establecido por el propietario del producto. El equipo de desarrollo suele ser de cinco a nueve personas; sus miembros deben tener colectivamente todas las habilidades necesarias para producir software de trabajo de buena calidad

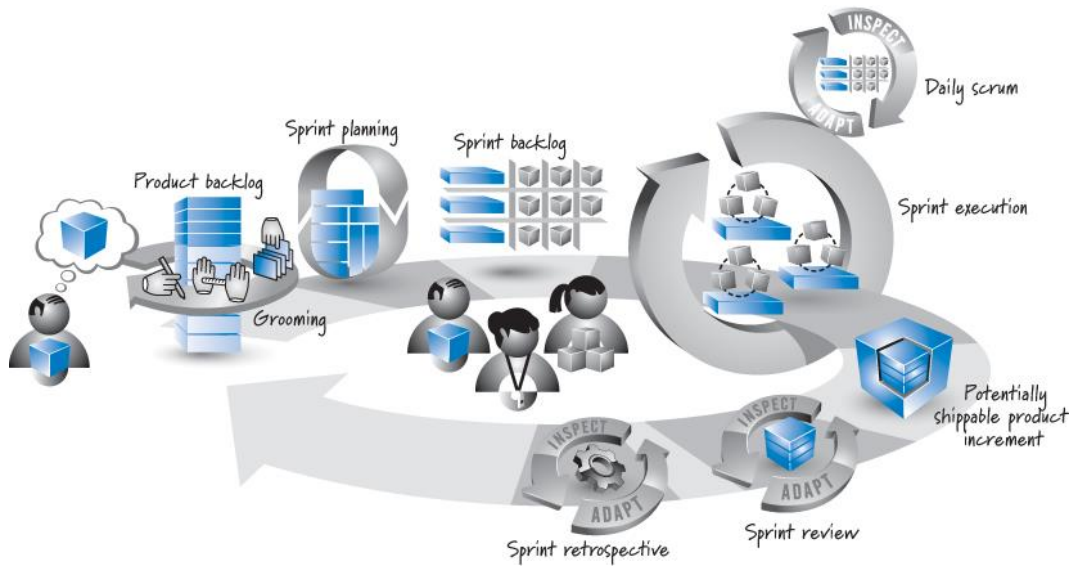
Es una colección multifuncional de personas (arquitecto, programador, tester, administrador de base de datos, diseñador de interfaz de usuario, etc.) que tienen habilidades requeridas para entregar el producto (software) con el valor comercial solicitado por el propietario del producto (Rubín, 2013).

4.3.2.4. ACTIVIDADES

Las actividades y artefactos de SCRUM según Kenneth S. (2012) describe que, el propietario del producto tiene una visión de lo que quiere crear (el gran cubo). Debido a que el cubo puede ser grande, a través de una actividad llamada preparación , se divide en un conjunto de características que se recopilan en una lista priorizada llamada la acumulación de productos.

Figura 12

Marco Scrum



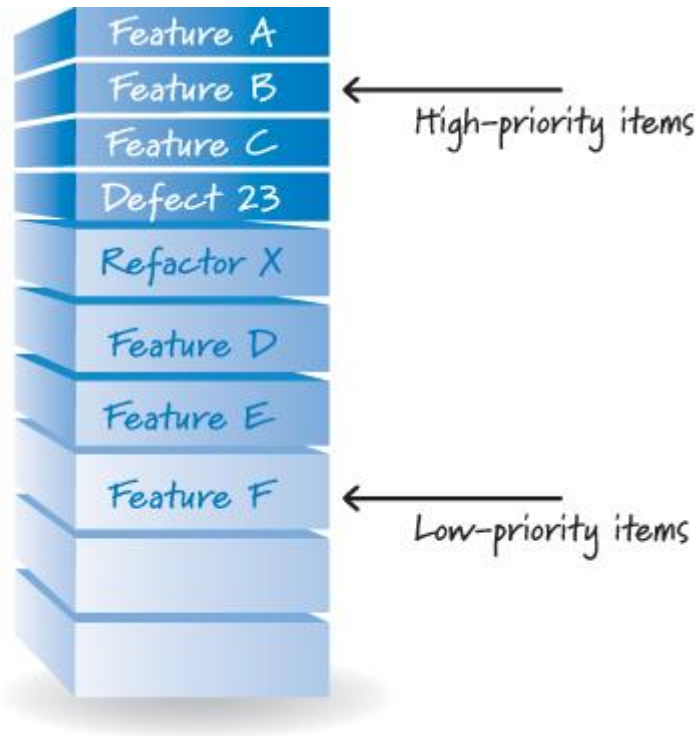
Nota. Kenneth, S. (2012, p. 17). Essential Scrum. A practical guide to the most popular agile process.

a. Pilar de Producto

Kenneth S. (2012). El propietario del producto, con el aporte del resto del equipo Scrum y las partes interesadas, es en última instancia responsable de determinar y administrar la secuencia de este trabajo y comunicarlo en forma de una lista priorizada (u ordenada) conocida como la cartera de productos. En el desarrollo de nuevos productos, los elementos de la cartera de pedidos del producto inicialmente son características necesarias para cumplir con la visión del propietario del producto. Para el desarrollo continuo del producto, la cartera de pedidos del producto también puede contener nuevas características, cambios en las características existentes, defectos que necesitan reparación, mejoras técnicas, etc.

Figura 13

Pila de Producto



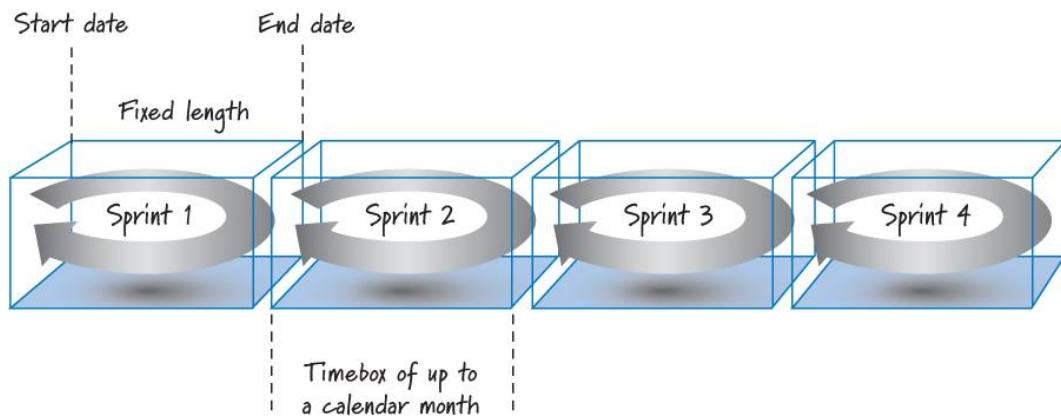
Nota. Kenneth, S. (2012, p. 19). Essential Scrum. A practical guide to the most popular agile process.

b. Sprint

Schwaber y Sutherland (2013), es un bloque de tiempo (time-box) de un mes o menos durante el cual se crea un incremento de producto “terminado”, utilizable y potencialmente desplegable. Es más conveniente si la duración de los Sprints es consistente a lo largo del esfuerzo de desarrollo. Cada nuevo Sprint comienza inmediatamente después de la finalización del Sprint previo.

Figura 14

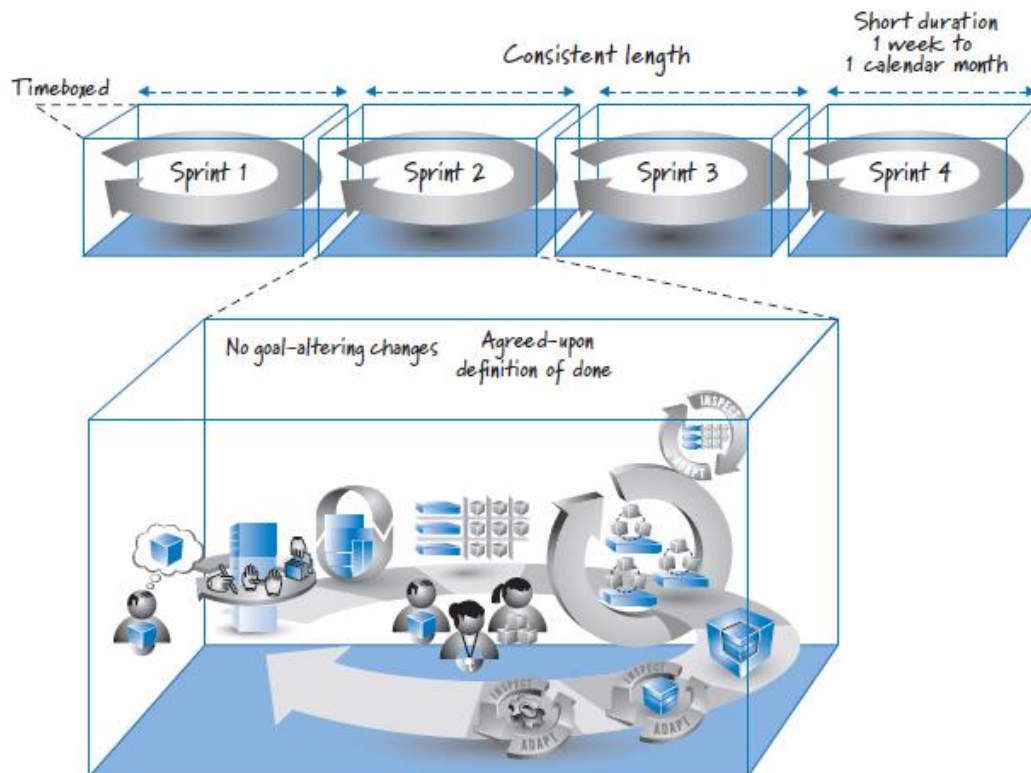
Características del sprint



Nota. Kenneth, S. (2012, p. 21). Essential Scrum. A practical guide to the most popular agile process.

Figura 15

Los Sprints son el esqueleto del marco de Scrum



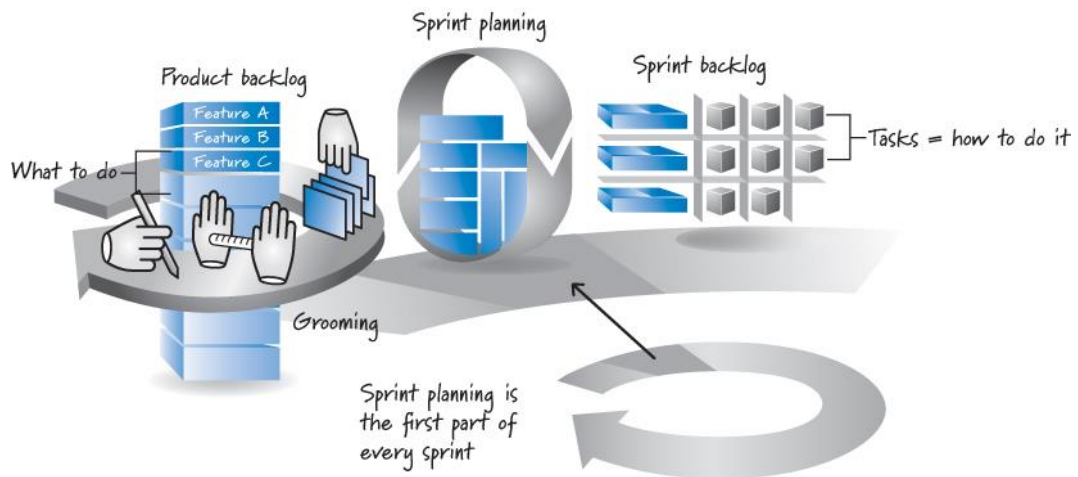
Nota. Kenneth, S. (2012, p. 61). Essential Scrum. A practical guide to the most popular agile process.

c. **Planificación de Sprint (Sprint Plannig)**

Kenneth S. (2012). Define, que es una acumulación de productos puede representar muchas semanas o meses de trabajo, que es mucho más de lo que se puede completar en un solo sprint corto. Para determinar el subconjunto más importante de elementos de la cartera de productos para construir en el próximo sprint, el propietario del producto, el equipo de desarrollo y ScrumMaster realizan la planificación del sprint.

Figura 16

Planificación de sprint



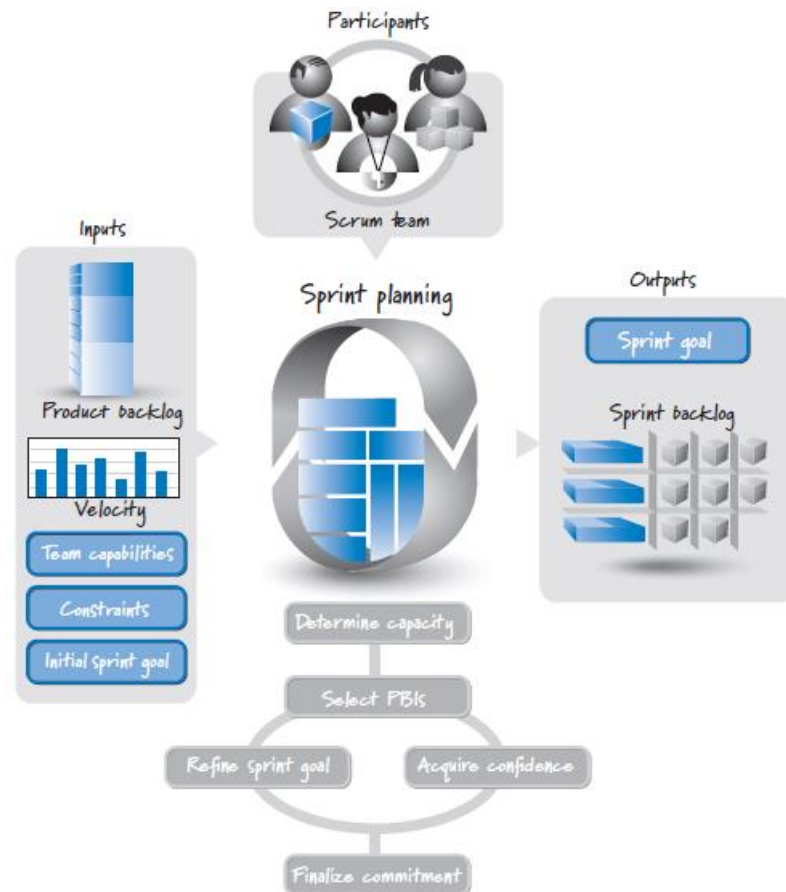
Nota. Kenneth, S. (2012, p. 21). Essential Scrum. A practical guide to the most popular agile process.

Es la planificación del trabajo a realizarse durante el Sprint. Este plan se crea mediante el trabajo colaborativo del equipo Scrum completo. La planificación de Sprint tiene un máximo de duración de ocho horas para un Sprint de un mes. Para Sprints más cortos, el evento usualmente es más corto. El Scrum Master se asegura de que el evento se lleve a cabo y que los asistentes entiendan su propósito. El Scrum Master enseña al equipo Scrum a mantenerse dentro del bloque de tiempo (Schwaber y Sutherland, 2013).

Durante la planificación del Sprint, el propietario del producto y el equipo de desarrollo acuerdan un objetivo de Sprint que define lo que se supone que alcanzará el próximo Sprint. Usando este objetivo el equipo de desarrollo revisan el backlog de producto y determina los elementos de alta prioridad que el equipo puede lograr de manera realista en el próximo Sprint trabajando a un ritmo sostenible, un ritmo al que el equipo de desarrollo pueda trabajar cómodamente por un periodo prolongado de tiempo (Rubín, 2013).

Figura 17

Actividades de la planificación del Sprint



Nota. Kenneth, S. (2012, p. 337). Essential Scrum. A practical guide to the most popular agile process.

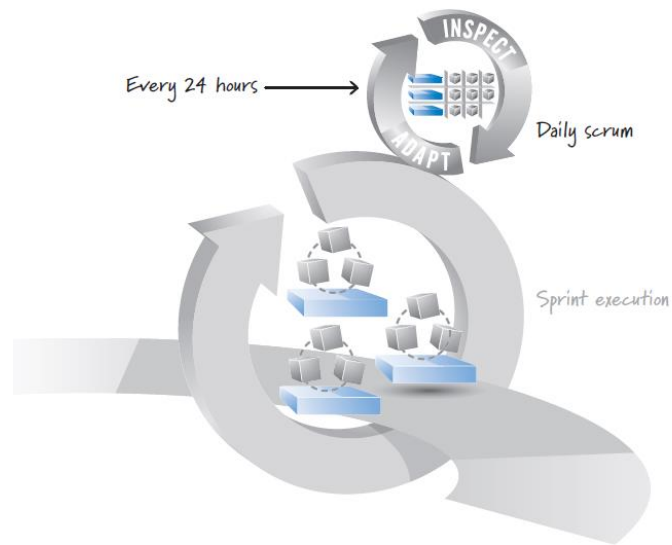
d. Daily Scrum

El Scrum Diario es una reunión con un bloque de tiempo de 15 minutos para que el equipo de desarrollo sincronice sus actividades y cree un plan para las siguientes 24 horas. Esto se lleva a cabo inspeccionando el trabajo avanzado desde el último Scrum Diario y haciendo una proyección que podría completarse antes del siguiente (Schwaber y Sutherland, 2013).

Las reuniones diarias para Scrum, son “conversaciones” de no más de 5-15 minutos, que el Scrum Master tendrá al comienzo de cada día, con cada miembro del equipo. En esta conversación, el Scrum Master deberá ponerse al día de lo que cada miembro ha desarrollado (en la jornada previa), lo que hará en la fecha actual, pero, sobre todo, conocer cuáles impedimentos estén surgiendo, a fin de resolverlos y que el Scrum Team pueda continuar sus labores, sin preocupaciones (Bahit, 2012).

Figura 18

Scrum Diario



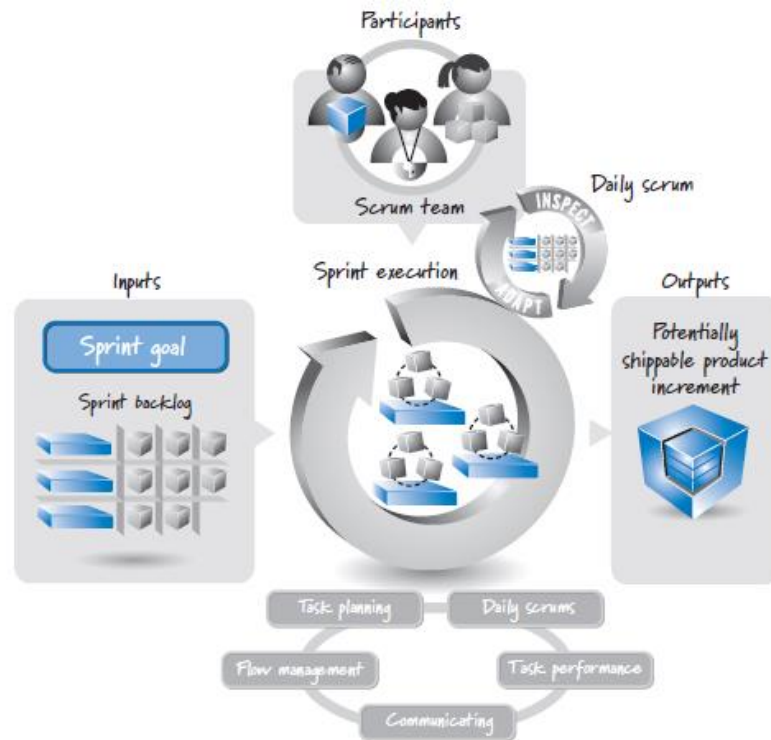
Nota. Kenneth, S. (2012, p. 24). Essential Scrum. A practical guide to the most popular agile process.

e. **Ejecución del Sprint (Sprint execution)**

Durante la ejecución del Sprint, los miembros del equipo de desarrollo realizan activamente trabajo creativo de diseño, construcción, integración y prueba de elementos de la lista de productos en incrementos de funcionalidad. Para hacer esto, se auto organizan y deciden como planificar, administrar y comunicar el trabajo. El equipo de desarrollo pasa la mayor parte de su tiempo realizando ejecución de Sprint (Rubin, 2013).

Figura 19

Actividad de ejecución de Sprint



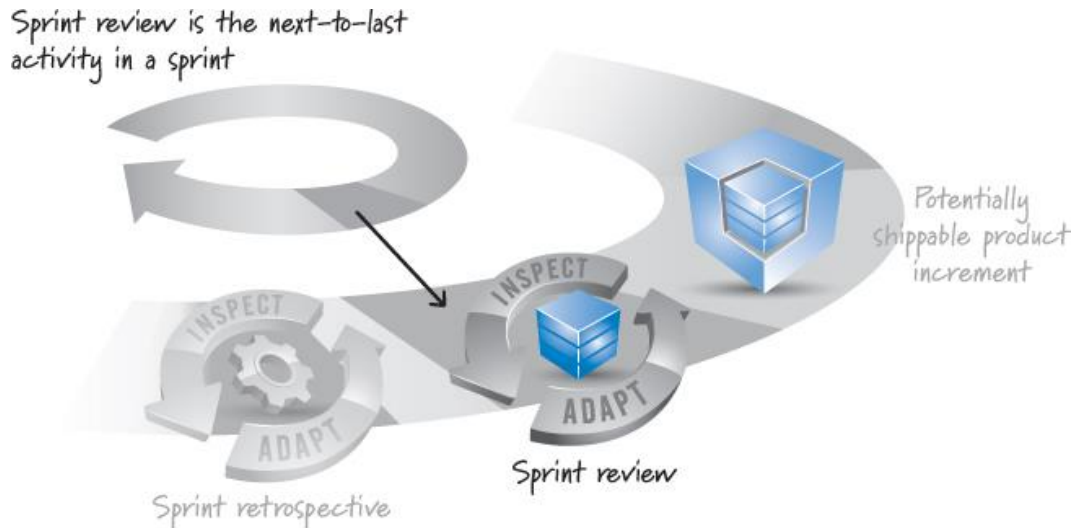
Nota. Kenneth, S. (2012, p. 348). Essential Scrum. A practical guide to the most popular agile process.

f. Revisión de Sprint (Sprint review)

La revisión de Sprint se realiza al final del Sprint, cuya finalidad es para inspeccionar el incremento y adaptar la lista de productos si fuese necesario. Durante la revisión de Sprint, el equipo Scrum y los interesados colaboran acerca de lo que se hizo durante el Sprint, los asistentes colaboran para determinar las siguientes cosas que podrían hacerse para optimizar el valor. Se trata de una reunión informal, no una reunión de seguimiento, y la presentación del incremento tiene como objetivo facilitar la retroalimentación de información y fomentar la colaboración (Schwaber y Sutherland, 2013).

Figura 20

Revisión de Sprint

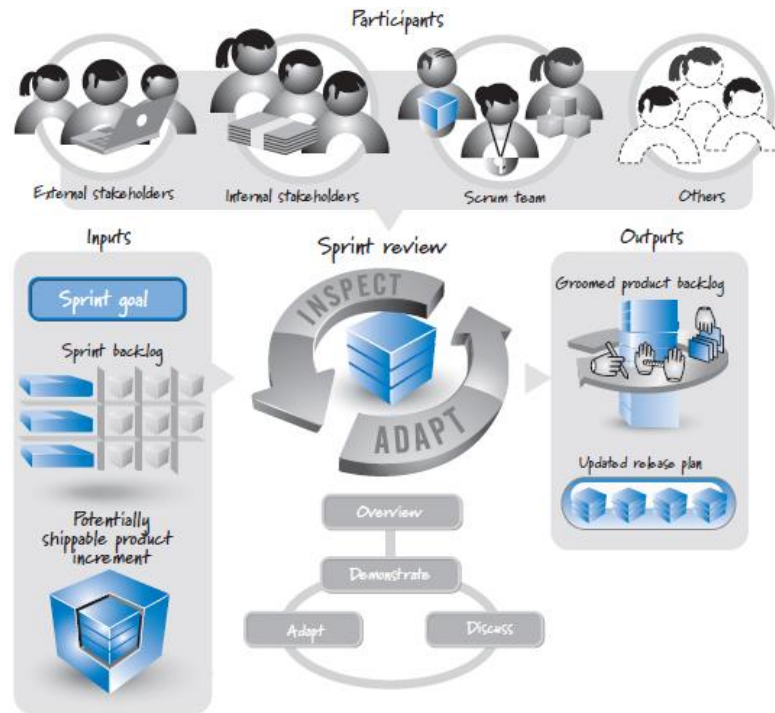


Nota. Kenneth, S. (2012, p. 27). Essential Scrum. A practical guide to the most popular agile process.

Rubin (2013) afirma que el objetivo de esta actividad es inspeccionar y adaptar el producto que se está construyendo. Crítico para esta actividad es la conversación que tiene lugar entre sus participantes, que incluyen el equipo Scrum, stakeholders, patrocinadores, clientes y miembros interesados de otros equipos. La conversación se centra en revisar las características que se acaban de completar en el contexto del esfuerzo de desarrollo general. Todos los asistentes obtienen una visibilidad clara de lo que está ocurriendo y tiene la oportunidad de ayudar a guiar el próximo desarrollo para garantizar la solución más adecuada para el negocio.

Figura 21

Actividad de revisión de Sprint



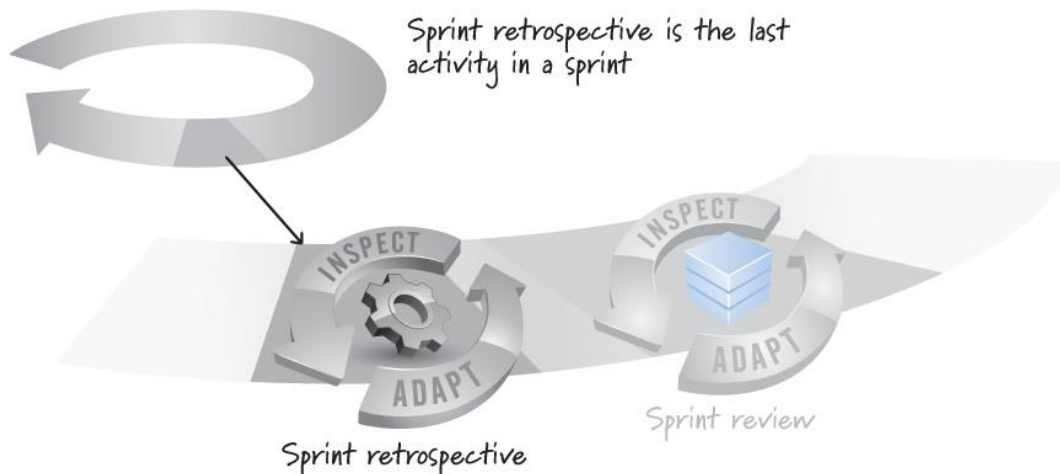
Nota. Kenneth, S. (2012, p. 369). Essential Scrum. A practical guide to the most popular agile process.

g. Retrospectiva de Sprint (Sprint Retrospective)

Kenneth S. (2012). Define que la retrospectiva del sprint es una oportunidad para inspeccionar y adaptar el proceso. Durante la retrospectiva del sprint, el equipo de desarrollo, ScrumMaster y el propietario del producto se unen para discutir qué funciona y qué no funciona con Scrum y las prácticas técnicas asociadas. La atención se centra en la mejora continua del proceso necesaria para ayudar a que un buen equipo Scrum sea excelente. Al final de una retrospectiva del sprint, el equipo Scrum debería haber identificado y comprometido con un número práctico de acciones de mejora de procesos que el equipo Scrum llevará a cabo en el próximo sprint.

Figura 22

Retrospectiva de Sprint



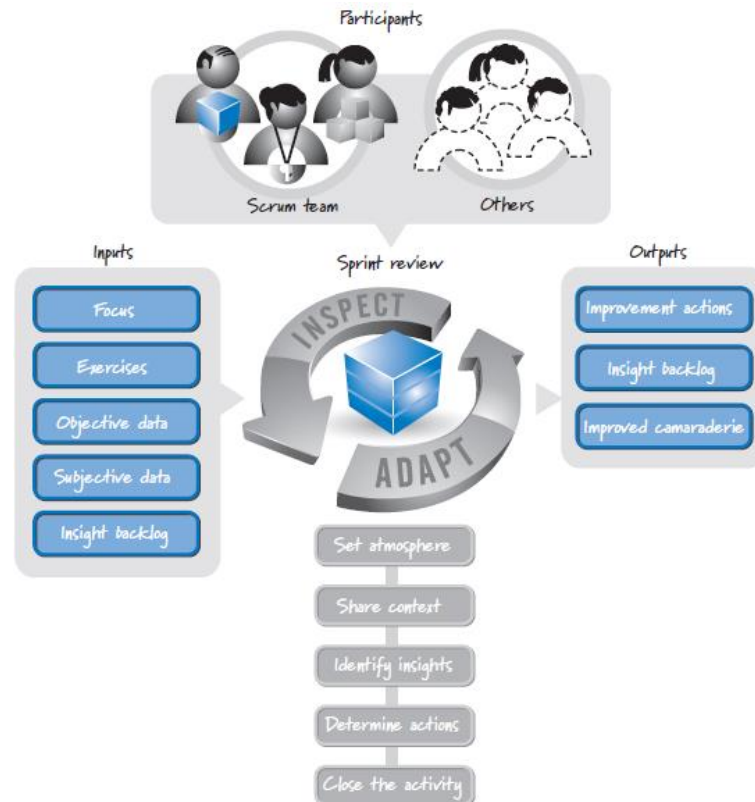
Nota. Kenneth, S. (2012, p. 27). Essential Scrum. A practical guide to the most popular agile process.

Es una oportunidad para el equipo Scrum de inspeccionarse a sí mismo y crear un plan de mejoras que sean abordadas durante el siguiente Sprint. La retrospectiva de Sprint tiene lugar después de la revisión del Sprint y antes de la siguiente reunión de planificación de Sprint. Se trata de una reunión restringida a un bloque de tiempo de tres horas para Sprints de un mes. Para Sprints más cortos se reserva un tiempo proporcionalmente menor (Schwaber y Sutherland, 2013).

Esta actividad ocurre con frecuencia después de la revisión de Sprint y antes de la próxima planificación de Sprint. Durante la retrospectiva de Sprint el equipo de desarrollo, Scrum Master y el propietario del producto se unen para discutir que funciona y que no funciona con Scrum y las prácticas técnicas asociadas. La retrospectiva es una oportunidad para inspeccionar y adaptar el proceso. Al final de una retrospectiva de Sprint, el equipo Scrum debería tener identificado y comprometido un número práctico de acciones de mejora de procesos que será llevado a cabo por el equipo Scrum en el próximo Sprint (Rubin, 2013).

Figura 23

Actividad de retrospectiva de Sprint



Nota. Kenneth, S. (2012, p. 381). Essential Scrum. A practical guide to the most popular agile process.

4.3.2.5. ARTEFACTOS

a. **Lista de Producto (Product backlog)**

Es una lista ordenada de todo lo que podría ser necesario en el producto, y es la única fuente de requisitos para cualquier cambio a realizarse en el producto. La lista de producto enumera todas las características, funcionalidades, requisitos, mejoras y correcciones que constituyen cambios a ser hechos sobre el producto para entregas futuras. Los elementos de la lista de producto tienen como atributos la descripción, la ordenación, la estimación y el valor (Schwaber y Sutherland, 2013).

Es una lista priorizada (u ordenada) de la funcionalidad deseada del producto. El propietario del producto colabora con las partes interesadas internas y externas para reunirse y definir los elementos de la lista de productos. Luego se asegura de que los elementos de la lista de producto se coloquen en la secuencia correcta (utilizando factores como el valor, costo, conocimiento y riesgo) para que los elementos de alto valor aparezcan en la parte superior y los elementos de menor valor aparecen hacia abajo. La lista de productos es un artefacto en constante

evolución, porque los elementos pueden ser agregados, eliminados y revisados por el propietario del producto como los cambios de las condiciones comerciales, o según la comprensión del producto por parte del equipo Scrum (a través de comentarios sobre el software producido durante cada Sprint) (Rubin, 2013).

b. Sprint backlog

Es el conjunto de elementos de la lista de productos seleccionados para el Sprint, más un plan para entregar el incremento de producto y conseguir el objetivo del Sprint. La lista de pendientes del Sprint es una predicción hecha por el equipo de desarrollo acerca de que funcionalidad formará parte del próximo incremento y del trabajo necesario para entregar esa funcionalidad en un incremento terminado (Schwaber y Sutherland, 2013).

c. Incremento del Producto

El incremento es la suma de todos los elementos de la lista de producto completados durante un Sprint y el valor de los incrementos de todos los Sprints anteriores. Al final del Sprint, el nuevo incremento debe estar “terminado”, lo cual significa que está en condiciones de ser utilizado y que cumple la definición de “terminado” del equipo Scrum. El incremento debe estar en condiciones de utilizarse sin importar si el dueño del producto decide liberarlo o no (Rubín, 2013).

4.4. HERRAMIENTAS

4.4.1. APLICACIÓN DE ESCRITORIO

Es una aplicación que está instalado en una computadora y está a la vez funciona sin internet, o boteado en un USB en otras palabras que está instalado en un USB, también aplicaciones de escritorio son a aquellos que están instalados en una máquina virtual también que funciona sin internet. Según portal web Buyto (2012). Define que “Aplicación de Escritorio (Desktop) es aquella que está instalada en el ordenador del Usuario, que es ejecutada directamente por el sistema operativo, ya sea Microsoft Windows, Mac OS X, Linux o Solaris, y cuyo rendimiento depende de diversas configuraciones de hardware como memoria RAM, disco duro, memoria de video, etc.”

La definición de una aplicación de escritorio es clara ya que este debe estar instalado en un ordenador y no necesita internet, pero también según Sergio (2009). En su blog define que “en una aplicación de escritorio normalmente no iniciamos sesión por cada aplicación que usemos, sólo se inicia sesión una vez cuando prendemos el sistema operativo, asumiendo que vamos a abrir una aplicación para ver nuestra lista de tareas”.

Características de aplicación de escritorio

- Son más robustas y estables que las aplicaciones web
- La ejecución no requiere con el exterior(internet), sino de forma local
- La seguridad depende de la complejidad de desarrollo del sistema
- Su acceso es únicamente donde el ordenador este
- Depende del sistema operativo que utilice el ordenador y sus capacidades
- Requiere instalación
- Requerimientos especiales de software

4.4.2. LENGUAJE DE PROGRAMACIÓN JAVA

La definición más clara de lenguaje de programación java en su portal web de Oracle sobre ¿Qué es java?, java (s.f.) define:

Java es una tecnología que se usa para el desarrollo de aplicaciones que convierten a la Web en un elemento más interesante y útil. Java no es lo mismo que javascript, que se trata de una tecnología sencilla que se usa para crear páginas web y solamente se ejecuta en el explorador. Java le permite jugar, cargar fotografías, chatear en línea, realizar visitas virtuales y utilizar servicios como, por ejemplo, cursos en línea, servicios bancarios en línea y mapas interactivos. Si no dispone de Java, muchas aplicaciones y sitios web no funcionarán. Por defecto, Java le notificará inmediatamente que hay nuevas actualizaciones listas para instalarse. Si desea estar al día y mantener la seguridad de su computadora, es importante que acepte e instale las actualizaciones. Si recibe una notificación de actualización de Java en su computadora Windows y no recuerda haberla descargado o instalado, lo más probable es que Java estuviera ya instalado en la nueva computadora.

Según, Joyanes, L. y Zahonero M. (2011). Afirma: “Java es un lenguaje de programación de propósito general, posiblemente, uno de los más populares y

más utilizados en el desarrollo de programas de software, especialmente para internet y web”. (p.20)

Java como lenguaje de programación y plataforma se encuentra en números aplicaciones, dispositivos, redes telecomunicación y comunicaciones, etc. como a la vez en base de datos relacional, servidores web, sistemas de información geográfica, teléfonos móviles, en sistemas medioambientales y bioinformática, etc. como plataforma constituye completa para el desarrollo de software, como a la vez posee una cantidad clases y aplicaciones con números código reutilizables y servicios tales como seguridad, portabilidad, etc.

Gómez (2019). Define “java es un lenguaje de programación de propósitos generales. Podemos usar java para desarrollar el mismo tipo de aplicaciones que programamos con otros lenguajes como C o Pascal”. (p.2)

4.4.3. PROGRAMACIÓN ORIENTADA A OBJETOS (POO)

Es una técnica o estilo para poder programar con objetos según, Barraza, Krol, Meléndez y Velásquez (2006) define:

“La programación orientada a objetos se puede definir como una técnica o estilo de programación que utiliza objetos como bloques esenciales de construcción. Los elementos básicos de la POO son: objetos, mensajes, métodos y clases. La POO es una extensión natural de la actual tecnología de programación y representa un enfoque relativamente nuevo y distinto al tradicional. En la POO, los Antecedentes de la programación orientada a objetos (POO), objetos son los elementos principales de construcción, pero simplemente comprenderlos o usarlos dentro de un programa, no significa que se esté programando en un modo orientado a objetos. Los objetivos de esta modalidad de programación son el mejoramiento de la productividad del programador por medio del manejo de la complejidad del software a través del uso de clases y sus objetos asociados”. (p.10)

Al pasar los años la metodología orientados a objetos comienza a madurar y simular el mundo real a través del significado de objetos que contiene características y funciones, a la vez los objetos residen en sus propiedades como en herencia, abstracción y polimorfismo.

a. Herencia

La herencia es la construcción de un objeto a partir de un objeto existente Camacho, Castañeda y Vargas (2013) define:

La herencia es propiedad que permite a los objetos construirse a partir de otros objetos. El concepto de herencia está presente en nuestras vidas diarias donde las clases se dividen en subclases. Hay diferentes tipos de herencia: los más importantes son simples y múltiples. La Herencia simple consiste en que una clase sólo puede heredar de otra clase llamada clase padre o superclase, en la herencia múltiple una clase puede heredar de una o más clases. (p.5)

b. Abstracción

Es la capacidad de crear tipos de datos definidos por el usuario. Es una definición de tipo que incluye datos y funciones (métodos) en una sola estructura definida llamada clase. La abstracción de datos es un concepto fundamental en el diseño de un programa, pues permite definir el dominio y la estructura de los datos (atributos), junto con una colección o conjunto de operaciones que tienen acceso a los mismos (métodos). La abstracción permite no preocuparse de los detalles accesorios, y generalizar y centrarse en los aspectos que permiten tener una visión global del problema (Barraza, Krol, Meléndez y Velásquez, 2006, p.10).

c. Polimorfismo

El polimorfismo es otra característica esencial de los lenguajes de POO. Un objeto solo tiene una forma, la que se le asigna cuando es creado, pero la referencia a ese objeto es polimorfita ya que puede referirse a objetos de diferentes clases. Para que esta referencia sea polimorfica debe haber necesariamente una relación de herencia entre las clases. Por ejemplo, en nuestro diagrama de clases anterior. Una referencia a objeto de la clase Empleado también puede ser una referencia a objeto de la clase Trabajador (Camacho, Castañeda y Vargas, 2013, p.7).

4.4.4. SISTEMA GESTOR DE BASE DE DATOS MYSQL

Un sistema gestor de base de datos (SGBD) consiste en una colección de datos interconectados y un conjunto de programas para acceder a dichos datos. La

colección de datos normalmente denominadas base de datos, contiene información relevante para una empresa. El objetivo principal de SGBD es proporcionar una forma de almacenar y recuperar la información de una base de datos de manera que sea tanto práctica como eficiente (Silberschatz et al.,2006).

Base de datos está compuesto por datos que están almacenados sistemáticamente para su respectivo uso, MySQL es un base de datos de código abierto ya que tiene muchos contribuyentes y es relacional según, Gilfillan (2003) concluye:

MySQL es un sistema de administración de bases de datos relacional (RDBMS). Se trata de un programa capaz de almacenar una enorme cantidad de datos de gran variedad y de distribuirlos para cubrir las necesidades de cualquier tipo de organización, desde pequeños establecimientos comerciales a grandes empresas y organismos administrativos. MySQL compite con sistemas RDBMS propietarios conocidos, como Oracle, SQL Server y DB2. (p.40)

En actualidad MySQL se puede utilizar en todos los programas de aplicación tanto aplicaciones de escritorio, aplicaciones web y aplicaciones móviles, como a la vez ejecutarse en todo el sistema operativo. MySQL utiliza el lenguaje de consulta estructurado, es un lenguaje que se utiliza también por todos los bases de datos relacionales, es un lenguaje que realiza crear base de datos, agregar, manipular, eliminar y recuperar datos.

4.4.5. PATRÓN DE ARQUITECTURA DE SOFTWARE MVC

Surge a causa de la necesidad de crear software más robusto y la separación de código, en líneas generales y Álvarez (2014) define:

Es una propuesta de diseño de software utilizada para implementar sistemas donde se requiere el uso de interfaces de usuario. Surge de la necesidad de crear software más robusto con un ciclo de vida más adecuado, donde se potencie la facilidad de mantenimiento, reutilización del código y la separación de conceptos. Su fundamento es la separación del código en tres capas diferentes, acotadas por su responsabilidad, en lo que se llaman Modelos, Vistas y Controladores, o lo que es lo mismo, Model, Views & Controllers, si lo prefieres en inglés. En la investigación estudiaremos con detalle estos conceptos, así como las ventajas de ponerlos en marcha cuando desarrollamos el sistema.

A continuación, mostramos una ilustración definiendo el modelo, vista y controlador.

Figura 24

Modelo, vista y controlador



Nota. Vila, J. (2014). Dreamweaver vs. MVC (Modelo-Vista-Controlador o Framework).

Recuperado de <https://blog.ayzweb.com/tutorial/dreamweaver-vs-mvc-modelo-vista-controlador-o-framework>

a. Modelo

Catalani (2007). Define que “El Modelo es el objeto que representa los datos del programa. Maneja los datos y controla todas sus transformaciones. El Modelo no tiene conocimiento específico de los Controladores o de las Vistas, ni siquiera contiene referencias a ellos. Es el propio sistema el que tiene encomendada la responsabilidad de mantener enlaces entre el Modelo y sus Vistas, y notificar a las Vistas cuando cambia el Modelo”.

b. Vista

Catalani (2007). Define que “La Vista es el objeto que maneja la presentación visual de los datos representados por el Modelo. Genera una representación visual del Modelo y muestra los datos al usuario. Interactúa preferentemente con el Controlador, pero es posible que trate directamente con el Modelo a través de una referencia al propio Modelo”.

c. Controlador

Catalani (2007). Define que “El Controlador es el objeto que proporciona significado a las órdenes del usuario, actuando sobre los datos representados por el Modelo, centra toda la interacción entre la Vista y el Modelo. Cuando se realiza algún cambio, entra en acción, bien sea por cambios en la información del Modelo o por alteraciones de la Vista. Interactúa con el Modelo a través de una referencia al propio Modelo”.

4.4.6. JAVA FX

Es una aplicación web que tiene características y funciones de aplicación de escritorio, pero en conclusión Osmandi (2017). Detalla:

JavaFX es un conjunto de paquetes de gráficos y medios que permite a los desarrolladores diseñar, crear, probar, depurar e implementar aplicaciones de cliente enriquecido que operan de forma consciente en diversas plataformas. Con JavaFX, puede crear muchos tipos de aplicaciones. Por lo general, son aplicaciones que cumplen con las redes que se despliegan en múltiples plataformas y muestran información en una interfaz de usuario moderna de alto rendimiento que incluye audio, vídeo, gráficos y animación.

4.4.7. SPRING BOOT

Gutiérrez (2014). Concluye “Spring es un framework de desarrollo de código libre para la plataforma java, por lo tanto, cualquier sistema operativo con una máquina virtual de java puede ejecutar aplicaciones desarrolladas con este framework. Su aspecto modular lo hace flexible y configurable para cualquier tipo de aplicaciones” (p.11).

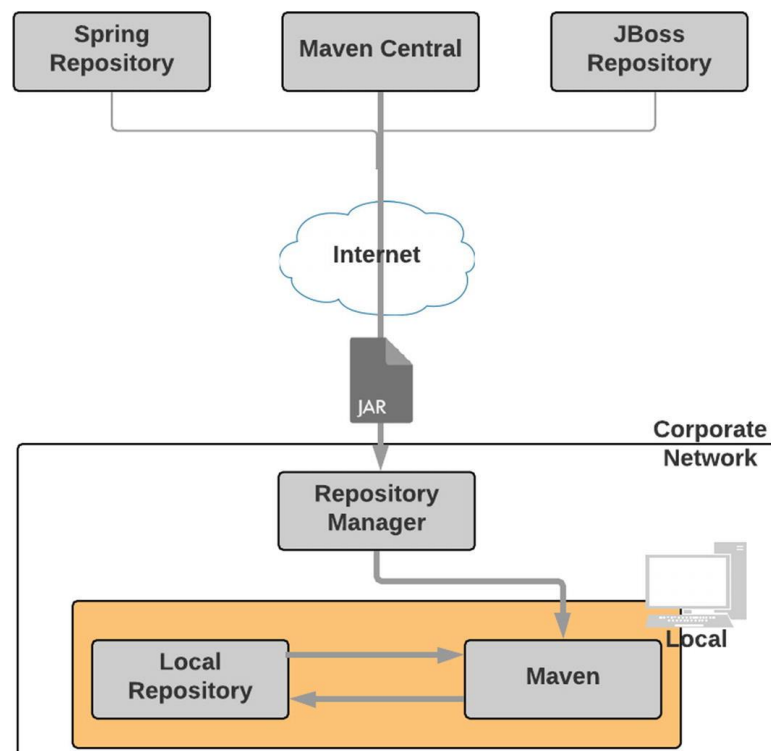
Spring Boot crea aplicaciones independientes y proporciona funciones listas para la función como métricas, comprobación de estado y configuración externa. Por lo general Spring Boot cumple los pasos de crear el proyecto, desarrollamos la aplicación y desplegamos en un servidor.

4.4.8. MAVEN

“Es un marco de gestión de proyectos de código abierto basado en estándares que simplifica la construcción, prueba, informes y empaque de proyectos. simplifica enormemente el proceso de construcción y automatiza las tareas de gestión de proyecto” (Balaji, 2019).

Figura 25

Arquitectura del repositorio de Enterprise Maven



Nota. Balaji, V. (2019). Introducing Maven. A Build Tool for Today’s Java Developers. 2nd ed. Salt Lake City, UT, USA. doi: ISBN 978-1-4842-5409-7e-ISBN 978-1-4842-5410-3.

4.5. POBLACIÓN

Según Monroy (2008), es la totalidad de los elementos que conforman el universo de estudio. Es el conjunto de valores de una variable por el cual existe algún interés. En nuestro campo pueden ser artículos de prensa, editoriales, películas, videos, novelas, series de televisión, programas radiales y por supuesto personas. (López, 2004, p.69)

Según Córdova (2003), se denomina población, a un conjunto de elementos (que consiste de personas, objetos, etc.), que contienen una o más características observables de naturaleza cualitativa o cuantitativa que se pueden

medir en ellos. A cada elemento de una población se denomina unidad elemental o unidad estadística.

4.6. MUESTRA

Según Monroy (2008), Es una parte de una población. El tamaño completo de una población aun siendo finita, puede ser demasiado grande o también a veces no se puede estudiar toda, por cuestiones de costos y recursos. Por eso es necesario o conveniente examinar sólo una fracción (muestra) de la población.

Según Córdova (2003), se denomina muestra a una parte de la población seleccionada de acuerdo con un plan o regla, con el fin de obtener información acerca de la población de la cual proviene. La muestra debe ser seleccionada de manera que sea representativa de la población. Un método de selección de muestras representativas es al azar simple, esto es, cada elemento de la población tiene la misma posibilidad de ser seleccionada para la muestra.

Es un subconjunto o parte del universo o población en que se llevará a cabo la investigación. Hay procedimientos para obtener la cantidad de los componentes de la muestra como fórmulas,

Lógica y otros que se verá más adelante. “La muestra es una parte representativa de la población” (López, 2004, p.69).

Fórmula para hallar muestra:

m = muestra

n = población

z = margen de error (debe expresarse en decimales)

e = es el error muestral deseado

p = es la proporción de individuos que poseen en la población la característica de estudio. Este dato es generalmente desconocido y se suele suponer que $p=q=0.5$ que es la opción más segura.

q = es la proporción de individuos que no poseen esa característica, es decir, es $1-p$

$$m = \frac{n * z^2 * p * q}{(n - 1) * e^2 + z^2 * p * q}$$

CAPÍTULO V

HIPÓTESIS Y VARIABLES DE INVESTIGACIÓN

5.1. HIPÓTESIS

No todas las investigaciones cuantitativas plantean hipótesis. El hecho de que formulemos o no hipótesis depende de un factor esencial: El alcance inicial del estudio. Las investigaciones cuantitativas que formulan hipótesis son aquellas cuyo planteamiento define que su alcance será correlacional o explicativo, o las que tienen un alcance descriptivo, pero que intentan pronosticar una cifra o un hecho (Hernández, Baptista y Fernández, 2014).

Los estudios que no llevan hipótesis no necesitan del planteamiento de hipótesis, y por ello no los hace menos valiosos o menos importantes, simplemente no llevan hipótesis, porque no es la intencionalidad del investigador, el propósito del estudio no expresa la necesidad de afirmar o negar. En la investigación cuantitativa, los estudios que no poseen hipótesis por lo general buscan la cuantificación de la relación entre las variables (Supo, 2012).

En la presente investigación **no se realizará el planteamiento de hipótesis**, considerando que esta investigación es de nivel descriptivo, además no pretendemos pronosticar una cifra o un hecho; sin embargo, lograremos los objetivos específicos planteados.

5.2. DEFINICIÓN CONCEPTUAL DE LAS VARIABLES

VARIABLE ESTUDIO

Sistema de Información de Recaudación Tributaria. Está basado en conjunto de elementos que se relacionan entre sí con el fin de apoyar a la municipalidad u optimizar la recaudación tributaria de la institución pública, tomando como base el equipo computacional ya que este hace funcionar el sistema de información de la recaudación tributaria por cual esta manejado por los trabajadores de la municipalidad.

INDICADORES DE LA VARIABLE DEPENDIENTE

Impuesto Predio Urbano. El predio urbano que está sujeto a un centro poblado o distrito de la zona lejano de ciudad y esto está destinado a vivienda, comercio, industria o cualquier otro fin urbano, como a los terrenos a sin edificar, siempre cuando cuenta con los servicios generales como agua, luz y desagüe.

Impuesto Predio Rural. Son a los terrenos ubicados en zona rural o rustica dedicados especialmente al agrícola, pecuaria, forestal y protección y a los eriazos susceptibles que no hayan sido habitados como urbano. forman parte del predio rustico, la tierra, el agua, y sus ecosistemas, así como los construcciones e instalaciones fijas y permanentes que existen.

VARIABLE DE ESTUDIO

Jira Scrum Agile Software. es una herramienta de gestión de proyectos ágil muy efectiva tanto para Scrum como para Kanban. Ahora es muy poderoso en ese sentido. el marco jira Scrum agile ha escalado utilizando herramientas y procesos estándar para brindar visibilidad al éxito en toda la empresa, no solo para software.

INDICADORES DE LA VARIABLE

Gestión regular de las expectativas del cliente. Basados en resultados tangibles, es una ventaja o beneficios de Scrum, el cliente estable sus criterios indicando el valor que lo aporta los requisitos del proyecto y comprueba de manera regular si va cumpliendo las expectativas, también el cliente puede tomar decisiones informadas a partir de resultados. teniendo estos resultados puede dirigir en iteración a iteración hacia la meta.

Equipos auto dirigidos y auto organizados. Son auto organizados y multifuncional, ya que eligen la mejor forma de llevar a cabo su trabajo y no son dirigidos por terceras personas o externas al equipo. Los equipos multifuncionales tienen todas las competencias necesarias para llevar a cabo el trabajo sin depender de otras personas que no son parte del equipo.

5.3. DEFINICIÓN OPERACIONAL DE LAS VARIABLES DE ESTUDIO

VARIABLE DE ESTUDIO

X: Sistema de Información de Recaudación Tributaria

INDICADORES

X1: Impuesto Predio Urbano

X2: Impuesto Predio Rural

VARIABLE DE ESTUDIO

Y: Jira Scrum Agile Software

INDICADORES

Y1: Gestión regular de las expectativas del cliente

Y2: Equipos auto dirigidos y auto organizados

CAPÍTULO VI METODOLOGÍA DE LA INVESTIGACIÓN

6.1. TIPO Y NIVEL DE INVESTIGACIÓN

TIPO DE INVESTIGACIÓN

En los estudios observacionales no existe intervención de ningún tipo por parte del investigador, de manera que los datos observados y la información consignado refleja la evolución natural de los eventos (Supo, 2012). Por esta consideración **la investigación es de tipo observacional.**

Los estudios retrospectivos utilizan datos que se obtienen de registros preexistentes, datos que provienen de mediciones en donde el investigador no tuvo participación alguna. A este tipo de información se le suele llamar datos secundarios (Supo, 2012). Por esta consideración **la investigación es de tipo retrospectivo.**

En un estudio transversal todas las variables (incluyendo la variable de estudio) son medidas en una sola ocasión bajo esta condición, si realizamos comparaciones entre estas mediciones se les suele llamar entre muestras independientes, aunque el nombre correcto sería entre grupos independientes (Supo, 2012). Por esta consideración **la investigación es de tipo transversal.**

NIVEL DE INVESTIGACIÓN

Los estudios descriptivos buscan especificar las propiedades, las características y los perfiles de personas, grupos, comunidades, procesos, objetos o cualquier otro fenómeno que se someta a un análisis. Es decir, únicamente pretenden medir o recoger información de manera independiente o conjunta sobre los conceptos o las variables a las que se refieren, esto es, su objetivo no es indicar como se relacionan estas (Hernández, et al, 2014). Por esta consideración el **nivel de investigación es Descriptiva.**

6.2. DISEÑO DE LA INVESTIGACIÓN

Los diseños no experimentales son estudios que se realizan **sin manipular deliberadamente variables.** Es decir, se trata de estudios en los que no hacemos variar en forma intencional las variables independientes para ver su efecto sobre

otras variables. Lo que hacemos en la investigación no experimental es observar fenómenos tal como se dan en su contexto natural, para analizarlos (Hernández et al., 2014). Por tanto, el diseño de **investigación es no experimental**.

Los diseños de investigación transversal recolectan datos en un solo momento, en un tiempo único. Su propósito es describir variables y analizar su incidencia e interrelación en un momento dado. Es como tomar una fotografía de algo que sucede (Hernández et al., 2014). Por esta consideración **la investigación es transversal**.

La investigación está enmarcada en el **diseño no experimental**, puesto que el caso de estudio no amerita la manipulación de las variables. Asimismo, se ha considerado el **diseño transversal**, ya que la recolección de datos se realizará en un solo momento.

6.3. POBLACIÓN Y MUESTRA

A. POBLACIÓN

La población de estudio estará compuesta por todas las metodologías ágiles de desarrollo y gestión de software ágil.

B. MUESTRA

Se tomó como muestra el Jira Scrum Agile Software para la gestión de desarrollo de software ágil.

6.4. TÉCNICAS E INSTRUMENTOS PARA RECOLECTAR INFORMACIÓN

A. TÉCNICAS

Según (Bernal, C., 2006), la investigación descriptiva se soporta principalmente en técnicas como la encuesta, la entrevista, la observación y la revisión documental. En definición en nuestra investigación se utiliza la técnica de entrevista tomando como cuenta las reuniones que se da con el representante (secretaria) de la Municipalidad Distrital de Santillana y también se utilizó el análisis documental de parte legal y jira Scrum existentes en el mercado de desarrollo de software para obtener un logro comparativo acerca del estudio.

B. INSTRUMENTOS

Según el apartado anterior, el instrumento guía es la entrevista de las reuniones que se desarrolla con el personal representante de la Municipalidad Distrital de Santillana, con el fin de generar información adecuada para el marco de trabajo del desarrollo del sistema de información de recaudación tributaria durante el tiempo que dura, la generación de información se encuentra en el apartado del anexo 1, anexo 2 y anexo 3. La revisión de los documentos legales se encuentra en el anexo 4 y anexo 5.

6.5. HERRAMIENTAS PARA EL DESARROLLO DE SISTEMA INFORMÁTICO

Las herramientas tecnológicas que se utilizan en este proyecto de investigación se da acuerdo a la necesidad de alcanzar el propósito, tomando en cuenta la facilidad y rapidez en su desarrollo, que facilite el uso de la complejidad del proyecto, el desarrollo del proyecto aplicando estándares y nos permita utilizar técnicas para asegurar la información crítica. Se ha seleccionado las tecnologías según la tabla 4.

Tabla 4

Herramientas tecnológicas para el desarrollo de sistemas Informáticos

SOFTWARE	FABRICANTE	SERVICIO
WINDOWS 10 ENTERPRISE	Microsoft Corporation	Es la versión del sistema operativo Windows para escritorio, ofrece el marco para instalar las herramientas de desarrollo.
JIRA SOFTWARE	Atlassian	Jira es una herramienta en línea con un marco de trabajo para la administración de las tareas de un proyecto, seguimiento de errores, incidencias y gestión operativa del proyecto.
NETBEANS	Apache Software Foundation ,Oracle	Es una instrumento de medio de fructificación de carta abierto,

	Corporation y Sun Microsystems	permite que las aplicaciones sean desarrolladas a seccionar de un universalismo de componentes de llamados módulos, un módulo es un archivo java que está compuesto por clases que esto unánimemente interactúa con los APIs de Netbeans y un archivo manifest file que lo identifica como módulos.
JAVA	Sun Microsystems y Oracle Corporation	Lenguaje de programación de efecto general, concurrente, encauzado a objetos, que fue diseñado específicamente para favor tan pocas dependencias de implementación como aexterior posible.
JAVAFX	Oracle Corporation y Sun Microsystems	En un totalidad de pertrechos de gráficos de utensilios que permite a los desarrolladores diseñar, crear, probar, higienizar e implementar aplicaciones de cliente enriquecido que operan de guisa consciente en diversas plataformas.
MYSQL	MySQL AB, Sun Microsystems y Oracle Corporation	Es un sistema de tratado de almohadilla de datos relacionales de código abierto, con un dechado cliente – servidor. RDBMS es un programa o servicio ajado para difundir y mandar almohadilla de datos basados en dechado relacional.

Nota. Fuente elaboración propia.

6.6. TÉCNICAS PARA APLICAR JIRA SCRUM

Revisando el marco teórico del marco de trabajo Scrum desarrollado en el capítulo IV, se formula las actividades y artefactos para el desarrollo del sistema de Información Recaudación Tributaria de la Municipalidad Distrital de Santillana.

Tabla 5

Actividades y Artefactos

TIPO	ACTIVIDAD/ ARTEFACTO	DESCRIPCIÓN	RESPONSABILIDAD/ PARTICIPANTES	DURACIÓN
Artefacto	Backlog de producto	Lista de características (historias de usuarios) priorizadas (por valor de negocio o riesgo) del producto.	(R) producto Owner Scrum Master	4hrs
Artefacto	Release RoadMap	Es un plan de alto nivel que indica lo que vamos a estar entregando y cuando (cantidad de reléase y tiempo).	Producto Owner Scrum Master Equipo de Desarrollo	
Actividad	Planificación del sprint	Se seleccionan las características a implementar en un sprint, se estima tiempo y esfuerzo (planning poker) y se adquiere el compromiso de realizarlo.	(R) Producto Owner Equipo de Desarrollo Scrum Master	4 Hrs. Selección 4 Hrs. Estimación
Artefacto	Sprint Backlog	Es una lista de tareas estimada de cada característica a implementar, cada tarea tiene un dificultad (Fibonacci) y una duración	(R) Equipo de Desarrollo	

Actividad	Ejecución del sprint	Tareas necesarias para implementar las características del sprint.	(R) Equipo de Desarrollo Producto Owner	4 semanas máximo por sprint
Actividad	Daily Scrum	Es una reunión diaria de revisión de las ejecuciones de las tareas.	(R) Scrum Master Producto Owner Development Team	10 a 15 minutos por día
Artefacto	Incremento del producto	Al finalizar la ejecución del sprint se obtiene una mejora o incremento del producto que es potencialmente entregable al cliente.	(R) Development Team	
Actividad	Sprint Review	El equipo y los stakeholders revisan el producto entregado y se recoge el feedback.	Development Team Producto Owner Scrum Master Stakeholders	4 Hrs. Máximo
Actividad	Sprint Retrospectiva	Acá se revisa el proceso Scrum con la finalidad de mejorar el proceso en sí.	Development Team Product Owner Scrum Master	3 Hrs. Máximo
Artefacto	Entrega del producto final	Tras una serie de sprint se finaliza la implementación del producto.	Development Team Producto Owner Scrum Master	

Nota. Recuperado de Rubín. (2013). Actividades y Artefactos de Jira Scrum.

CAPÍTULO VII

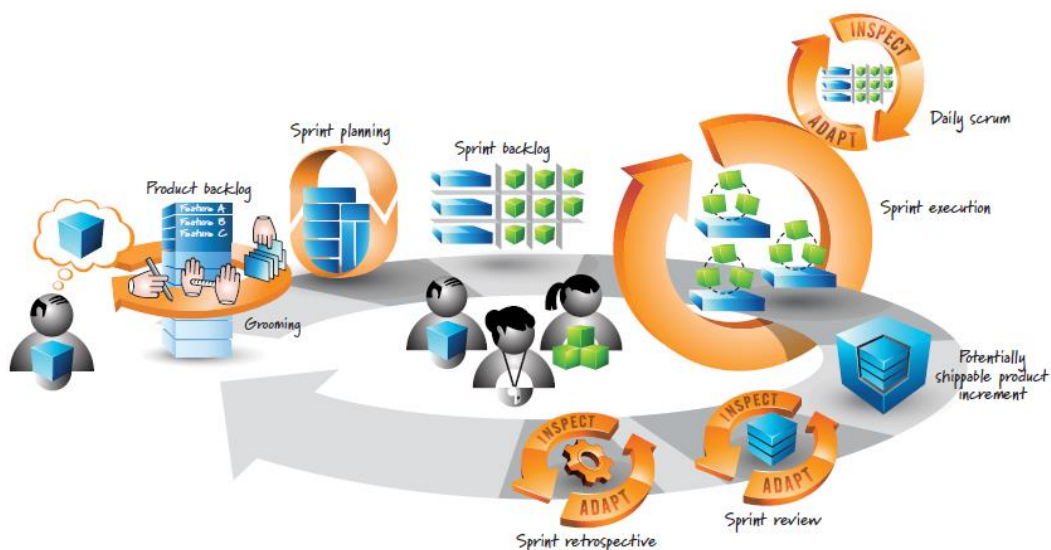
ANÁLISIS Y RESULTADOS DE LA INVESTIGACIÓN

Para este capítulo se aplicará el marco de trabajo Scrum, para el desarrollo ágil de software del sistema de información de recaudación tributaria de la Municipalidad Distrital de Santillana, con la ayuda de jira software. Como bien sabemos Scrum es un conjunto de buenas prácticas para un proyecto, trabaja en equipo o colaborativamente, que permite iteraciones cortas de tiempo para la entrega del producto a la municipalidad.

En resumen, Scrum está basado en actividades y artefactos como ilustra en la siguiente figura 26 donde los elementos azules representan el alcance, las verdes tareas y los elementos naranjas representan proceso, en consecuencia, durante este capítulo se analizará y se encontrará resultados a la investigación, aplicando todas las fases de actividades y artefactos de Scrum.

Figura 26

Actividades y artefactos de SCRUM



Nota. Recuperado de Kenneth, S. (2012, p. 17). Essential Scrum. A practical guide to the most popular agile process.

7.1. FASE N° 1: PILA DE PRODUCTO (PRODUCT BACKLOG)

Esta fase es el corazón del proyecto que se va investigar, por lo que se interviniera en el marco de trabajo de Scrum del proyecto sistema de información de recaudación tributaria de la Municipalidad Distrital de Santillana, con ayuda

software jira. Por una parte, la empresa que acoge este marco de trabajo para este proyecto es Ennova m&f s.a.c. encabezado por el Sr. Yonny Huarancca Ñaupari.

7.1.1. DESCRIPCIÓN DEL PROYECTO

El sistema de recaudación tributación de la Municipalidad Distrital de Santillana, es el software que se va implementar con el marco de trabajo de Scrum con ayuda de jira software. Teniendo el conocimiento vamos a describir el contenido el proyecto desde que se analizó los problemas al entorno de la necesidad de la construcción del sistema.

Cuando se empezó analizar los requisitos para construcción del sistema se verifico varias pautas por lo que se tomó una decisión de implementar un nuevo sistema tanto para la zona rural y urbano.

Entre las pautas son:

- Existía un sistema llamado “sysautovaluo”, donde no cumplía estándares tanto política, seguridad, normas del estado, etc. como en Tabla 6 que muestra sysautovaluo y el nuevo sistema que construyo. por otro lado, analizando el base de datos se encontró un conjunto de incoherencias con el cliente (poblador que pagaba de su autovaluo), por lo que algunos montos eran muy altos o bien bajos. El sistema “sysautovaluo” como a la vez en los reportes no encajaban los letras y números en los textbox.
- El sysautovaluo ya siendo obsoleto y dejando de funcionar, más de tres años no se recaudaba los tributos Municipales de la población que comprende los conceptos de impuesto predial de autovaluo urbana y rural.
- Conociendo la falta viabilidad de no tener un sistema de recaudación tributaria, lo que ocasionaba sin conocer el autovaluo de sus propiedades es que muchos pobladores tenían la dificultad de entrar en bonos de techo propio del estado, teniendo que movilizarse a otros distritos para que puedan analizar sus propiedades así generando pérdida de tiempo.
- La Municipalidad Distrital de Santillana teniendo como preocupación la falta de ingreso a la caja chica de la municipalidad.

Tabla 6

Evaluación del sistema “sysautovaluo” y el sistema construido

EVALUACIÓN	ESTANDAR	
	Sysautovaluo (lo que se tenía el sistema)	lo que se tendrá al construir el sistema
CODIGO FUENTE		
El código fuente se tiene	NO	SI
El código fuente se puede mejorar	NO	SI
El código fuente cuenta con estándares de seguridad	NO	SI
Tiene documentación o manuales	NO	SI
BASE DE DATOS		
La Base de Datos se tiene	SI	SI
Se puede mejorar la Base de Datos	NO	SI
Cumple con los estándares de Desarrollo	NO	SI
Se tiene Usuario y contraseña al acceso al Base de Datos	NO	SI
Se puede recuperar la base de datos de todo los registros y pagos de la población	NO	SI
El Base de Datos está desarrollado en una versión actual de un gestor de Base de datos	NO	SI
ESTANDARES DE NORMA		
Resolución Ministerial N° 370-2018-Vivienda	NO	SI
Decreto supremo N° 153-2004-EF Valores totales de predio	SI	SI
Resoluciones ministeriales N° 126-2007-Vivienda y N° 266-2012-Vivienda de la tabla de depreciación por antigüedad y estado de conservación según el material estructural predominante	SI	SI
IMPUESTO PREDIAL		
Registro de Declaraciones Juradas	SI	SI
Predios Urbanos y Rústicos	SI	SI
Cálculo de Base Imponible	SI	SI
Mantenimiento de Tablas para el cálculo	NO	SI
Cálculo y recalcado automático	NO	SI
Información confiable de los contribuyentes y su Predios	NO	SI
ARBITRIOS MUNICIPALES		
Determinación de Arbitrios Municipales	SI	SI
Calculo por fecha de declaración	SI	SI
Mantenimiento de Tablas para el cálculo	NO	SI
Cálculo y recalcado automático	NO	SI
FISCALIZACIÓN		
Registro de la Base de Datos de la Fiscalización ejecutada sobre determinados contribuyentes	NO	SI

Cálculo automático obteniendo las diferencias en las Cuentas Corrientes	NO	SI
Resoluciones de Determinación	SI	SI

Nota. Fuente elaboración propia.

META

Teniendo todo los estándares y políticas, desarrollar un sistema de información de recaudación tributaria de la Municipalidad Distrital de Santillana, un software de aplicación de escritorio para automatizar la información del impuesto predial tanto de la zona rural y urbana (rustica), utilizando tecnologías como Mysql, spring boot, lenguaje java, javafx, programación orientados a objetos(POO) y patrón de arquitectura de software MVC, durante tres meses establecidas.

OBJETIVOS PRINCIPAL

Desarrollar un software de aplicación de escritorio para automatizar la información del impuesto predial, mediante técnicas e instrumentos como un gestor de base datos Mysql, spring boot, lenguaje javo, javafx, programación orientados a objetos(POO) y patrón de arquitectura de software MVC, para la recaudación de tributo municipal de la Municipalidad Distrital de Santillana, Provincia de Huanta.

OBJETIVOS SECUNDARIA

Requerimiento, Diseñar, Implementar, Verificación y Mantenimiento para automatizar la información del impuesto predio urbano e impuesto predio rural, para recaudación de tributo municipal, con el fin de obtener concepto de pago al impuesto predial en impuesto anual e impuesto trimestral, resultado del valor total del terreno y total de autovaluo del contribuyente y los reportes.

TECNOLOGÍAS DE UTILIZACIÓN

Ganando la licitación a mediados de 2019, la empresa Ennova m&f s.a.c. empieza organizar y a la vez utiliza las siguientes tecnologías.

- Jira software
- Netbeans

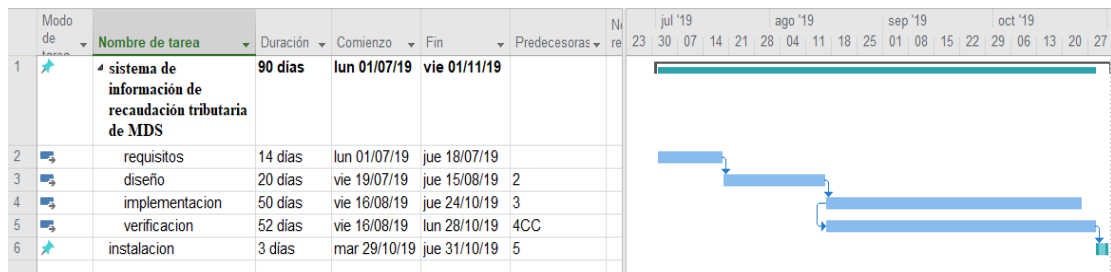
- Mysql
- Java 8.0.
- Javajx
- Windows 10 Enterprise

CRONOGRAMA DEL PROYECTO

En el siguiente Figura 27 se observa todo el cronograma del proyecto del “sistema de información de recaudación tributaria de la Municipalidad Distrital de Santillana”.

Figura 27

Cronograma de tareas del proyecto



Nota. Fuente elaboración propia, generado con Jira Software que nos muestra todas las tareas del proyecto.

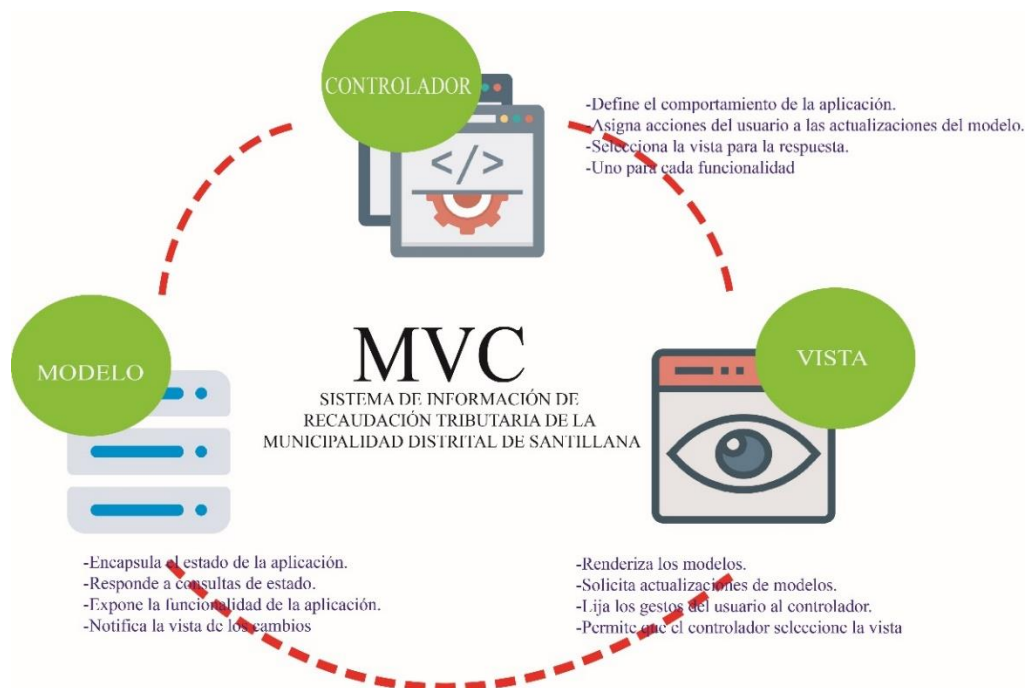
De la Figura 27 se observa que se empieza a realizar el proyecto lunes 01 de julio del 2019 hasta el jueves 31 de noviembre del 2019, todo según la secuencia de los procesos del desarrollo del software como recaudando requisitos, diseño, implementación, verificación y por último la instalación en el área de tesorería de la mismo Municipalidad Distrital de Santillana. Con una duración de 3 meses según estipula la contrata.

ARQUITECTURA DEL PROYECTO

El sistema utilizara una arquitectura de patrón de modelo – vista – controlador (MVC), como en la Figura 28 ilustra.

Figura 28

MVC del sistema de información de recaudación tributaria de la Municipalidad Distrital de Santillana



Nota. Fuente elaboración propia, explicaciones de modelo, vista y controlador.

De la Figura 28 se puede tomar las siguiente:

- En el lado del modelo de nuestra arquitectura del sistema, se puede observar que tiene que interrelacionar tanto con la vista (modelo de actualización) y el controlador (modelo de consulta), tomando los parámetros cumple las tareas de encapsular los parámetros de la aplicación, responde consultas del estado, expone las funcionalidades de la aplicación y notifica las vistas de los cambios.
- En el lado de vista teniendo como evento de entrada y como salida de controlador (controlador de vista) y modelo (modelo de actualización), cumpliendo las tareas de renderizar los modelos, solicita la actualización de modelos, fija los gastos de del usuario al controlador y permite que el controlador seleccione la vista.
- Y por último en lado del controlador re relaciona con vista (controlador de vista) y modelo (modelo de actualización), con las tareas de definir el comportamiento de la aplicación, asigna acciones de usuario a las actualizaciones del modelo, selecciona la vista para la respuesta y uno para cada funcionalidad.

REQUISITOS FUNCIONALES

Tabla 7

Requisitos funcionales

N° Requisito	Requisitos
FUNCIONALES	
1	La secretaria de la Municipalidad y el administrador deben ser capaz de acceder al sistema Autovaluo
2	El sistema debe ser capaz de registrar nuevo usuario de sistema
3	El sistema debe ser capaz de registrar nuevo contribuyente
4	El sistema debe ser capaz de crear una propiedad de zona urbana
5	El sistema debe ser capaz de crear un propiedad de zona rural o rustico
6	El sistema debe de registrar predio urbano y predio rural
7	El sistema de debe de registrar los años de lo predios
8	El sistema debe generar lista de pago urbano y pago rural
9	El sistema debe de generar una búsqueda de contribuyentes
10	El sistema debe de generar un listado de propiedades
11	El sistema debe generar pago de impuesto predial
12	El sistema debe de generar reportes de urbano y rustico
13	El sistema debe ser capaz de generar una boleta de predio rural y urbano
14	El sistema debe ser capaz de generar una declaración jurada de Autovaluo

Nota. Fuente elaboración propia, listado de requisitos funcionales.

REQUISITOS NO FUNCIONALES

Tabla 8

Requisitos funcionales

N° Requisito	Requisitos
NO FUNCIONALES	
1	El sistema Autovaluo de cumplir el patrón de arquitectónico de modelo, control y vista.
2	El sistema Autovaluo debe adaptarse al marco de trabajo Jira Scrum Agile Software.
3	El sistema debe ser de fácil uso por la secretaria de la Municipalidad
4	El sistema debe ser flexible al funcionamiento de 24/7
5	El sistema debe ser desarrollado en java, javafox, mysql, spring, mavel y en el metodología scrum
6	El sistema Autovaluo debe ser desarrollado para usar en múltiples sistemas operativos.

Nota. Fuente elaboración propia, listado de requisitos no funcionales.

GLOSARIO DE TERMINO

- Secretaria
- Contribuyente
- Propiedad
- Predio urbano
- Predio rural o rustico
- Impuesto predial
- Boleta
- Declaración jurada de Auotovaluo

7.1.2. CARTERA DE PRODUCTOS

El propietario del producto la Municipalidad con representación del secretario colaboran con el Scrum master y el equipo de trabajo, para reunir y definir los elementos de carteta de productos. Luego el producto Owner se asegura que los elementos se colocan de forma secuencial de mayor a menor valor. Estimar y priorizar se conoce como preparación. Para definir los elementos y así convertirlos en historias, vamos utilizar una secuencia de buenas prácticas de Jira Scrum en lo siguiente.

7.1.2.1. TABLA AGÍL SCRUM

Para facilitación de la investigación debemos plantear físicamente un tablero de backlog, para así ver la distribución de Release (prioridad de lanzamientos), la división correcta entre tareas e historial de usuario, como se muestra en la Figura 29.

Figura 29

Tabla ágil Scrum



Nota. Fuente elaboración propia.

En la Figura 29 se muestra nuestros elementos de trabajo ya que estos pueden variar al avance del proyecto, tenemos en la parte superior de color rosa 7 elementos (historia de usuarios), en tanto los historias de usuarios pueden subdividirse en historias de usuarios más pequeñas llamadas tareas (incidencias) que constan de 32 para nuestra investigación, a más detalle cómo se realizó con el dueño del producto, Scrum master y el equipo de trabajo se muestra en el anexo 3.

7.1.2.2. PANEL DE EPICS

Los epics en Jira Scrum es una gran cantidad que se puede desglosar en varias historias de usuario de menor tamaño llamadas tareas en Jira Scrum se llama incidencias. Un epic es un historial de usuario, pero en mayor complejidad de donde se desglosa a tamaño más adecuados o pequeña para ser gestionada.

Figura 30

Epics



Nota. Fuente elaboración propia, generado con Jira Software.

Las incidencias de los epics de la Figura 30 nos ayuda a gestionar en los parámetros incidencias, terminadas, sin estimar y estimar. Sobre todo, para relacionar con los historiales desglosado (Figura 33), para una buena gestión ágil. Nuestro epics en la investigación son: login de sistema, registro de usuario, contribuyente urbano, contribuyente rural o rustico, caja, reportes y normas legales.

7.1.2.3. VERSIONES DE BACKLOG

Son lanzamientos de más prioridad a menos para hacer seguimientos de las entregas de proyecto, estableciendo fecha de inicio y fecha final de entrega Tabla 9 también se muestra en la Figura 31 en tres niveles nuestro proyecto o reléase para lanzamiento en los tiempos establecidos como Release - SIRT Básico desde 19/08/2019 al 13/09/2019, Release 2 – SIRT lo necesario desde 15/09/2019 al 11/10/2019 y Release 3 – menos básico desde 14/10/2019 al 24/10/2019. En la Figura 31 y Tabla 10 nos muestra la asignación de prioridades según el avance.

Tabla 9

Tabla de Release

RELEASE	Fecha de Inicio	Fecha de Entrega
---------	-----------------	------------------

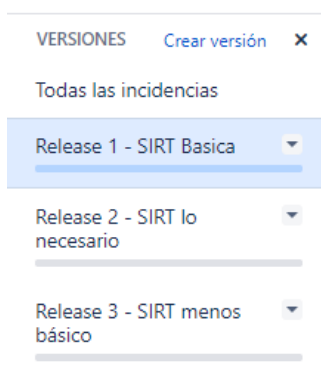
Release 1 - SIRT Básico	19/08/2019	13/09/2019
Release 2 – SIRT Lo necesario	15/09/2019	11/10/2019
Release 3 – menos básico	14/10/2019	24/10/2019

Nota. Fuente elaboración propia.

Las versiones también constan de incidencias, terminados y no terminados, sin estimar y estimar. Cada versión consta con un conjunto de historial de usuario según la prioridad como se muestra en la Figura 33.

Figura 31

Lanzamiento de Backlog



Nota. Fuente elaboración propia, generado con Jira Software.

En la Tabla 10 se visualiza las versiones del proyecto en las fechas establecidos desde mayor prioridad a menos, con la condición de publicarse. La realización de versión en jira Scrum software recae al grupo como Scrum team, Scrum Master y producto Owner.

Tabla 10

Versiones de Backlog

Versión :	Estado	Progreso	Fecha de inicio :	Fecha de publicación :	Descripción :	
Release 3 - SIRT menos básico	SIN PUBLIC...	<div style="width: 100%;"></div>	14 oct. 2019	24 oct. 2019	menos básico para publicar	...
Release 2 - SIRT lo necesario	SIN PUBLIC...	<div style="width: 100%;"></div>	15 sept. 2019	11 oct. 2019	lo necesario para publicar	...
Release 1 - SIRT Basica	SIN PUBLIC...	<div style="width: 100%;"></div>	19 ago. 2019	13 sept. 2019	lo mínimo necesaria para publicar	...

Nota. Fuente elaboración propia, generado con Jira Software.

7.1.3. LISTADO DE LA PILA DE PRODUCTOS

Utilizando el marco de trabajo Scrum en jira software, se hace un listado de actividades para el buen funcionamiento del desarrollo del sistema de recaudación tributaria de la Municipalidad Distrital de Santillana, como a la vez la organización de los involucrados del desarrollo.

Figura 32

Lista de Historias de Usuarios

Backlog 32 incidencias Crear sprint ...

■ HU01 - LOGUEARSE	SIS-8	↑	-
■ HU02 - SALIR DE LOGIN	SIS-9	↑	-
■ HU03 - REGISTRAR USUARIO	SIS-10	↑	-
■ HU04 - MODIFICAR USUARIO	SIS-11	↑	-
■ HU05 - ELIMINAR USUARIO	SIS-12	↑	-
■ HU06 - VISUALIZACIÓN DE USUARIO	SIS-13	↑	-
■ HU07 - AGREGAR NUEVO CONTRIBUYENTE URBANO	SIS-14	↑	-
■ HU08 - BUSCADOR DE CONTRIBUYENTE URBANO	SIS-15	↑	-
■ HU09 - EDITAR/ELIMINAR CONTRIBUYENTE URBANO	SIS-16	↑	-
■ HU10 - ABRIR UN CONTRIBUYENTE URBANO	SIS-17	↑	-
■ HU11 - VISUALIZAR ABRIR PERSONA URBANA	SIS-18	↑	-
■ HU12 - CREAR PROPIEDAD DE ZONA URBANA	SIS-19	↑	-
■ HU13 - CREAR AÑO PARA PREDIO URBANO	SIS-20	↑	-
■ HU14 - CREAR NUEVO PREDIO URBANO	SIS-21	↑	-
■ HU15 - GENERAR LISTA DE PAGO URBANO	SIS-22	↑	-
■ HU16 - CERRAR/MINIMIZAR LA VISTA ABRIR PERSONA URBANA	SIS-23	↑	-
■ HU17 - AGREGAR UN NUEVO CONTRIBUYENTE RURAL O RUSTICO	SIS-24	↑	-
■ HU18 - BUSCADOR DE CONTRIBUYENTE RURAL O RUSTICO	SIS-25	↑	-
■ HU19 - EDITAR/ELIMINAR CONTRIBUYENTE DE ZONA RURAL O RUSTICO	SIS-26	↑	-
■ HU20 - ABRIR UN CONTRIBUYENTE DE DE LA ZONA RURAL O RUSTICO	SIS-27	↑	-
■ HU21 - VISUALIZAR LA VENTANA DE ABRIR PERSONA RURAL O RUSTICO	SIS-28	↑	-
■ HU22 - CREAR PROPIEDAD DE ZONA RURAL O RUSTICO	SIS-29	↑	-
■ HU23 - CREAR AÑO PARA PREDIO RURAL O RUSTICO	SIS-30	↑	-
■ HU24 - CREAR NUEVO PREDIO RURAL O RUSTICO	SIS-31	↑	-
■ HU25 - LISTA DE PAGO DE ZONA RURAL O RUSTICO	SIS-32	↑	-
■ HU26 - CERRAR/MINIMIZAR LA VISTA ABRIR PERSONA URBANO O RUSTICO	SIS-33	↑	-
■ HU27 - BUSCAR LISTA DE CONTRIBUYENTE	SIS-34	↑	-
■ HU28 - VISUALIZACIÓN DEL LISTADO DE PROPIEDAD	SIS-35	↑	-
■ HU29 - GENERAR PAGO DEL IMPUESTO PREDIAL	SIS-36	↑	-
■ HU30 - VISUALIZACIÓN DE REPORTES URBANO Y RURAL	SIS-37	↑	-
■ HU31 - VISUALIZACIÓN DE DOCUMENTOS	SIS-38	↑	-
■ HU32 - VISUALIZACIÓN DE AYUDA	SIS-39	↑	-

Nota. Fuente elaboración propia, generado con Jira Software.

En la Figura 32 vemos un backlog de 32 de incidencias, también llamado historial de usuarios, estos son actividades del sistema información recaudación tributaria de la Municipalidad Distrital de Ayacucho. Está formado según la importancia del lanzamiento (Release). Esta lista de Historias de Usuarios se centra en el valor del cliente (representante de la Municipalidad de Santilla) y la fuerte intención de fomentar la comunicación para un mejor desarrollo del sistema Autovaluo y el trabajo en equipo.

7.1.4. RELACIÓN DE HISTORIAL DE USUARIO ENTRE EPICS Y VERSIONES

Nos facilita que cada epic tenga un conjunto de historias de usuarios (tareas) o una actividad tenga un conjunto de sub actividades conformado en prioridades, versiones de Release 1, Release 2 y Release 3, establecidos con las fechas de inicio y entrega.

El principal objetivo en esta etapa es definir bien nuestro backlog que consta principalmente de las historias de usuarios (tareas) como loguearse, salir de login, registrar usuario, modificar usuario, eliminar usuario, visualización de usuario, agregar nuevo contribuyente, buscador de contribuyente, editar y eliminar contribuyente urbano, abrir un contribuyente urbano, visualizar abrir persona urbana, crear propiedad de zona urbana, crear año para predio urbano, crear un nuevo predio urbano, generar lista pagos urbanos, cerrar y minimizar la vista abrir urbano, agregar nuevo contribuyente rural, buscador de contribuyente rural, editar y eliminar contribuyente rural, abrir un contribuyente rural, visualizar abrir persona rural, crear propiedad de zona rural, crear año para predio rural, crear un nuevo predio rural, generar lista pagos rural, cerrar y minimizar la vista abrir rural, buscar lista de contribuyentes, visualización del listado de propiedad, generar pago del impuesto predial, visualización de reportes urbanos y rural, visualización de documentos y visualización de ayuda como se muestra en Figura 33 perteneciendo cada uno a un reléase fecha de inicio y fecha final o versiones, como a la vez perteneciendo a un epic que es más que una historia usuario primordial.

Figura 33

Lista de Historias de Usuario con asignación de epics y release

Backlog 32 incidencias Crear sprint ...

■	HU01 - LOGUEARSE	RELEASE 1 - SIRT BA...	login del sistema	SIS-8	↑	-
■	HU02 - SALIR DE LOGIN	RELEASE 1 - SIRT BA...	login del sistema	SIS-9	↑	-
■	HU03 - REGISTRAR USUARIO	RELEASE 1 - SIRT BA...	registrador de usuarios	SIS-10	↑	-
■	HU04 - MODIFICAR USUARIO	RELEASE 1 - SIRT BA...	registrador de usuarios	SIS-11	↑	-
■	HU05 - ELIMINAR USUARIO	RELEASE 1 - SIRT BA...	registrador de usuarios	SIS-12	↑	-
■	HU06 - VISUALIZACIÓN DE USUARIO	RELEASE 2 - SIRT LO ...	registrador de usuarios	SIS-13	↑	-
■	HU07 - AGREGAR NUEVO CONTRIBUYEN	RELEASE 1 - SIRT BA...	contribuyente urbano	SIS-14	↑	-
■	HU08 - BUSCADOR DE CONTRIBUYENTE	RELEASE 1 - SIRT BA...	contribuyente urbano	SIS-15	↑	-
■	HU09 - EDITAR/ELIMINAR CONTRIBUYEN	RELEASE 1 - SIRT BA...	contribuyente urbano	SIS-16	↑	-
■	HU10 - ABRIR UN CONTRIBUYENTE URB.	RELEASE 1 - SIRT BA...	contribuyente urbano	SIS-17	↑	-
■	HU11 - VISUALIZAR ABRIR PERSONA UR	RELEASE 2 - SIRT LO ...	contribuyente urbano	SIS-18	↑	-
■	HU12 - CREAR PROPIEDAD DE ZONA UR	RELEASE 2 - SIRT LO ...	contribuyente urbano	SIS-19	↑	-
■	HU13 - CREAR AÑO PARA PREDIO URBA	RELEASE 2 - SIRT LO ...	contribuyente urbano	SIS-20	↑	-
■	HU14 - CREAR NUEVO PREDIO URBANO	RELEASE 2 - SIRT LO ...	contribuyente urbano	SIS-21	↑	-
■	HU15 - GENERAR LISTA DE PAGO URBAN	RELEASE 2 - SIRT LO ...	contribuyente urbano	SIS-22	↑	-
■	HU16 - CERRAR/MINIMIZAR LA VISTA AÍ	RELEASE 3 - SIRT ME...	contribuyente urbano	SIS-23	↑	-
■	HU17 - AGREGAR UN NUEVO CONTR	RELEASE 1 - SIRT BA...	contribuyente rural o r...	SIS-24	↑	-
■	HU18 - BUSCADOR DE CONTRIBUYE!	RELEASE 1 - SIRT BA...	contribuyente rural o r...	SIS-25	↑	-
■	HU19 - EDITAR/ELIMINAR CONTRIBU	RELEASE 1 - SIRT BA...	contribuyente rural o r...	SIS-26	↑	-
■	HU20 - ABRIR UN CONTRIBUYENTE I	RELEASE 1 - SIRT BA...	contribuyente rural o r...	SIS-27	↑	-
■	HU21 - VISUALIZAR LA VENTANA DE	RELEASE 2 - SIRT LO ...	contribuyente rural o r...	SIS-28	↑	-
■	HU22 - CREAR PROPIEDAD DE ZONA	RELEASE 2 - SIRT LO ...	contribuyente rural o r...	SIS-29	↑	-
■	HU23 - CREAR AÑO PARA PREDIO RU	RELEASE 2 - SIRT LO ...	contribuyente rural o r...	SIS-30	↑	-
■	HU24 - CREAR NUEVO PREDIO RURA	RELEASE 2 - SIRT LO ...	contribuyente rural o r...	SIS-31	↑	-
■	HU25 - LISTA DE PAGO DE ZONA RU!	RELEASE 2 - SIRT LO ...	contribuyente rural o r...	SIS-32	↑	-
■	HU26 - CERRAR/MINIMIZAR LA VIST/	RELEASE 3 - SIRT ME...	contribuyente rural o r...	SIS-33	↑	-
■	HU27 - BUSCAR LISTA DE CONTRIBUYENTE	RELEASE 1 - SIRT BA...	caja	SIS-34	↑	-
■	HU28 - VISUALIZACIÓN DEL LISTADO DE PROPIEDAD	RELEASE 1 - SIRT BA...	caja	SIS-35	↑	-
■	HU29 - GENERAR PAGO DEL IMPUESTO PREDIAL	RELEASE 1 - SIRT BA...	caja	SIS-36	↑	-
■	HU30 - VISUALIZACIÓN DE REPORTES URBANO Y RUF	RELEASE 1 - SIRT BA...	reportes	SIS-37	↑	-
■	HU31 - VISUALIZACIÓN DE DOCUMENTOS	RELEASE 1 - SIRT BA...	normas legales	SIS-38	↑	-
■	HU32 - VISUALIZACIÓN DE AYUDA	RELEASE 1 - SIRT BA...	normas legales	SIS-39	↑	-

Nota. Fuente elaboración propia, generado con Jira Software.

7.2. FASE N° 2: PLANIFICACIÓN DE SPRINT (SPRINT PLANNING)

Conociendo que backlog está bien definido, ahora debemos hacer un conjunto de procesos de spring para llevar las reuniones de planificación para cumplir según las fechas establecidas para la entrega, como a la vez atender las dudas y comprobar que el equipo comprende y comparte su objetivo, pero no antes definiendo cada actividad con su respectivo responsable, para esto se asigna en:

Producto Owner: Marcelino Centeno Quispe

Scrum Master: Yonny Huarancca Ñaupari.

Scrum Team:

- Elvis Teneo Morales
- Manrique Meneses Yaranga
- Papel Bautista Acuña
- Roberth Areste Palomino

7.2.1. RESERVA DE SPRINT (SPRINT BACKLOG)

Son todas las historias de usuarios destacados de sistema de información de recaudación tributaria de la Municipalidad Distrital de Santillana, donde que el equipo de sprint se compromete trabajar durante el sprint actual. Cada tarea (historia de usuario) tiene un valor en puntos como se muestra en la Figura 34, Figura 36, Figura 38 y Figura 40 eso refleja la cantidad de trabajo que el equipo Scrum realiza lo necesario para completar esa tarea, a continuación, en Jira Scrum se describe a detalle en cada historia de usuario para que el equipo de trabajo desarrolle y la Municipalidad Distrital de Santillana se sienta satisfecho de lo que desea que sea haga como se muestra en siguiente Tabla 11, Tabla 12, Tabla 13, Tabla 14, Tabla 15, Tabla 16, Tabla 17, Tabla 18, Tabla 19, Tabla 20, Tabla 21, Tabla 22, Tabla 23, Tabla 24, Tabla 25, Tabla 26, Tabla 27, Tabla 28, Tabla 29, Tabla 30, Tabla 31, Tabla 32, Tabla 33, Tabla 34, Tabla 35, Tabla 36, Tabla 37, Tabla 38, Tabla 39, Tabla 40, Tabla 41 y Tabla 42.

Tabla 11

Detalle de Historia de Usuario: HU01 - LOGUEARSE

[SIS-8] HU01 - LOGUEARSE Creada: 19/abr/20 Actualizada: 21/abr/20 Resuelta: 21/abr/20			
Estado:		Finalizada	
Proyecto:		SistemaRT	
Componentes:		Ninguno	
Versiones afectadas:		Ninguno	
Versiones corregidas:		Release 1 - SIRT Basica	
Tipo:	Historia	Prioridad:	High
Informador:	YONNY HUARANCCA ÑAUPARI	Responsable:	YONNY HUARANCCA ÑAUPARI
Resolución:	Listo	Votos:	0

Etiquetas: Ninguno
 Estimación 1 day
 Restante:
 Tiempo 2 days
 Trabajado:
 Estimación 1 day
 original:
 Epic Link: [login del sistema](#)
 Sprint: Tablero Sprint 1

como: usuario de sistema.

quiero: ingresar al sistema

para: administrar el sistema

condición:

- la pantalla de login debe contener en la parte izquierda el logo de la municipalidad y su nombre de la institución.
- debe contener textbox de usuario y contraseña
- debe contener un botón de ingreso y salida

Nota. Fuente elaboración propia, generado con Jira Software. Generado a las Tue Apr 21 02:44:45 UTC 2020 por YONNY HUARANCCA ÑAUPARI usando JIRA 1001.0.0-SNAPSHOT#100125-sha1:af682b6c94104bf5538206af5af6eb285a90e8b7.

Tabla 12

Detalle de Historia de Usuario: HU02 - SALIR DE LOGIN

[SIS-9] HU02 - SALIR DE LOGIN Creada: 19/abr/20 Actualizada: 21/abr/20			
Estado:	En curso		
Proyecto:	SistemaRT		
Componentes:	Ninguno		
Versiones afectadas:	Ninguno		
Versiones corregidas:	Release 1 - SIRT Basica		
Tipo:	Historia	Prioridad:	Medium
Informador:	YONNY HUARANCCA ÑAUPARI	Responsable:	YONNY HUARANCCA ÑAUPARI
Resolución:	Sin resolver	Votos:	0
Etiquetas:	Ninguno		
Estimación	0 minutes		
Restante:			

Tiempo 1 day
 Trabajado:
 Estimación 6 hours
 original:
 Epic Link: [login del sistema](#)
 Sprint: Tablero Sprint 1

como: usuario de sistema

quiero: salir o cerrar la ventana de login

para: dar como finalizado la tarea

condición:

- la ventana de login debe contener en la parte izquierdo logo de la institución y su nombre.
- debe contener de usuario y contraseña
- debe contener botón de salida o cerrar

Nota. Fuente elaboración propia, generado con Jira Software. Generado a las Tue Apr 21 03:30:46 UTC 2020 por YONNY HUARANCCA ÑAUPARI usando JIRA 1001.0.0-SNAPSHOT#100125-sha1:af682b6c94104bf5538206af5af6eb285a90e8b7.

Tabla 13

Detalle de Historia de Usuario: HU03 - REGISTRAR USUARIO

[SIS-10] HU03 - REGISTRAR USUARIO Creada: 19/abr/20 Actualizada: 21/abr/20			
Estado:	En curso		
Proyecto:	SistemaRT		
Componentes:	Ninguno		
Versiones afectadas:	Ninguno		
Versiones corregidas:	Release 1 - SIRT Basica		
Tipo:	Historia	Prioridad:	Highest
Informador:	YONNY HUARANCCA ÑAUPARI	Responsable:	pabel.bautista.27
Resolución:	Sin resolver	Votos:	0
Etiquetas:	Ninguno		
Estimación Restante:	6 hours		
Tiempo Trabajado:	2 days		
Estimación original:	1 day		
Epic Link:	registrador de usuarios		
Sprint:	Tablero Sprint 1		
como: usuario de sistema			
quiero: crear un nuevo usuario.			

para: que pueda ingresar al sistema

condición:

- debe contener en la ventana acceso, un boton con nombre “nuevo “
- al hacer clic en el botón nuevo debe abrir un modal.
- en la ventana modal, debe contener textbox de los campos nombre y apellido, DNI, usuario y contraseña.
- debe existir un botón de guardar y cerrar

Nota. Fuente elaboración propia, generado con Jira Software. Generado a las Tue Apr 21 04:38:05 UTC 2020 por YONNY HUARANCCA ÑAUPARI usando JIRA 1001.0.0-SNAPSHOT#100125-sha1:a1a1592ef3306a126a3ffbb2e5ab6e40de1058d.

Tabla 14

Detalle de Historia de Usuario: HU04 - MODIFICAR USUARIO

[SIS-11] <u>HU04 - MODIFICAR USUARIO</u> Creada: 19/abr/20 Actualizada: 21/abr/20			
Estado:	Tareas por hacer		
Proyecto:	SistemaRT		
Componentes:	Ninguno		
Versiones afectadas:	Ninguno		
Versiones corregidas:	Release 1 - SIRT Basica		
Tipo:	Historia	Prioridad:	High
Informador:	YONNY HUARANCCA ÑAUPARI	Responsable:	pabel.bautista.27
Resolución:	Sin resolver	Votos:	0
Etiquetas:	Ninguno		
Estimación Restante:	1 day		
Tiempo Trabajado:	3 days		
Estimación original:	1 day		
Epic Link:	registrador de usuarios		
Sprint:	Tablero Sprint 1		

como: usuario de sistemas

quiero: modificar el usuario

para: la facilidad del usuario

condición:

- debe contener en la ventana acceso un botón con nombre “modificar “.
- al hace clic en el botón modificar debe abrir un modal.

- en la ventana modal debe existir un textbox con campos de nombre, DNI, usuario y contraseña. el usuario modifica y guarda o bien cancela con los botones respectivos.

Nota. Fuente elaboración propia, generado con Jira Software. Generado a las Tue Apr 21 04:53:23 UTC 2020 por YONNY HUARANCCA ÑAUPARI usando JIRA 1001.0.0-SNAPSHOT#100125-sha1:a1a1592ef3306a126a3ffbb2e5ab6e40de1058d.

Tabla 15

Detalle de Historia de Usuario: HU05 - ELIMINAR USUARIO

[SIS-12] HU05 - ELIMINAR USUARIO Creada: 19/abr/20 Actualizada: 21/abr/20			
Estado:	Tareas por hacer		
Proyecto:	SistemaRT		
Componentes:	Ninguno		
Versiones afectadas:	Ninguno		
Versiones corregidas:	Release 1 - SIRT Basica		
Tipo:	Historia	Prioridad:	Low
Informador:	YONNY HUARANCCA ÑAUPARI	Responsable:	pabel.bautista.27
Resolución:	Sin resolver	Votos:	0
Etiquetas:	Ninguno		
Estimación Restante:	0 minutes		
Tiempo Trabajado:	1 day		
Estimación original:	1 day		
Epic Link:	registrador de usuarios		
Sprint:	Tablero Sprint 1		
<p>como: usuario del sistema</p> <p>quiero: eliminar a un usuario</p> <p>para: restringir el ingreso al personal exterior.</p> <p>condición:</p> <ul style="list-style-type: none"> • en la parte izquierda de la ventana “acceso “se visualiza una tabla de usuarios. • se seleccionar una fila de usuarios y hacer clic en botón “eliminar“ 			

Nota. Fuente elaboración propia, generado con Jira Software. Generado a las Tue Apr 21 04:54:44 UTC 2020 por YONNY HUARANCCA ÑAUPARI usando JIRA 1001.0.0-SNAPSHOT#100125-sha1:a1a1592ef3306a126a3ffbb2e5ab6e40de1058d.

Tabla 16

Detalle de Historia de Usuario: HU06 - VISUALIZACIÓN DE USUARIO

[SIS-13] [HU06 - VISUALIZACIÓN DE USUARIO](#) Creada: 19/abr/20 Actualizada: 21/abr/20

Estado: Tareas por hacer
Proyecto: [SistemaRT](#)
Componentes: Ninguno
Versiones afectadas: Ninguno
Versiones corregidas: [Release 2 - SIRT lo necesario](#)

Tipo: Historia **Prioridad:** Lowest
Informador: [YONNY HUARANCCA ÑAUPARI](#) **Responsable:** [YONNY HUARANCCA ÑAUPARI](#)

Resolución: Sin resolver **Votos:** 0
Etiquetas: Ninguno
Estimación: 0 minutes
Restante:
Tiempo: 4 hours
Trabajado:
Estimación original: 4 hours
Epic Link: [registrador de usuarios](#)
Sprint: Tablero Sprint 4

como: visualización de sistema

quiero: visualizar todos los autorizados al sistema

para: ver responsabilidad

condición:

- en la parte superior de izquierdo de visualizar el usuario actual.
- a la vez una tabla de todos los usuarios registrados, con columna de nombre y apellido y DNI.

Nota. Fuente elaboración propia, generado con Jira Software. Generado a las Tue Apr 21 04:55:56 UTC 2020 por YONNY HUARANCCA ÑAUPARI usando JIRA 1001.0.0-SNAPSHOT#100125-sha1:a1a1592ef3306a126a3ffbb2e5ab6e40de1058d.

Tabla 17

Detalle de Historia de Usuario: HU07 - AGREGAR UN NUEVO

CONTRIBUYENTE URBANO

[SIS-14] [HU07 - AGREGAR NUEVO CONTRIBUYENTE URBANO](#) Creada:

19/abr/20 Actualizada: 21/abr/20

Estado: Tareas por hacer
Proyecto: [SistemaRT](#)

Componentes:	Ninguno		
Versiones afectadas:	Ninguno		
Versiones corregidas:	Release 1 - SIRT Basica		
Tipo:	Historia	Prioridad:	Medium
Informador:	YONNY HUARANCCA ÑAUPARI	Responsable:	roberth.arest.27
Resolución:	Sin resolver	Votos:	0
Etiquetas:	Ninguno		
Estimación Restante:	1 day		
Tiempo Trabajado:	3 days, 2 minutes		
Estimación original:	6 hours		
Epic Link:	contribuyente urbano		
Sprint:	Tablero Sprint 2		

como: tesorera

quiero: registrar nuevo contribuyente urbano

para: que puede pagar sus tributos

condición:

- se abre un nuevo modal
- en el modal debe existir una caja de textbox para llenar los campos como tipodoc, numdoc, nombre y apellido, celular, correo y observación.
- un botón de agregar y cancelar

Nota. Fuente elaboración propia, generado con Jira Software. Generado a las Tue Apr 21 04:57:05 UTC 2020 por YONNY HUARANCCA ÑAUPARI usando JIRA 1001.0.0-SNAPSHOT#100125-sha1:a1a1592ef3306a126a3fffb2e5ab6e40de1058d.

Tabla 18

Detalle de Historia de Usuario: HU08 - BUSCADOR DE UN CONTRIBUYENTE URBANO

[SIS-15] HU08 - BUSCADOR DE CONTRIBUYENTE URBANO Creada:			
19/abr/20 Actualizada: 21/abr/20			
Estado:	Tareas por hacer		
Proyecto:	SistemaRT		
Componentes:	Ninguno		
Versiones afectadas:	Ninguno		
Versiones corregidas:	Release 1 - SIRT Basica		
Tipo:	Historia	Prioridad:	Medium
Informador:	YONNY HUARANCCA ÑAUPARI	Responsable:	roberth.arest.27

Resolución:	Sin resolver	Votos:	0
Etiquetas:	Ninguno		
Estimación Restante:	1 day		
Tiempo Trabajado:	2 days		
Estimación original:	6 hours		
Epic Link:	contribuyente urbano		
Sprint:	Tablero Sprint 2		

como: tesorera

quiero: encontrar rapido a un contribuyente

para: abrir un nuevo proceso de pago

condición:

- en la parte superior se encuentra dos cajas de textbox para ingresar nombre o número de doc. y así de puede buscar.
- en la parte medio debe visualizase un tabla del contribuyente.

Nota. Fuente elaboración propia, generado con Jira Software. Generado a las Tue Apr 21 04:58:37 UTC 2020 por YONNY HUARANCCA ÑAUPARI usando JIRA 1001.0.0-SNAPSHOT#100125-sha1:a1a1592ef3306a126a3ffbb2e5ab6e40de1058d.

Tabla 19

Detalle de Historia de Usuario: HU09 - EDITAR/ELIMINAR CONTRIBUYENTE

URBANO

[SIS-16] [HU09 - EDITAR/ELIMINAR CONTRIBUYENTE URBANO](#) Creada:

19/abr/20 Actualizada: 21/abr/20

Estado:	Tareas por hacer		
Proyecto:	SistemaRT		
Componentes:	Ninguno		
Versiones afectadas:	Ninguno		
Versiones corregidas:	Release 1 - SIRT Basica		
Tipo:	Historia	Prioridad:	Highest
Informador:	YONNY HUARANCCA ÑAUPARI	Responsable:	roberth.aresta.27

Resolución:	Sin resolver	Votos:	0
Etiquetas:	Ninguno		
Estimación Restante:	1 day		
Tiempo Trabajado:	2 days		
Estimación original:	6 hours		
Epic Link:	contribuyente urbano		
Sprint:	Tablero Sprint 2		

como: tesorera

quiero: editar o eliminar contribuyente

para: facilitar el orden

condición:

- en la tabla se selecciona una de las filas de los contribuyentes para luego abrir el modal y modificar algún campo.
- al seleccionar de la fila de la tabla y hace clic en el botón eliminar para borrar el contribuyente urbano.

Nota. Fuente elaboración propia, generado con Jira Software. Generado a las Tue Apr 21 04:59:46 UTC 2020 por YONNY HUARANCCA ÑAUPARI usando JIRA 1001.0.0-SNAPSHOT#100125-sha1:a1a1592ef3306a126a3ffbb2e5ab6e40de1058d.

Tabla 20

Detalle de Historia de Usuario: HU10 - ABRIR UN CONTRIBUYENTE URBANO

[SIS-17] [HU10 - ABRIR UN CONTRIBUYENTE URBANO](#) Creada:

19/abr/20 Actualizada: 21/abr/20

Estado:	Tareas por hacer		
Proyecto:	SistemaRT		
Componentes:	Ninguno		
Versiones afectadas:	Ninguno		
Versiones corregidas:	Release 1 - SIRT Basica		
Tipo:	Historia	Prioridad:	Highest
Informador:	YONNY HUARANCCA ÑAUPARI	Responsable:	roberth.aresto.27
Resolución:	Sin resolver	Votos:	0
Etiquetas:	Ninguno		
Estimación Restante:	3 hours, 59 minutes		
Tiempo Trabajado:	1 minute		
Estimación original:	4 hours		
Epic Link:	contribuyente urbano		
Sprint:	Tablero Sprint 2		

como: tesorera

quiero: visualizar el predio y la lista de propiedad

para: crear propiedad y generar lista de pago

condición:

- debe haber un botón llamado “abrir “al momento de seleccionar un contribuyente.
- se visualizará un modal llamado “abrir persona urbana “

Nota. Fuente elaboración propia, generado con Jira Software. Generado a las Tue Apr 21 05:01:06 UTC 2020 por YONNY HUARANCCA ÑAUPARI usando JIRA 1001.0.0-SNAPSHOT#100125-sha1:a1a1592ef3306a126a3ffbb2e5ab6e40de1058d.

Tabla 21

Detalle de Historia de Usuario: HU11 - VISUALIZAR ABRIR PERSONA

URBANA

[SIS-18] [HU11 - VISUALIZAR ABRIR PERSONA URBANA](#) Creada:

19/abr/20 Actualizada: 21/abr/20

Estado:	Tareas por hacer		
Proyecto:	SistemaRT		
Componentes:	Ninguno		
Versiones afectadas:	Ninguno		
Versiones corregidas:	Release 2 - SIRT lo necesario		
Tipo:	Historia	Prioridad:	Highest
Informador:	YONNY HUARANCCA ÑAUPARI	Responsable:	roberth.aresta.27
Resolución:	Sin resolver	Votos:	0
Etiquetas:	Ninguno		
Estimación Restante:	1 day		
Tiempo Trabajado:	2 days		
Estimación original:	3 hours		
Epic Link:	contribuyente urbano		
Sprint:	Tablero Sprint 2		

como: tesorera

quiero: visualizar el modal

para: realizar los procesos de generar lista de pagos.

condición:

- debe existir el proceso de crear nueva propiedad urbano.
- debe existir para generar nuevo año.
- de haber un proceso para agregar predio urbano.
- tres tablas como propiedades, predios y lista de pagos.

Nota. Fuente elaboración propia, generado con Jira Software. Generado a las Tue Apr 21 05:02:54 UTC 2020 por YONNY HUARANCCA ÑAUPARI usando JIRA 1001.0.0-SNAPSHOT#100125-sha1:a1a1592ef3306a126a3ffbb2e5ab6e40de1058d.

Tabla 22

Detalle de Historia de Usuario: HUI2 - CREAR PROPIEDAD DE ZONA URBANA

[SIS-19] HUI2 - CREAR PROPIEDAD DE ZONA URBANA Creada:			
19/abr/20 Actualizada: 21/abr/20			
Estado:	Tareas por hacer		
Proyecto:	SistemaRT		
Componentes:	Ninguno		
Versiones afectadas:	Ninguno		
Versiones corregidas:	Release 2 - SIRT lo necesario		
Tipo:	Historia	Prioridad:	Highest
Informador:	YONNY HUARANCCA ÑAUPARI	Responsable:	manrique.meneses.27
Resolución:	Sin resolver	Votos:	0
Etiquetas:	Ninguno		
Estimación Restante:	4 hours		
Tiempo Trabajado:	2 days		
Estimación original:	1 day		
Epic Link:	contribuyente urbano		
Sprint:	Tablero Sprint 2		
como: tesorera			
quiero: crear nueva propiedad urbano.			
para: agregar un predio.			
condición:			
<ul style="list-style-type: none">• al hacer clic en el botón crear propiedad, se debe abrir un modal, con los campos de textbox de llenado como dirección, departamento, provincia y distrito.• debe existir botón de guardar y cancelar.			

Nota. Fuente elaboración propia, generado con Jira Software. Generado a las Tue Apr 21 05:03:58 UTC 2020 por YONNY HUARANCCA ÑAUPARI usando JIRA 1001.0.0-SNAPSHOT#100125-sha1:a1a1592ef3306a126a3fffb2e5ab6e40de1058d.

Tabla 23

Detalle de Historia de Usuario: HUI3 - CREAR AÑO PARA PREDIO URBANO

[SIS-20] HUI3 - CREAR AÑO PARA PREDIO URBANO Creada: 19/abr/20 Actualizada:	
21/abr/20	
Estado:	Tareas por hacer
Proyecto:	SistemaRT

Componentes:	Ninguno		
Versiones afectadas:	Ninguno		
Versiones corregidas:	Release 2 - SIRT lo necesario		
Tipo:	Historia	Prioridad:	Highest
Informador:	YONNY HUARANCCA ÑAUPARI	Responsable:	manrique.meneses.27
Resolución:	Sin resolver	Votos:	0
Etiquetas:	Ninguno		
Estimación Restante:	4 hours		
Tiempo Trabajado:	1 day		
Estimación original:	1 day		
Epic Link:	contribuyente urbano		
Sprint:	Tablero Sprint 2		

como: tesorera.

quiero: crear un nuevo año

quiero: crear un nuevo predio.

condición:

- debe existir cajas de textbox para llenado de año, arancel y área total.
- debe tener un botón de crear año.

Nota. Fuente elaboración propia, generado con Jira Software. Generado a las Tue Apr 21 05:05:28 UTC 2020 por YONNY HUARANCCA ÑAUPARI usando JIRA 1001.0.0-SNAPSHOT#100125-sha1:a1a1592ef3306a126a3ffbb2e5ab6e40de1058d.

Tabla 24

Detalle de Historia de Usuario: HU14 - CREAR NUEVO PREDIO URBANO

[SIS-21] HU14 - CREAR NUEVO PREDIO URBANO Creada: 19/abr/20 Actualizada: 21/abr/20			
Estado:	Tareas por hacer		
Proyecto:	SistemaRT		
Componentes:	Ninguno		
Versiones afectadas:	Ninguno		
Versiones corregidas:	Release 2 - SIRT lo necesario		
Tipo:	Historia	Prioridad:	Highest
Informador:	YONNY HUARANCCA ÑAUPARI	Responsable:	manrique.meneses.27
Resolución:	Sin resolver	Votos:	0
Etiquetas:	Ninguno		
Estimación Restante:	1 hour		
Tiempo Trabajado:	1 day		

Estimación original: 1 day
 Epic Link: [contribuyente urbano](#)
 Sprint: Tablero Sprint 2

como: tesorera

quiero: crear un nuevo predio

para: medir predio urbano

condición:

- al hacer clic en el botón crear predio se debe abrir modal.
- debe tener seleccionador de clasificación de material, estado y año.
- debe tener cajas de textbox, como bloque, piso, año y antigüedad.
- seleccionadores de los valores de área de techado como: muro, techo, piso, puerta, revestimiento, baño y instalación eléctrica.
- y una caja de textbox de respuesta del predio y un botón de guardar y cancelar.

Nota. Fuente elaboración propia, generado con Jira Software. Generado a las Tue Apr 21 05:06:28 UTC 2020 por YONNY HUARANCCA ÑAUPARI usando JIRA 1001.0.0-SNAPSHOT#100125-sha1:a1a1592ef3306a126a3ffbb2e5ab6e40de1058d.

Tabla 25

Detalle de Historia de Usuario: HU15 - GENERAR LISTA DE PAGO URBANO

[SIS-22] HU15 - GENERAR LISTA DE PAGO URBANO Creada: 19/abr/20 Actualizada: 21/abr/20			
Estado:	Tareas por hacer		
Proyecto:	SistemaRT		
Componentes:	Ninguno		
Versiones afectadas:	Ninguno		
Versiones corregidas:	Release 2 - SIRT lo necesario		
Tipo:	Historia	Prioridad:	Highest
Informador:	YONNY HUARANCCA ÑAUPARI	Responsable:	manrique.menes.es.27
Resolución:	Sin resolver	Votos:	0
Etiquetas:	Ninguno		
Estimación Restante:	4 hours		
Tiempo Trabajado:	1 day		
Estimación original:	4 hours		
Epic Link:	contribuyente urbano		
Sprint:	Tablero Sprint 2		
como: tesorera			

quiero: generar una lista de pagos para el año actual.

para: saber los que cumplan con el pago de concepto de pago al impuesto predial.

condición:

- en la parte inferior izquierda debe tener un ingreso de una caja de textbox año y uit.
- botón de generar y una tabla de visualización de lista de pagos.

Nota. Fuente elaboración propia, generado con Jira Software. Generado a las Tue Apr 21 05:08:17 UTC 2020 por YONNY HUARANCCA ÑAUPARI usando JIRA 1001.0.0-SNAPSHOT#100125-sha1:a1a1592ef3306a126a3ffbb2e5ab6e40de1058d.

Tabla 26

Detalle de Historia de Usuario: HU16 - CERRAR/MINIMIZAR LA VISTA DE ABRIR PERSONA URBANA

[SIS-23] [HU16 - CERRAR/MINIMIZAR LA VISTA ABRIR PERSONA](#)

[URBANA](#) Creada: 19/abr/20 Actualizada: 21/abr/20

Estado:	Tareas por hacer		
Proyecto:	SistemaRT		
Componentes:	Ninguno		
Versiones afectadas:	Ninguno		
Versiones corregidas:	Release 3 - SIRT menos básico		
Tipo:	Historia	Prioridad:	Low
Informador:	YONNY HUARANCCA ÑAUPARI	Responsable:	manrique.menes.es.27
Resolución:	Sin resolver	Votos:	0
Etiquetas:	Ninguno		
Estimación Restante:	0 minutes		
Tiempo Trabajado:	1 day		
Estimación original:	1 hour		
Epic Link:	contribuyente urbano		
Sprint:	Tablero Sprint 2		

como: tesorera.

quiero: cerrar o minimizar el modal

para: facilidad de trabajo.

condición:

- debe tener un campo amigable de un botón cerrar en la parte inferior derecha.

- tanto cerrar y minimizar en la parte superior derecha debe existir.

Nota. Fuente elaboración propia, generado con Jira Software. Generado a las Tue Apr 21 05:09:41 UTC 2020 por YONNY HUARANCCA ÑAUPARI usando JIRA 1001.0.0-SNAPSHOT#100125-sha1:a1a1592ef3306a126a3fffb2e5ab6e40de1058d.

Tabla 27

Detalle de Historia de Usuario: HUI7 - AGREGAR UN NUEVO

CONTRIBUYENTE RURAL O RUSTICO

[SIS-24] HUI7 - AGREGAR UN NUEVO CONTRIBUYENTE RURAL O

RUSTICO Creada: 19/abr/20 Actualizada: 21/abr/20

Estado: Tareas por hacer

Proyecto: [SistemaRT](#)

Componentes: Ninguno

Versiones: Ninguno

afectadas:

Versiones [Release 1 - SIRT Basica](#)

corregidas:

Tipo: Historia **Prioridad:** Highest

Informador: [YONNY HUARANCCA ÑAUPARI](#) **Responsable:** [YONNY HUARANCCA ÑAUPARI](#)

Resolución: Sin resolver **Votos:** 0

Etiquetas: Ninguno

Estimación: 3 hours

Restante:

Tiempo: 2 days

Trabajado:

Estimación: 2 hours

original:

Epic Link: [contribuyente rural o rustico](#)

Sprint: Tablero Sprint 3

como: tesorera

quiero: agregar un nuevo contribuyente de la zona rural.

para: que puede pagar sus tributos.

condición:

- se abre un nuevo modal al hacer clic en botón agregar.
- en el modal del nuevo contribuyente existe las cajas de textbox de tipo doc., num. doc, celular correo y observación.
- un botón para agregar y cancelar.

Nota. Fuente elaboración propia, generado con Jira Software. Generado a las Tue Apr 21 05:14:22 UTC 2020 por YONNY HUARANCCA ÑAUPARI usando JIRA 1001.0.0-SNAPSHOT#100125-sha1:a1a1592ef3306a126a3fffb2e5ab6e40de1058d.

Tabla 28

Detalle de Historia de Usuario: HU18 - BUSCADOR DE UN CONTRIBUYENTE RURAL O RUSTICO

[SIS-25] HU18 - BUSCADOR DE CONTRIBUYENTE RURAL O RUSTICO			
Creada: 19/abr/20 Actualizada: 21/abr/20			
Estado:	Tareas por hacer		
Proyecto:	SistemaRT		
Componentes:	Ninguno		
Versiones afectadas:	Ninguno		
Versiones corregidas:	Release 1 - SIRT Basica		
Tipo:	Historia	Prioridad:	Medium
Informador:	YONNY HUARANCCA ÑAUPARI	Responsable:	YONNY HUARANCCA ÑAUPARI
Resolución:	Sin resolver	Votos:	0
Etiquetas:	Ninguno		
Estimación	0 minutes		
Restante:			
Tiempo	1 day		
Trabajado:			
Estimación	1 hour		
original:			
Epic Link:	contribuyente rural o rustico		
Sprint:	Tablero Sprint 3		

como: tesorera

quiero: encontrar rápido a un contribuyente de la zona rural.

para: agilizar los siguientes procesos.

condición:

- debe existir dos cajas de textbox para llenar nombre o DNI y así se busque rápido
- un botón de buscar

Nota. Fuente elaboración propia, generado con Jira Software. Generado a las Tue Apr 21 05:15:25 UTC 2020 por YONNY HUARANCCA ÑAUPARI usando JIRA 1001.0.0-SNAPSHOT#100125-sha1:a1a1592ef3306a126a3fffb2e5ab6e40de1058d.

Tabla 29

Detalle de Historia de Usuario: HUI9 - EDITAR/ELIMINAR CONTRIBUYENTE RURAL O RUSTICO

[SIS-26] [HUI9 - EDITAR/ELIMINAR CONTRIBUYENTE DE ZONA RURAL](#)

[O RUSTICO](#) Creada: 19/abr/20 Actualizada: 21/abr/20

Estado: Tareas por hacer

Proyecto: [SistemaRT](#)

Componentes: Ninguno

Versiones: Ninguno

afectadas:

Versiones [Release 1 - SIRT Basica](#)

corregidas:

Tipo: Historia **Prioridad:** Highest

Informador: [YONNY](#) **Responsable:** [YONNY HUARANCCA ÑAUPARI](#)

[HUARANCCA](#)

[ÑAUPARI](#)

Resolución: Sin resolver **Votos:** 0

Etiquetas: Ninguno

Estimación: 1 day

Restante:

Tiempo: 2 days

Trabajado:

Estimación: 4 hours

original:

Epic Link: [contribuyente rural o rustico](#)

Sprint: Tablero Sprint 3

como: tesorera

quiero: editar o eliminar a un contribuyente de zona rural.

para: tener más orden en el registro.

condición:

- en la parte media del panel existe una tabla contenido los registros de todo contribuyente, se selecciona para editar o eliminar al contribuyente de zona rural o rustica.

Nota. Fuente elaboración propia, generado con Jira Software. Generado a las Tue Apr 21 05:16:31 UTC 2020 por YONNY HUARANCCA ÑAUPARI usando JIRA 1001.0.0-SNAPSHOT#100125-sha1:a1a1592ef3306a126a3fffb2e5ab6e40de1058d.

Tabla 30

Detalle de Historia de Usuario: HU20 - ABRIR UN CONTRIBUYENTE RURAL O RUSTICO

[SIS-27] [HU20 - ABRIR UN CONTRIBUYENTE DE DE LA ZONA RURAL O RUSTICO](#) Creada: 19/abr/20 Actualizada: 21/abr/20

Estado:	Tareas por hacer		
Proyecto:	SistemaRT		
Componentes:	Ninguno		
Versiones afectadas:	Ninguno		
Versiones corregidas:	Release 1 - SIRT Basica		
Tipo:	Historia	Prioridad:	High
Informador:	YONNY HUARANCCA ÑAUPARI	Responsable:	elvis.tineo.27
Resolución:	Sin resolver	Votos:	0
Etiquetas:	Ninguno		
Estimación Restante:	6 hours		
Tiempo Trabajado:	1 day		
Estimación original:	1 day		
Epic Link:	contribuyente rural o rustico		
Sprint:	Tablero Sprint 3		

como: tesorera

quiero: visualizar el predio y lista de propiedades.

para: crear propiedad, agregar predio y generar lista de pago.

condición:

- existe un botón abrir, primero selecciona un contribuyente en la lista de la tabla de contenido.
- al hace clic en el botón abrir se abre modal.

Nota. Fuente elaboración propia, generado con Jira Software. Generado a las Tue Apr 21 05:17:45 UTC 2020 por YONNY HUARANCCA ÑAUPARI usando JIRA 1001.0.0-SNAPSHOT#100125-sha1:a1a1592ef3306a126a3ffbb2e5ab6e40de1058d.

Tabla 31

Detalle de Historia de Usuario: HU21 - VISUALIZAR ABRIR PERSONA RURAL O RUSTICO

[SIS-28] [HU21 - VISUALIZAR LA VENTANA DE ABRIR PERSONA RURAL O RUSTICO](#) Creada: 19/abr/20 Actualizada: 21/abr/20

Estado:	Tareas por hacer		
Proyecto:	SistemaRT		
Componentes:	Ninguno		
Versiones afectadas:	Ninguno		
Versiones corregidas:	Release 2 - SIRT lo necesario		
Tipo:	Historia	Prioridad:	Low
Informador:	YONNY HUARANCCA ÑAUPARI	Responsable:	elvis.tineo.27
Resolución:	Sin resolver	Votos:	0
Etiquetas:	Ninguno		
Estimación Restante:	0 minutes		
Tiempo Trabajado:	1 day		
Estimación original:	2 hours		
Epic Link:	contribuyente rural o rustico		
Sprint:	Tablero Sprint 3		

como: tesorera

quiero: visualizar el modal.

para: generar lista de pago.

condición:

- debe existir el proceso de crear un nueva propiedad urbano.
- debe existir para generar un nuevo año.
- un proceso para agregar predio rural.
- tres tablas como propiedad, predio y lista de pago.

Nota. Fuente elaboración propia, generado con Jira Software. Generado a las Tue Apr 21 05:21:12 UTC 2020 por YONNY HUARANCCA ÑAUPARI usando JIRA 1001.0.0-SNAPSHOT#100125-sha1:a1a1592ef3306a126a3fffb2e5ab6e40de1058d.

Tabla 32

Detalle de Historia de Usuario: HU22 - CREAR PROPIEDAD DE ZONA RURAL O RUSTICO

[SIS-29] [HU22 - CREAR PROPIEDAD DE ZONA RURAL O RUSTICO](#)

Creada: 19/abr/20 Actualizada: 21/abr/20

Estado:	Tareas por hacer		
Proyecto:	SistemaRT		
Componentes:	Ninguno		
Versiones afectadas:	Ninguno		
Versiones corregidas:	Release 2 - SIRT lo necesario		
Tipo:	Historia	Prioridad:	Highest

Informador:	YONNY HUARANCCA ÑAUPARI	Responsable:	manrique.meneses.27
Resolución:	Sin resolver	Votos:	0
Etiquetas:	Ninguno		
Estimación Restante:	1 day		
Tiempo Trabajado:	4 days		
Estimación original:	1 day		
Epic Link:	contribuyente rural o rustico		
Sprint:	Tablero Sprint 3		

como: tesorera

quiero: crear propiedad

para: agregar una nueva propiedad

condición:

- en la vista del modal, debe existir cajas de textbox de dirección, departamento, provincia, distrito, colindante sur, norte, oeste, oeste y referencia.
- dos botones de guardar y cancelar

Nota. Fuente elaboración propia, generado con Jira Software. Generado a las Tue Apr 21 05:22:19 UTC 2020 por YONNY HUARANCCA ÑAUPARI usando JIRA 1001.0.0-SNAPSHOT#100125-sha1:a1a1592ef3306a126a3fffb2e5ab6e40de1058d.

Tabla 33

Detalle de Historia de Usuario: HU23 - CREAR AÑO PARA PREDIO RURAL O RUSTICO

[SIS-30] [HU23 - CREAR AÑO PARA PREDIO RURAL O RUSTICO](#)

Creada: 19/abr/20 Actualizada: 21/abr/20

Estado:	Tareas por hacer		
Proyecto:	SistemaRT		
Componentes:	Ninguno		
Versiones afectadas:	Ninguno		
Versiones corregidas:	Release 2 - SIRT lo necesario		
Tipo:	Historia	Prioridad:	Highest
Informador:	YONNY HUARANCCA ÑAUPARI	Responsable:	manrique.meneses.27
Resolución:	Sin resolver	Votos:	0
Etiquetas:	Ninguno		
Estimación Restante:	1 day		
Tiempo Trabajado:	4 days		

Estimación original: 1 day
 Epic Link: [contribuyente rural o rustico](#)
 Sprint: Tablero Sprint 3

como: tesorera

quiero: crear año actual

para: generar nuevo predio.

condición:

- debe de existir cajas de textbox para llenado de año, arancel y área total.
- debe tener un botón de crear año

Nota. Fuente elaboración propia, generado con Jira Software. Generado a las Tue Apr 21 05:24:05 UTC 2020 por YONNY HUARANCCA ÑAUPARI usando JIRA 1001.0.0-SNAPSHOT#100125-sha1:a1a1592ef3306a126a3ffbb2e5ab6e40de1058d.

Tabla 34

Detalle de Historia de Usuario: HU24 - CREAR NUEVO PREDIO RURAL O RUSTICO

[SIS-31] HU24 - CREAR NUEVO PREDIO RURAL O RUSTICO Creada:			
19/abr/20 Actualizada: 21/abr/20			
Estado:	Tareas por hacer		
Proyecto:	SistemaRT		
Componentes:	Ninguno		
Versiones afectadas:	Ninguno		
Versiones corregidas:	Release 2 - SIRT lo necesario		
Tipo:	Historia	Prioridad:	Highest
Informador:	YONNY HUARANCCA ÑAUPARI	Responsable:	manrique.meneses.27
Resolución:	Sin resolver	Votos:	0
Etiquetas:	Ninguno		
Estimación Restante:	2 hours		
Tiempo Trabajado:	1 day		
Estimación original:	4 hours		
Epic Link:	contribuyente rural o rustico		
Sprint:	Tablero Sprint 3		

como: tesorera

quiero: crear un nuevo predio rustico

para: medir predio rustico

condición:

- al hacer clic en botón “crear predio” se abre un modal.
- debe existir un botón de selección como el clasificador de terreno de cultivo, ganadería o otras tierras.
- textbox de arrea terreno
- botón de calcular, guardar y cancelar

Nota. Fuente elaboración propia, generado con Jira Software. Generado a las Tue Apr 21 05:25:09 UTC 2020 por YONNY HUARANCCA ÑAUPARI usando JIRA 1001.0.0-SNAPSHOT#100125-sha1:a1a1592ef3306a126a3ffbb2e5ab6e40de1058d.

Tabla 35

Detalle de Historia de Usuario: HU25 - GENERAR LISTA DE PAGO RURAL O RUSTICO

[SIS-32] [HU25 - LISTA DE PAGO DE ZONA RURAL O RUSTICO](#) Creada:

19/abr/20 Actualizada: 21/abr/20

Estado:	Tareas por hacer		
Proyecto:	SistemaRT		
Componentes:	Ninguno		
Versiones afectadas:	Ninguno		
Versiones corregidas:	Release 2 - SIRT lo necesario		
Tipo:	Historia	Prioridad:	Medium
Informador:	YONNY HUARANCCA ÑAUPARI	Responsable:	manrique.meneses.27
Resolución:	Sin resolver	Votos:	0
Etiquetas:	Ninguno		
Estimación Restante:	0 minutes		
Tiempo Trabajado:	2 days		
Estimación original:	4 hours		
Epic Link:	contribuyente rural o rustico		
Sprint:	Tablero Sprint 3		

como: tesorera

quiero: generar una lista de pago para el año actual.

para: saber los que cumplen con el pago de concepto al pago del impuesto predial.

condición:

- en la parte inferior izquierda debe tener un ingreso de una caja de textbox de año y UIT.
- botón de generar y una tabla de visualización de lista de pago.

Nota. Fuente elaboración propia, generado con Jira Software. Generado a las Tue Apr 21 05:26:26 UTC 2020 por YONNY HUARANCCA ÑAUPARI usando JIRA 1001.0.0-SNAPSHOT#100125-sha1:a1a1592ef3306a126a3ffbb2e5ab6e40de1058d.

Tabla 36

Detalle de Historia de Usuario: HU26 - CERRAR/MINIMIZAR LA VISTA DE ABRIR PERSONA RURAL O RUSTICO

[SIS-33] HU26 - CERRAR/MINIMIZAR LA VISTA ABRIR PERSONA RURAL O RUSTICO Creada: 19/abr/20 Actualizada: 21/abr/20

Estado:	Tareas por hacer		
Proyecto:	SistemaRT		
Componentes:	Ninguno		
Versiones afectadas:	Ninguno		
Versiones corregidas:	Release 3 - SIRT menos básico		
Tipo:	Historia	Prioridad:	Low
Informador:	YONNY HUARANCCA ÑAUPARI	Responsable:	manrique.meneses.27
Resolución:	Sin resolver	Votos:	0
Etiquetas:	Ninguno		
Estimación Restante:	0 minutes		
Tiempo Trabajado:	1 day		
Estimación original:	4 hours		
Epic Link:	contribuyente rural o rustico		
Sprint:	Tablero Sprint 3		

como: tesorera

como: cerrar o minimizar el modal

para: facilidad de trabajo

condición:

- debe tener un campo amigable de un botón cerrar en la parte inferior derecha.
- tanto cerrar y minimizar en la parte superior derecha.

Nota. Fuente elaboración propia, generado con Jira Software. Generado a las Tue Apr 21 05:27:38 UTC 2020 por YONNY HUARANCCA ÑAUPARI usando JIRA 1001.0.0-SNAPSHOT#100125-sha1:a1a1592ef3306a126a3ffbb2e5ab6e40de1058d.

Tabla 37

Detalle de Historia de Usuario: HU27 - BUSCAR LISTA DE CONTRIBUYENTE

[SIS-34] [HU27 - BUSCAR LISTA DE CONTRIBUYENTE](#) Creada:
19/abr/20 Actualizada: 21/abr/20

Estado:	Tareas por hacer		
Proyecto:	SistemaRT		
Componentes:	Ninguno		
Versiones afectadas:	Ninguno		
Versiones corregidas:	Release 1 - SIRT Basica		
Tipo:	Historia	Prioridad:	High
Informador:	YONNY HUARANCCA ÑAUPARI	Responsable:	pabel.bautista.27
Resolución:	Sin resolver	Votos:	0
Etiquetas:	Ninguno		
Estimación Restante:	1 hour		
Tiempo Trabajado:	1 day		
Estimación original:	4 hours		
Epic Link:	caja		
Sprint:	Tablero Sprint 4		

como: tesorera

quiero: buscar un contribuyente

para: agilizar el proceso.

condición:

- debe existir una caja de textbox con nombre documento para encontrar rápido al contribuyente y botón buscar.
- se genera en una tabla con los parámetros documento y nombre en la parte inferior en lado izquierdo.

Nota. Fuente elaboración propia, generado con Jira Software. Generado a las Tue Apr 21 05:29:00 UTC 2020 por YONNY HUARANCCA ÑAUPARI usando JIRA 1001.0.0-SNAPSHOT#100125-sha1:a1a1592ef3306a126a3ffbb2e5ab6e40de1058d.

Tabla 38

Detalle de Historia de Usuario: HU28 - VISUALIZACIÓN DEL LISTADO DE PROPIEDAD

[SIS-35] [HU28 - VISUALIZACIÓN DEL LISTADO DE PROPIEDAD](#)
Creada: 19/abr/20 Actualizada: 21/abr/20

Estado:	Tareas por hacer
Proyecto:	SistemaRT

Componentes:	Ninguno		
Versiones afectadas:	Ninguno		
Versiones corregidas:	Release 1 - SIRT Basica		
Tipo:	Historia	Prioridad:	Medium
Informador:	YONNY HUARANCCA ÑAUPARI	Responsable:	pabel.bautista.27
Resolución:	Sin resolver	Votos:	0
Etiquetas:	Ninguno		
Estimación Restante:	0 minutes		
Tiempo Trabajado:	2 hours		
Estimación original:	2 minutes		
Epic Link:	caja		
Sprint:	Tablero Sprint 4		

como: tesorera

quiero: visualizar el listado de propiedades

para: generar pago

condición:

- al hacer clic en contribuyente se muestra en una tabla las propiedades del dueño, con los parámetros dirección y distrito.

Nota. Fuente elaboración propia, generado con Jira Software. Generado a las Tue Apr 21 05:30:03 UTC 2020 por YONNY HUARANCCA ÑAUPARI usando JIRA 1001.0.0-SNAPSHOT#100125-sha1:a1a1592ef3306a126a3ffbb2e5ab6e40de1058d.

Tabla 39

Detalle de Historia de Usuario: HU29 - GENERAR PAGO DEL IMPUESTO PREDIAL

[SIS-36] HU29 - GENERAR PAGO DEL IMPUESTO PREDIAL Creada:			
19/abr/20 Actualizada: 21/abr/20			
Estado:	Tareas por hacer		
Proyecto:	SistemaRT		
Componentes:	Ninguno		
Versiones afectadas:	Ninguno		
Versiones corregidas:	Release 1 - SIRT Basica		
Tipo:	Historia	Prioridad:	Highest
Informador:	YONNY HUARANCCA ÑAUPARI	Responsable:	pabel.bautista.27
Resolución:	Sin resolver	Votos:	0
Etiquetas:	Ninguno		

Estimación Restante: 0 minutos
 Tiempo Trabajado: 4 days
 Estimación original: 1 day
 Epic Link: [caja](#)
 Sprint: Tablero Sprint 4

como: tesorería

quiero: generar pago

para: emitir boleta de impuesto predial y declaración jurado de autovaluo.

condición:

- se debe seleccionar una propiedad para visualizar se está cancelado, clic en el botón pagar y se genera para imprimir dos pdfs del documento de boleta y declaración jurada de autovaluo.

Nota. Fuente elaboración propia, generado con Jira Software. Generado a las Tue Apr 21 05:31:03 UTC 2020 por YONNY HUARANCCA ÑAUPARI usando JIRA 1001.0.0-SNAPSHOT#100125-sha1:a1a1592ef3306a126a3ffbb2e5ab6e40de1058d.

Tabla 40

Detalle de Historia de Usuario: HU30 - VISUALIZACIÓN DE REPORTE

URBANO O RURAL

[SIS-37] HU30 - VISUALIZACIÓN DE REPORTES URBANO Y RURAL

Creada: 19/abr/20 Actualizada: 21/abr/20

Estado:	Tareas por hacer		
Proyecto:	SistemaRT		
Componentes:	Ninguno		
Versiones afectadas:	Ninguno		
Versiones corregidas:	Release 1 - SIRT Basica		
Tipo:	Historia	Prioridad:	Highest
Informador:	YONNY HUARANCCA ÑAUPARI	Responsable:	roberth.arest.27
Resolución:	Sin resolver	Votos:	0
Etiquetas:	Ninguno		
Estimación Restante:	1 day		
Tiempo Trabajado:	1 week		
Estimación original:	1 day		
Epic Link:	reportes		
Sprint:	Tablero Sprint 4		

como: tesorera y usuario de sistema.

quiero: visualizar reporte semanal, mensual y anual.

para: ver el ingreso de capita a la municipalidad y para su respectivo análisis.

condición:

- debe existir un seleccionador de tiempo como semanal, mensual y anual, se debe mostrar en una tabla el ingreso cápita a la municipalidad como la cantidad de personas que pago o cantidad de personas que faltan pagar su tributo.
- existe un botón para imprimir el reporte.

Nota. Fuente elaboración propia, generado con Jira Software. Generado a las Tue Apr 21 05:32:03 UTC 2020 por YONNY HUARANCCA ÑAUPARI usando JIRA 1001.0.0-SNAPSHOT#100125-sha1:a1a1592ef3306a126a3ffbb2e5ab6e40de1058d.

Tabla 41

Detalle de Historia de Usuario: HU31 - VISUALIZACIÓN DE DOCUMENTO

[SIS-38] HU31 - VISUALIZACIÓN DE DOCUMENTOS Creada:

19/abr/20 Actualizada: 21/abr/20

Estado:	Tareas por hacer		
Proyecto:	SistemaRT		
Componentes:	Ninguno		
Versiones afectadas:	Ninguno		
Versiones corregidas:	Release 1 - SIRT Basica		
Tipo:	Historia	Prioridad:	High
Informador:	YONNY HUARANCCA ÑAUPARI	Responsable:	elvis.tineo.27
Resolución:	Sin resolver	Votos:	0
Etiquetas:	Ninguno		
Estimación Restante:	1 day		
Tiempo Trabajado:	3 days		
Estimación original:	1 day		
Epic Link:	normas legales		
Sprint:	Tablero Sprint 4		

como: tesorera

quiero: visualizar los documentos

para: informarse y saber normas legales.

condición:

- se muestra un conjunto de documentos legales relacionado a la construcción del sistema.

- visualización de los documentos en formato jpg .

Nota. Fuente elaboración propia, generado con Jira Software. Generado a las Tue Apr 21 05:33:02 UTC 2020 por YONNY HUARANCCA ÑAUPARI usando JIRA 1001.0.0-SNAPSHOT#100125-sha1:a1a1592ef3306a126a3fffb2e5ab6e40de1058d.

Tabla 42

Detalle de Historia de Usuario: HU32 - VISUALIZACIÓN DE AYUDA

[SIS-39] HU32 - VISUALIZACIÓN DE AYUDA Creada: 19/abr/20 Actualizada: 21/abr/20			
Estado:	Tareas por hacer		
Proyecto:	SistemaRT		
Componentes:	Ninguno		
Versiones afectadas:	Ninguno		
Versiones corregidas:	Release 1 - SIRT Basica		
Tipo:	Historia	Prioridad:	High
Informador:	YONNY HUARANCCA ÑAUPARI	Responsable:	elvis.tineo.27
Resolución:	Sin resolver	Votos:	0
Etiquetas:	Ninguno		
Estimación Restante:	2 hours		
Tiempo Trabajado:	3 days		
Estimación original:	2 hours		
Epic Link:	normas legales		
Sprint:	Tablero Sprint 4		

como: tesorera y usuario del sistema

quiero: visualizar ayuda el buen uso del sistema

para: el buen uso del sistema

condición:

- en la parte de medio se visualiza un tutorial de vídeo para aprendizaje y el buen uso del sistema

Nota. Fuente elaboración propia, generado con Jira Software.

7.2.2. REUNIONES DE PLANIFICACIÓN DE SPRINT

Teniendo en cuenta el compromiso del equipo donde todos los miembros tienen una misma misión, por lo que tienen que reunirse para planificar el trabajo que se va realizar durante el sprint, con la participación de Producto Owner, Scrum Master y el equipo de trabajo.

Al cabo de cada reunión de planificación se hace un tablero de sprint, que cada uno de ellos tiene un conjunto de incidencias o tareas. Cada sprint tiene una fecha de inicio y una fecha de entrega, donde acabado un sprint inmediatamente se comienza el siguiente sprint en el tablero, así ocasionando una iteración de tareas.

A. Primera reunión de planificación (SPRINT 1)

Fecha: 19/08/2019

Hora: 9:30am -11:30am

Lugar: empresa Ennova m&f s.a.c.

Próxima reunión: 02/09/2019

- 9:30am -10:30am. Se reúnen tanto el representante de la municipalidad (cliente) y los involucrados en el desarrollo de la implementación del Sistema de Información de Recaudación Tributaria de la Municipalidad de Distrital de Santilla. Donde acuerdan la meta del sprint.

Meta de primer sprint:

- ✓ Desarrollo de la parte de login
- ✓ Desarrollo de la parte de registrador de usuario
- 10:30am – 11:30am. El equipo de Scrum analiza y da un tiempo estimado, también divide los elementos de acuerdo a la experiencia de Scrum team, como la asignación de responsabilidad. En este sprint 1 se elaboró un conjunto de historias de usuarios concorde a la prioridad como: hu01 – loguearse, hu02 – salir de login, hu03 – registrar usuario, hu04 – modificar usuario y hu05 – eliminar usuario. También se asigna a cada incidencia la prioridad y puntos de historia para hacer el seguimiento de cada actividad en la tabla de sprint como muestra en la Figura 34.

Figura 34

Lista del tablero sprint 1



Nota. Fuente elaboración propia, generado con Jira Software.

La reunión dado en el primer sprint es para formular un conjunto de planificaciones, ya que una mala planificación podría ocasionar el incumplimiento de un sprint o un bug que puede ser perjudicial en los siguientes sprint, teniendo como consecuencia falta de entrega en hora establecida o bien disgustos en el sistema. Por lo que se establece un tablero de sprint para organizar adecuadamente todas las actividades elegidas como se muestra en la Figura 35.

Figura 35

Tablero sprint 1



Nota. Fuente elaboración propia, generado con Jira Software.

En la Figura 35 el tablero sprint para una buena planificación se muestra en tres columnas “en hacer”, “en curso” y “listo”, teniendo en cuenta que cada actividad comienza desde la primera columna, transcurrido las actividades según el desarrollo del sistema se mueve hasta culminar. Hu01-loguearse y hu02-salir login con su responsable Yonny Huarancca Ñaupari con una prioridad alta de 10 ya

culmino de hacer. El hu03-registrar usuario y hu04-modificar usuario con responsable Pabel Bautista Acuña está en curso, significa que está desarrollando las dos actividades, pero el hu05-eliminar usuario con prioridad 7 no muy importante está recién en la cola de inicio. Esto significa al momento de pasar de una columna a otro el desarrollador desliza el cuadro de Historia de Usuario a la siguiente columna, así hasta finalizar el proceso siempre cuando haya terminado de desarrollar su parte.

Al terminar el sprint 1 se busca estimar en lo siguiente: se debe finalizar con un total de puntos de historia de 42.

B. Segunda reunión de planificación (SPRINT 2)

Fecha: 02/09/2019

Hora: 9:30am -11:30am

Lugar: empresa Ennova m&f s.a.c.

Próxima reunión: 21/09/2019

- 9:30am -10:00am. El equipo Scrum proporciona al cliente (secretaria de la Municipalidad) los entregables de producto de las metas de sprint 1 acordadas en la reunión anterior. También el cliente verifica acuerdo a lo establecido, en este caso el cliente da visto bueno, por lo que fue favorable.
- 10:00am -10:30am. El representante de la municipalidad (cliente) y los involucrados en el desarrollo de la implementación del Sistema de Información de Recaudación Tributaria de la Municipalidad de Distrital de Santilla, donde acuerdan la meta del sprint 2.

Meta del segundo sprint:

- ✓ Desarrollo del contribuyente urbano
- 10:30am – 11:30am. El equipo de Scrum analiza y da un tiempo estimado, también divide los elementos de acuerdo a la experiencia de Scrum team, como la asignación de responsabilidad. En este sprint 2 donde el equipo selecciona las historias que se incluirán concorde a la prioridad como:

- ✓ hu07 – agregar un nuevo contribuyente urbano
- ✓ hu08 – buscador de contribuyente usuario
- ✓ hu09 – editar/eliminar contribuyente urbano
- ✓ hu10 – abrir un contribuyente urbano
- ✓ hu11 – visualizar abrir persona urbana
- ✓ hu12 – crear propiedad de zona urbano
- ✓ hu13 – crear año para predio urbano
- ✓ hu14 – crear nuevo predio urbano
- ✓ hu15 – generar lista de pago urbano
- ✓ hu16 – cerrar/minimizar la vista de abrir persona urbana

También se asigna a cada incidencia el responsable, prioridad y puntos de historia para hacer el seguimiento de cada actividad en la tabla de sprint como muestra en la Figura 36 donde el redondo de color azul debe realizar Robert y el azul oscuro Manrique sus respectivas tareas.

Figura 36

Lista del tablero sprint 2

86 0 0 Planificar sprint ...

▼ Tablero Sprint 2 10 incidencias
desarrollo del contribuyente urbano
02/sep/19 9:30 AM - 21/sep/19 9:30 AM

R M ...

■ HU07 - AGREGAR NUEVO CONTRIBUYENTE URBANO	RELEASE 1 - SIRT BA...	contribuyente urbano	R	SIS-14	↑	10
■ HU08 - BUSCADOR DE CONTRIBUYENTE URBANO	RELEASE 1 - SIRT BA...	contribuyente urbano	R	SIS-15	↑	8
■ HU09 - EDITAR/ELIMINAR CONTRIBUYENTE URBANO	RELEASE 1 - SIRT BA...	contribuyente urbano	R	SIS-16	↑	9
■ HU10 - ABRIR UN CONTRIBUYENTE URBANO	RELEASE 1 - SIRT BA...	contribuyente urbano	R	SIS-17	↑	10
■ HU11 - VISUALIZAR ABRIR PERSONA URBANA	RELEASE 2 - SIRT LO ...	contribuyente urbano	R	SIS-18	↑	7
■ HU12 - CREAR PROPIEDAD DE ZONA URBANA	RELEASE 2 - SIRT LO ...	contribuyente urbano	M	SIS-19	↑	10
■ HU13 - CREAR AÑO PARA PREDIO URBANO	RELEASE 2 - SIRT LO ...	contribuyente urbano	M	SIS-20	↑	10
■ HU14 - CREAR NUEVO PREDIO URBANO	RELEASE 2 - SIRT LO ...	contribuyente urbano	M	SIS-21	↑	10
■ HU15 - GENERAR LISTA DE PAGO URBANO	RELEASE 2 - SIRT LO ...	contribuyente urbano	M	SIS-22	↑	7
■ HU16 - CERRAR/MINIMIZAR LA VISTA ABRIR PERSONA URBANA	RELEASE 3 - SIRT ME...	contribuyente urbano	M	SIS-23	↓	5

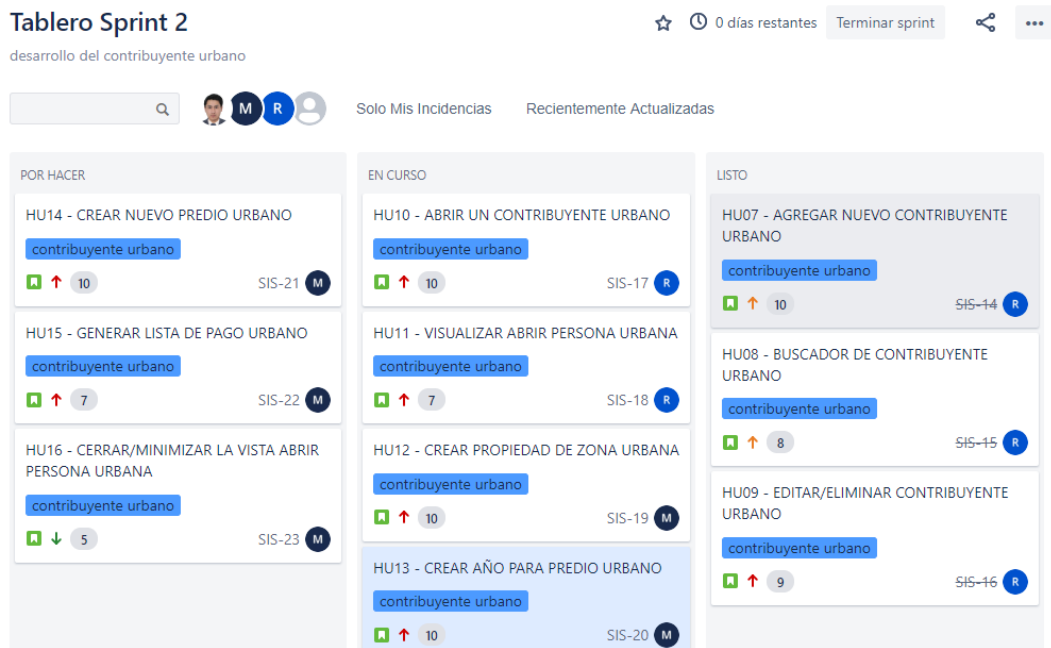
Nota. Fuente elaboración propia, generado con Jira Software.

La reunión dado en el segundo sprint es para formular un conjunto de planificaciones, después del terminado la primera sprint, por lo que una mala planificación podría ocasionar el incumplimiento del sprint 2 o general un bug, teniendo como consecuencia falta de entrega en hora establecida o falta de

continuidad al desarrollo del sistema. Por lo que se establece un tablero de sprint para organizar adecuadamente todas las actividades elegidas como se muestra en la Figura 37.

Figura 37

Tablero sprint 2



Nota. Fuente elaboración propia, generado con Jira Software.

En la Figura 37 es el tablero sprint 2 que consiste de igual manera de planificar, teniendo en cuenta que cada actividad comienza desde la primera columna, transcurrido las actividades según el desarrollo del sistema se mueve hasta culminar. Donde hu07 – agregar un nuevo contribuyente urbano, hu08 – buscador de contribuyente usuario y hu09 – editar/eliminar contribuyente urbano con su responsable Roberth Arete Palomino con una puntuación de historia de 10, 8 y 9 ya culminó satisfactoriamente. En tanto hu10 – abrir un contribuyente urbano y hu11 – visualizar abrir persona urbana, con responsabilidad Roberth Arete Palomino y hu12 – crear propiedad de zona urbano y hu13 – crear año para predio urbano, con responsabilidad Manrique Meneses Yaranga significa que estas actividades están desarrollo, mientras hu14 – crear nuevo predio urbano, hu15 – generar lista de pago urbano y hu16 – cerrar/minimizar la vista de abrir persona urbana con responsable Manrique Meneses Yaranga no muy importante está recién

en la cola de inicio. Por lo que en este parte de sprint 1 el scrum master analiza y define que hay 3 tareas terminados, 4 en proceso y 3 que falta comenzar.

Y al terminar el sprint 2 se busca estimar en lo siguiente: se debe finalizar con un total de puntos de historia de 86.

C. Tercera reunión de planificación (SPRINT 3)

Fecha: 23/sep/19 11:30 AM

Hora: 11:30 AM - 01:30 PM

Lugar: empresa Ennova m&f s.a.c.

Próxima reunión: 9/oct/19

- 11:30am - 12:00m. El equipo Scrum proporciona al cliente (secretaria de la Municipalidad) los entregables de producto de las metas de sprint 2 acordadas en la reunión anterior. También el cliente verifica acuerdo a lo establecido, en este caso el cliente da visto bueno que es favorable.
- 12:00 m - 12:30 pm. El representante de la municipalidad (cliente) y los involucrados en el desarrollo de la implementación del Sistema de Información de Recaudación Tributaria de la Municipalidad de Distrital de Santilla, donde acuerdan la meta del sprint 3.

Meta del tercer sprint:

- ✓ Desarrollo del contribuyente rural o rustico
- 12:30 pm – 01:30 pm. El equipo de Scrum analiza y da un tiempo estimado, también divide los elementos de acuerdo a la experiencia de Scrum team, como la asignación de responsabilidad a cada actividad. En este sprint 3 se selecciona un conjunto de historias de usuarios concorde a la prioridad como:
 - ✓ Hu17 – agregar un nuevo contribuyente rural o rustico
 - ✓ Hu18 – buscador de contribuyente rural o rustico
 - ✓ Hu19 – editar/eliminar contribuyente rural o rustico
 - ✓ Hu20 – abrir un contribuyente rural o rustico
 - ✓ Hu21 – visualizar abrir persona rural o rustico

- ✓ Hu22 – crear propiedad de zona rural o rustico
- ✓ Hu23 – crear año para predio rural o rustico
- ✓ Hu24 – crear nuevo predio rural o rustico
- ✓ Hu25 – generar lista de pago rural o rustico
- ✓ Hu26 – cerrar/minimizar la vista de abrir persona urbana

También se asigna a cada incidencia el responsable, prioridad y puntos de historia para hacer el seguimiento de cada actividad en la tabla de sprint como muestra en la Figura 38 como vemos el desarrollador yonny tiene 3 tareas, Elvis 2 tareas y Manrique 5 tareas o Historias de Usuarios.

Figura 38

Lista del tablero sprint 3

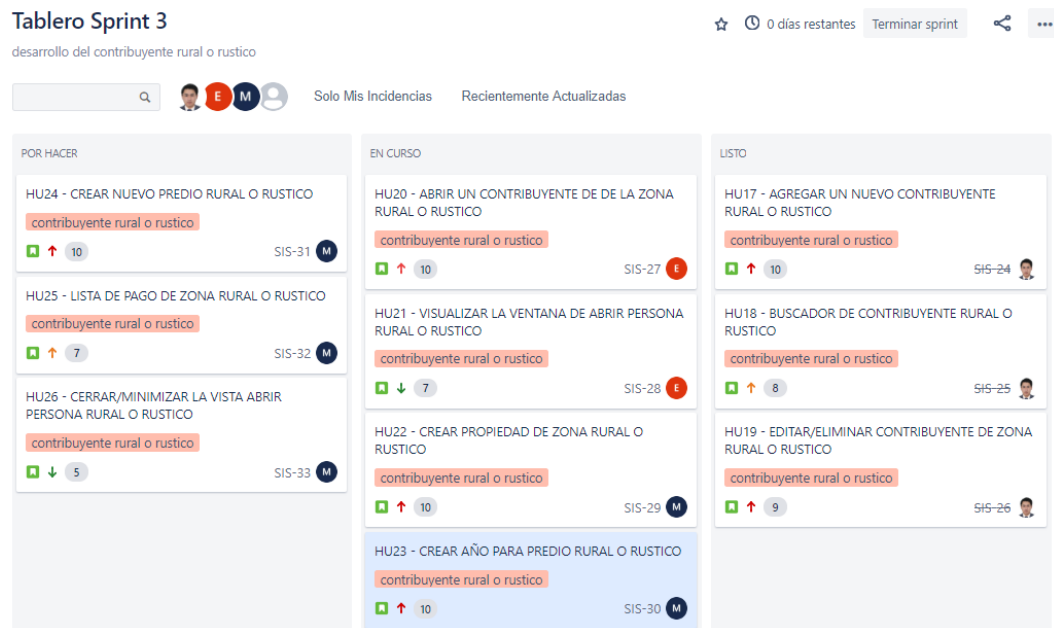
ID	Título	Release	Asignado a	Historias
HU17	AGREGAR UN NUEVO CONTRIBUYENTE RURAL O RUSTICO	RELEASE 1 - SIRT BA...	conTRIBUYENTE rural o r...	10
HU18	BUSCADOR DE CONTRIBUYENTE RURAL O RUSTICO	RELEASE 1 - SIRT BA...	conTRIBUYENTE rural o r...	8
HU19	EDITAR/ELIMINAR CONTRIBUYENTE DE ZONA RURAL O RU	RELEASE 1 - SIRT BA...	conTRIBUYENTE rural o r...	9
HU20	ABRIR UN CONTRIBUYENTE DE DE LA ZONA RURAL O RUS'	RELEASE 1 - SIRT BA...	conTRIBUYENTE rural o r...	10
HU21	VISUALIZAR LA VENTANA DE ABRIR PERSONA RURAL O RU	RELEASE 2 - SIRT LO ...	conTRIBUYENTE rural o r...	7
HU22	CREAR PROPIEDAD DE ZONA RURAL O RUSTICO	RELEASE 2 - SIRT LO ...	conTRIBUYENTE rural o r...	10
HU23	CREAR AÑO PARA PREDIO RURAL O RUSTICO	RELEASE 2 - SIRT LO ...	conTRIBUYENTE rural o r...	10
HU24	CREAR NUEVO PREDIO RURAL O RUSTICO	RELEASE 2 - SIRT LO ...	conTRIBUYENTE rural o r...	10
HU25	LISTA DE PAGO DE ZONA RURAL O RUSTICO	RELEASE 2 - SIRT LO ...	conTRIBUYENTE rural o r...	7
HU26	CERRAR/MINIMIZAR LA VISTA ABRIR PERSONA RURAL O RI	RELEASE 3 - SIRT ME...	conTRIBUYENTE rural o r...	5

Nota. Fuente elaboración propia, generado con Jira Software.

La reunión dado en el tercer sprint es para formular un conjunto de planificaciones, después del terminado la segunda sprint, tratando de evitar los bugs que pueden ocasionar como la falta del desarrollo de alguna tarea del anterior sprint, por lo que se debe planificar igual en este sprint 3, si una mala planificación puede dar como consecuencia también la falta de entrega en hora establecida o falta de continuidad al desarrollo del sistema. Por lo que se establece un tablero de sprint para organizar adecuadamente todas las actividades elegidas como se muestra en la Figura 39 que el desarrollador yonny es el redondo con icono de la foto, Elvis es del redondo de color rojo y Manrique de color azul oscuro.

Figura 39

Tablero sprint 3



Nota. Fuente elaboración propia, generado con Jira Software.

En la Figura 39 de igual manera el tablero sprint 3 para una buena planificación se muestra en tres columnas “en hacer”, “en curso” y “listo”, teniendo en cuenta que cada actividad comienza desde la primera columna, transcurrido las actividades según el desarrollo del sistema se mueve hasta culminar. Donde hu17 – agregar un nuevo contribuyente rural o rustico, hu18 – buscador de contribuyente rural o rustico y hu19 – editar/eliminar contribuyente rural o rustico con su responsable Yonny Huaranca Ñaupari con una puntuación de historia de 10, 8 y 9 ya culminó satisfactoriamente. En tanto hu20 – abrir un contribuyente rural o rustico y hu21 – visualizar abrir persona rural o rustico, con responsabilidad Elvis Teneo Morales y hu22 – crear propiedad de zona rural o rustico y hu23 – crear año para predio rural o rustico, con responsabilidad Manrique Meneses Yaranga significa que estas actividades están desarrollo, mientras hu24 – crear nuevo predio rural o rustico, hu25 – generar lista de pago rural o rustico y hu26 – cerrar/minimizar la vista de abrir persona rural o rustico con responsable Manrique Meneses Yaranga no muy importante está recién en la cola de inicio. Donde en este parte el scrum master analiza y concluye que yonny termino sus tres tareas, tanto Elvis y Manrique

con 2 tareas cada uno están desarrollando y con tres tareas restantes Manrique que falta comenzar su tarea.

Al terminar el sprint 3 se busca estimar en lo siguiente: se debe finalizar con un total de puntos de historia de 86.

D. Cuarto reunión de planificación (SPRINT 4)

Fecha: 09/oct/19 9:00 AM

Hora: 9:00 AM- 11:00 AM

Lugar: empresa Ennova m&f s.a.c.

Próxima reunión: 23/oct/19 9:00 AM

- 09:00am - 09:30m. El equipo Scrum proporciona al cliente (secretaria de la Municipalidad) los entregables de producto de las metas de sprint 3 acordadas en la reunión anterior. También el cliente verifica acuerdo a lo establecido, en este caso el cliente da visto bueno y está satisfecho.
- 9:30 m - 10:00 pm. El representante de la municipalidad (cliente) y los involucrados en el desarrollo de la implementación del Sistema de Información de Recaudación Tributaria de la Municipalidad de Distrital de Santilla, donde acuerdan la meta del sprint 4.

Metas del cuarto sprint:

- ✓ Desarrollo de sistema de caja
- ✓ Desarrollo de sistema reportes
- ✓ Desarrollo de sistema de la parte norma legal
- ✓ Desarrollo de una parte del sistema de registro de usuarios.
- 12:30 pm – 01:30 pm. El equipo de Scrum analiza y da un tiempo estimado de dos semanas y 1 día, también divide los elementos de acuerdo a la experiencia de Scrum team, como la asignación de responsabilidad a cada actividad. En este sprint 4 se selecciona las últimas historias de usuarios concorde a la prioridad como:
 - ✓ Hu6 – visualización de usuario
 - ✓ Hu27 – buscar lista de contribuyente

- ✓ Hu28 – visualización del listado de propiedad
- ✓ Hu29 – general el pago del impuesto predial
- ✓ Hu30 – visualización de reporte urbano y rural
- ✓ Hu31 – visualización de documento
- ✓ Hu32 – visualización de ayuda

También se asigna a cada incidencia el responsable, prioridad y puntos de historia para hacer el seguimiento de cada actividad en la tabla de sprint como muestra en la Figura 40 que en este sprint participan 4 desarrolladores juntamente con el scrum master, como Yonny (fondo de foto), Elvis (rojo), Robert (azul) y Pavel (azul claro).

Figura 40

Lista del tablero sprint 4

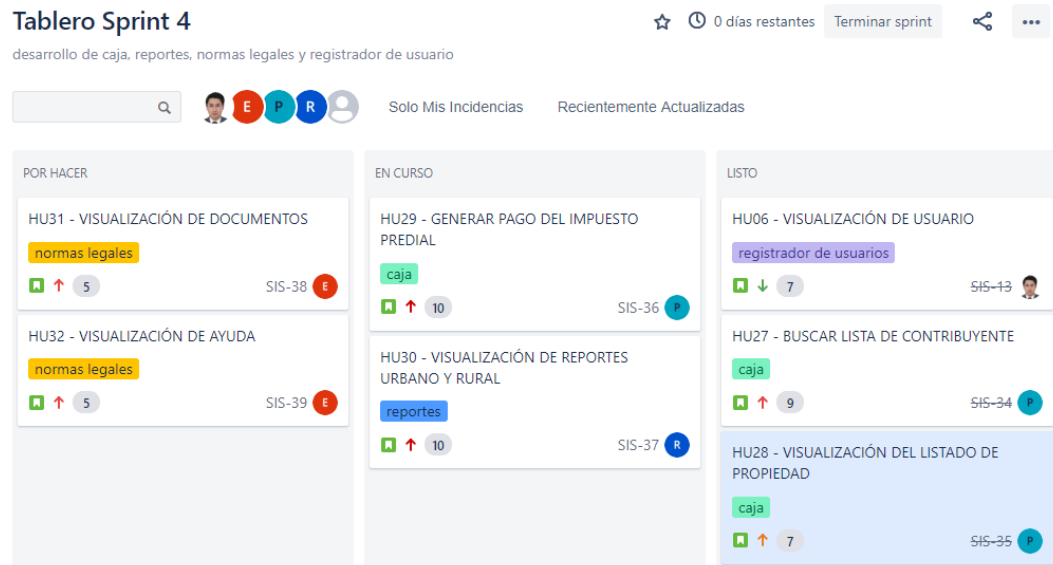


Nota. Fuente elaboración propia, generado con Jira Software.

La reunión dado en el cuarto sprint o el ultimo sprint también se formula un conjunto de planificaciones, después del terminado el tercer sprint. Se establece un tablero de sprint para organizar adecuadamente todas las actividades elegidas como se muestra en la Figura 41 en este tablero se visualiza que yonny con 1 tarea y Pavel con 2 tareas ya terminaron, mientras Robert y Pavel cada uno con una tarea están desarrollando y Elvis con dos tareas que falta por comenzar o hacer, esto se debe a que puede pasar muchos factores como la investigación de su parte para comenzar hacer o el acuerdo en la reunión de las mañanas para asignación de otras actividades que no son parte del proyecto, que después de terminar va comenzar a realizar.

Figura 41

Tablero sprint 7



Nota. Fuente elaboración propia, generado con Jira Software.

En la Figura 41 el tablero sprint 7 de igual manera se muestra para una buena planificación tres columnas “en hacer”, “en curso” y “listo”, teniendo en cuenta que cada actividad comienza desde la primera columna, transcurrido las actividades según el desarrollo del sistema se mueve hasta culminar. Donde hu06 – visualización de usuario (responsable: Yonny Huaranca Ñaupari), hu27 – buscar lista de contribuyente (responsable: Pabel Bautista Acuña) y hu28 –visualización del listado de propiedad (responsable: Pabel Bautista Acuña) con una puntuación de historia de 7, 9 y 7 ya culminó satisfactoriamente. En tanto hu29 – generar pago del impuesto predial y hu30 – visualización de reportes urbanos y rural, con responsabilidad Pabel Bautista Acuña y Roberth Areste Palomino significa que esta actividad está en desarrollo, mientras hu31 – visualización de documento, hu32 – visualización de ayuda con responsable Elvis Teneo Morales no muy importante está recién en la cola de inicio.

Al terminar el sprint 4 se busca estimar en lo siguiente: se debe finalizar con un total de puntos de historia de 53.

En cada finalización de cada sprint se suma cada punto de trabajo cumplido, así obteniendo un puntaje requerida, si en caso no se llegan a la suma exacta se debe a que algún Scrum team no cumplió con su tarea, pero en el proyecto

de alguna forma termina con la ayuda del equipo del Scrum Team (Desarrolladores de sistema) para no perjudicar el sprint y no generar algún bug.

7.3. FASE N° 3: SCRUM DIARIO (DAILY SCRUM)

A. COMUNICACIÓN DE SPRINT BACKLOG

La comunicación se basa en reuniones diarias con Scrum Master y equipo de trabajo, para ver el avance de los cuatro sprint, verificando y evaluando con cada responsable del equipo de trabajo, con la finalidad de terminar satisfactoriamente en cada tarea en los sprint.

Para la supervisión de Scrum Master (Yonny Huaranca Ñaupari) del equipo del trabajo del día a día el avance hasta la fecha de entrega, se facilitan en tablero de sprint 1, tablero de sprint 2, tablero de sprint 3 y el tablero de sprint 4. Como tal cada equipo según las fechas establecidos avanza su respectiva tarea hasta terminar “listo” de la columna del tablero sprint.

B. GRÁFICO DE TRABAJO PENDIENTE (BURNDOWN CHARTS)

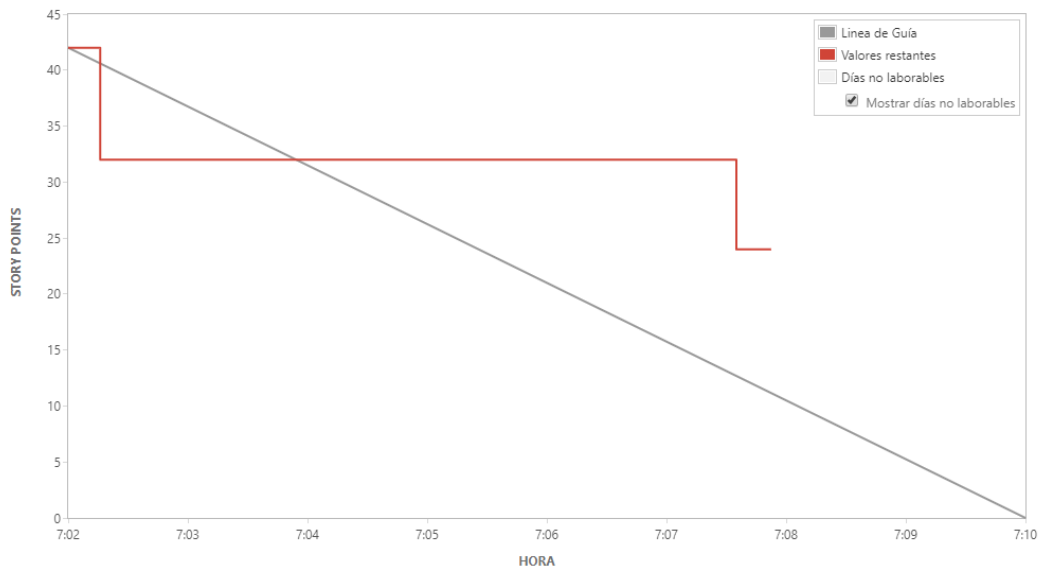
• SPRINT 1

En la Figura 42 se muestra el trabajo pendiente del sprint 1 que consiste en la entrega del desarrollo del login y registrar usuario. Teniendo en cuenta que la simulación se hace como plan estudio de esta investigación en la fecha actual (Figura 42) con una ponderación de 8 minutos, que no corresponde a fecha del que se elaboró de dos semanas, por mismo sentido que la investigación se hace culminado el desarrollo el sistema Información de Recaudación Tributaria de la Municipalidad Distrital de Santillana en el año 2019.

Figura 42

Trabajo pendiente del tablero sprint 01

Gráfica de trabajo pendiente



Nota. Fuente elaboración propia, generado con Jira Software.

Un trabajo pendiente es el restante de las tareas realizadas por Scrum team durante el tiempo de sprint 1, como se muestra en la Figura 42 que desde las 7:08 hasta 7:09 es tiempo que falta para culminar el sprint 1. Esto es muy importante tanto el Scrum master y Scrum Team puedan medir su trabajo, para que no haya retrasos.

Tabla 43

Avance del sprint 1

Fecha	Incidencia	Tipo de Evento	Detalle de Evento	Story Points		Tiempo Restante Estimado
				Incr.	Decr.	
21/abr/20 7:02 AM	SIS-10	Comienzo de Sprint		9		42
	SIS-11			8		
	SIS-12			7		
	SIS-8			10		
	SIS-9			8		
	SIS-8	Trabajo pendiente	Incidencia completada		10	32
21/abr/20 7:07 AM	SIS-9	Trabajo pendiente	Incidencia completada		8	24

Nota. Fuente elaboración propia, generado con Jira Software.

En la Figura 42 y la Tabla 43 en resumen, se muestra que el avance del sprint 1 con los responsables Yonny Hurannca Ñaupari y Pabel Bautista Acuña son encargados de realizar las tareas de login del sistema y registrar usuario desde las 7:02 am a 7:07 am tiempo transcurrido del avance de las tareas como muestra en la

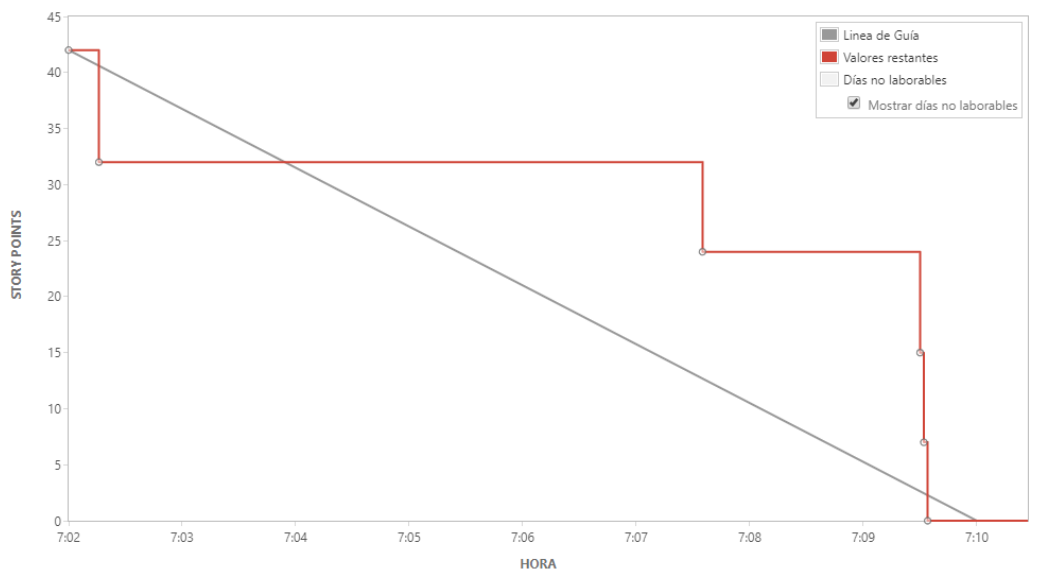
Figura 51, Figura 52 y Figura 53 donde ya se completó dos incidencias HU01 – loguearse y HU02 – salir de login con puntuación de historia 10 y 8 finalizados, del total de 42 puntuaciones, mientras las demás incidencias están haciéndose o bien en cola de inicio. Teniendo en cuenta que falta hacer 24 puntos para culminar todas las tareas.

Figura 43

Trabajo culminado del tablero sprint 01

Proyectos / SistemaRT / Tablero SIS / Informes

Gráfica de trabajo pendiente



Nota. Fuente elaboración propia, generado con Jira Software.

En la Figura 43 se muestra que el sprint 1 se finaliza satisfactoriamente todas las incidencias con un punto de historia de 42, en la fecha establecida de 21/abr/20 7:02 am a 21/abr/20 7:09 am, demostrando que el equipo está comprometido con el proyecto Sistema de Información de Recaudación Tributaria de la Municipalidad Distrital de Santillana.

Tabla 44

Finalizada del sprint 1

Fecha	Incidencia	Tipo de Evento	Detalle de Evento	Story Points		Tiempo Restante Estimado
				Incr.	Descr.	
21/abr/20 7:02 AM	SIS-10	Comienzo de Sprint		9		42
	SIS-11			8		
	SIS-12			7		
	SIS-8			10		
	SIS-9			8		
21/abr/20 7:07 AM	SIS-8	Trabajo pendiente	Incidencia completada		10	32
	SIS-9	Trabajo pendiente	Incidencia completada		8	24
21/abr/20 7:09 AM	SIS-10	Trabajo pendiente	Incidencia completada		9	15
	SIS-11	Trabajo pendiente	Incidencia completada		8	7
	SIS-12	Trabajo pendiente	Incidencia completada		7	0

Nota. Fuente elaboración propia, generado con Jira Software.

Los trabajos pendientes se finalizaron, completando las incidencias con un total de 42 puntos en las fechas establecidas, con tareas cada uno como: 10, 8, 9, 8 y 7 como se muestra en Tabla 44.

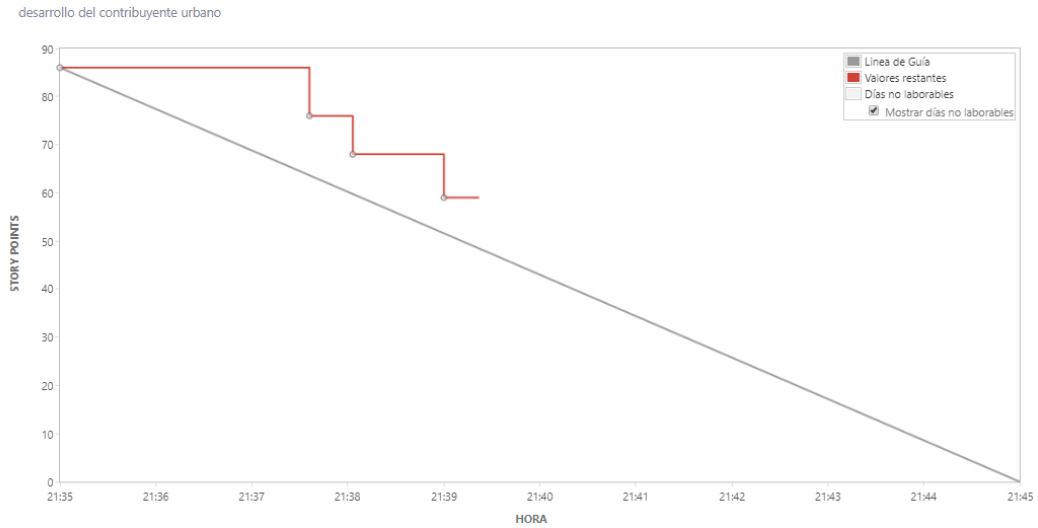
- **SPRINT 2**

Después del terminado el sprint 1 se proseguí con siguiente actividad de sprint 2, tomando en cuenta que es secuencial. En la Figura 44 se muestra el trabajo pendiente del sprint 2 que consiste en la entrega del desarrollo del contribuyente urbano. Teniendo en cuenta que la simulación se hace como plan estudio de esta investigación en la fecha actual (Figura 44), en este estudio de investigación se simula con una ponderación de 10 minutos de 9:35 pm a 9:45 pm. En la gráfica se muestra que descende la pendiente de forma escalada, significa que está siendo apagado cada incidencia por Scrum Team, que es lo mismo decir terminando su tarea o su Historia de usuario en el tablero de Scrum. Pero en el grafico falta finalizar como se muestra la línea roja.

Figura 44

Trabajo pendiente del tablero sprint 2

Gráfica de trabajo pendiente



Nota. Fuente elaboración propia, generado con Jira Software.

En el avance del sprint 2 de la Tabla 45 se visualiza que se completó tres incidencias o tareas con puntuaciones de 10, 8 y 9 puntos, a más detalle se describe en la siguiente hoja.

Tabla 45

Avance de sprint 2

Fecha	Incidencia	Tipo de Evento	Detalle de Evento	Story Points	
				Incr. puntos	Tiempo Restante Estimado
21/abr/20 9:35 PM	SIS-14 SIS-15 SIS-16 SIS-17 SIS-18 SIS-19 SIS-20 SIS-21 SIS-22 SIS-23	Comienzo de Sprint		10 8 9 10 7 10 10 10 7 5	86
21/abr/20 9:37 PM	SIS-14	Trabajo pendiente	Incidencia completada	10	76
21/abr/20 9:38 PM	SIS-15	Trabajo pendiente	Incidencia completada	8	68
21/abr/20 9:39 PM	SIS-16	Trabajo pendiente	Incidencia completada	9	59

Nota. Fuente elaboración propia, generado con Jira Software.

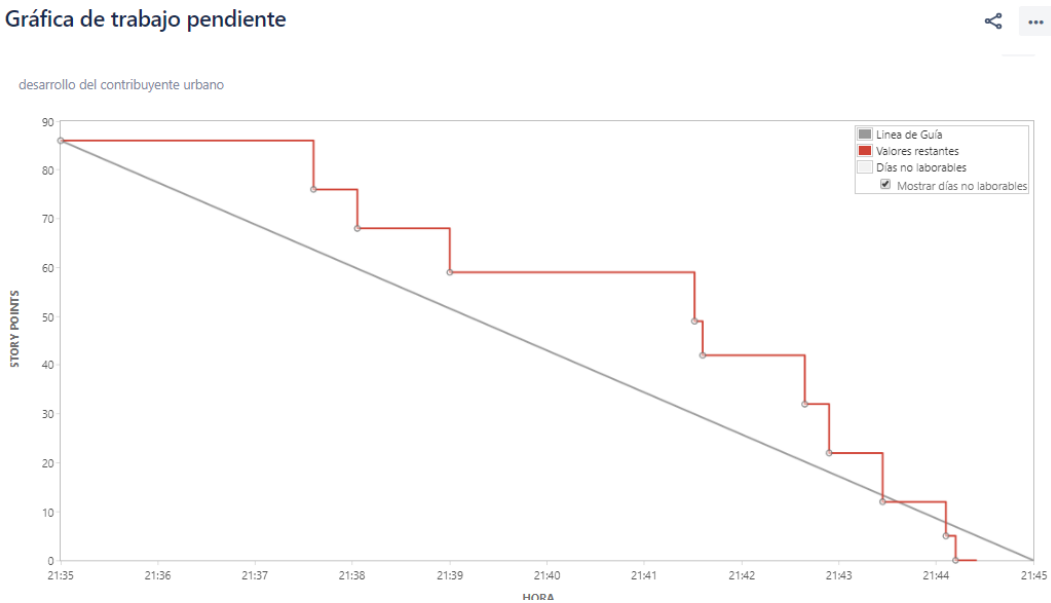
En la Figura 44 y en la Tabla 45 en resumen, se muestra que el avance del sprint 2 con los responsables Roberth Areste Palomino y Manrique Meneses Yaranga son encargados de realizar el desarrollo del contribuyente urbano desde las 9:35 pm a 9:45 pm. am total de tiempo que debe transcurrir, pero en primer análisis de la Figura 44 y de la Figura 44 se muestra un avance de acumulación de

27 puntos de historia desde 9:35 pm hasta 9:39 pm. Esta acumulación consta de la finalización de tres incidencias hu07 – agregar un nuevo contribuyente urbano, hu08 – buscador de contribuyente urbano y hu09 – editar/eliminar contribuyente urbano con su responsable Roberth Areste Palomino con una puntuación de historia de 10, 8 y 9.

Figura 45

Trabajo culminado del tablero sprint 2

Gráfica de trabajo pendiente



Nota. Fuente elaboración propia, generado con Jira Software.

Tabla 46

Finalizada del sprint 2

Fecha	Incidencia	Tipo de Evento	Detalle de Evento	Incr.	Decr.	Tiempo Restante Estimado
21/abr/20 9:35 PM	SIS-14 SIS-15 SIS-16 SIS-17 SIS-18 SIS-19 SIS-20 SIS-21 SIS-22 SIS-23	Comienzo de Sprint		10 8 9 10 7 10 10 10 7 5		86
21/abr/20 9:37 PM	SIS-14	Trabajo pendiente	Incidencia completada	10		76
21/abr/20 9:38 PM	SIS-15	Trabajo pendiente	Incidencia completada	8		68
21/abr/20 9:39 PM	SIS-16	Trabajo pendiente	Incidencia completada	9		59
21/abr/20 9:41 PM	SIS-17 SIS-18	Trabajo pendiente	Incidencia completada Incidencia completada	10 7		49 42
21/abr/20 9:42 PM	SIS-21 SIS-19	Trabajo pendiente	Incidencia completada Incidencia completada	10 10		32 22
21/abr/20 9:43 PM	SIS-20	Trabajo pendiente	Incidencia completada	10		12
21/abr/20 9:44 PM	SIS-22 SIS-23	Trabajo pendiente	Incidencia completada Incidencia completada	7 5		5 0

Nota. Fuente elaboración propia, generado con Jira Software.

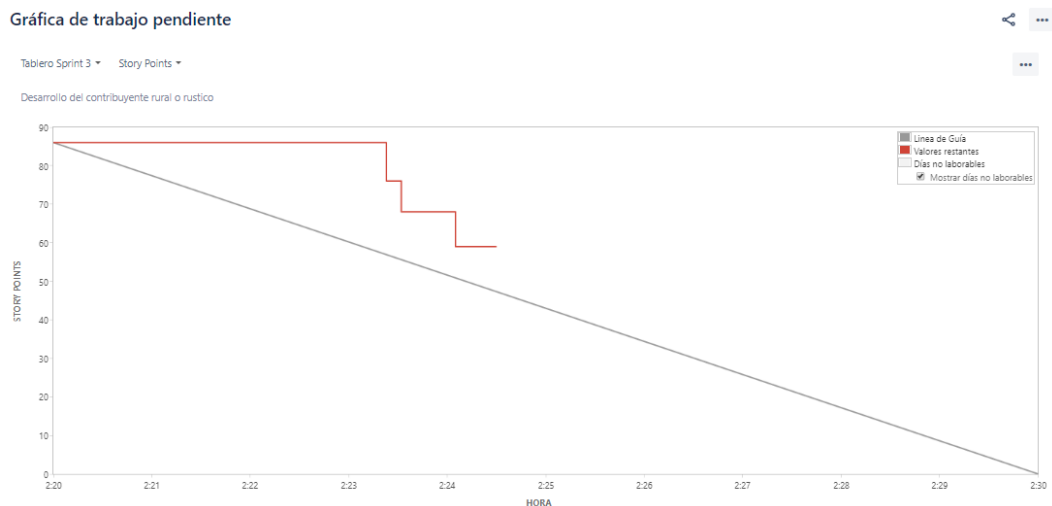
En la Figura 45 y Tabla 46 se muestra que el sprint 2 se finaliza satisfactoriamente todas las incidencias con un punto de historia de 86, en la fecha establecida de 21/abr/20 9:35 pm a 21/abr/20 9:45 pm según la simulación, demostrando el desarrollo de la tarea de “contribuyente urbano”, que a la vez que el equipo está comprometido con el proyecto Sistema de Información de Recaudación Tributaria de la Municipalidad Distrital de Santillana.

- **SPRINT 3**

Terminado con el sprint 2 se sigue con sprint 3 del desarrollo de contribuyente rural o rustico. En la Figura 46 se muestra el trabajo pendiente del sprint 3 que consiste en la entrega del desarrollo del contribuyente rustico o rural. Teniendo en cuenta que la simulación se hace como plan estudio de esta investigación en la fecha actual (Figura 46), en este estudio de investigación se simula con una ponderación de 10 minutos de 2:20 am a 2:30 am.

Figura 46

Trabajo pendiente del tablero sprint 3



Nota. Fuente elaboración propia, generado con Jira Software.

En el sprint 3 el trabajo pendiente como se muestra en la Figura 46 es desde 2:24 a 2:30, eso es el tiempo que falta realizar las tareas (Historias de Usuarios) por parte de los desarrolladores.

Tabla 47

Avance del sprint 3

Fecha	Incidencia	Tipo de Evento	Detalle de Evento	Story Points	
				Incr.	Decr.
22/abr/20 2:20 AM	SIS-24 SIS-25 SIS-26 SIS-27 SIS-28 SIS-29 SIS-30 SIS-31 SIS-32 SIS-33	Comienzo de Sprint		10 8 9 10 7 10 10 10 7 5	86
22/abr/20 2:23 AM	SIS-24	Trabajo pendiente	Incidencia completada	10	76
	SIS-25	Trabajo pendiente	Incidencia completada	8	68
22/abr/20 2:24 AM	SIS-26	Trabajo pendiente	Incidencia completada	9	59

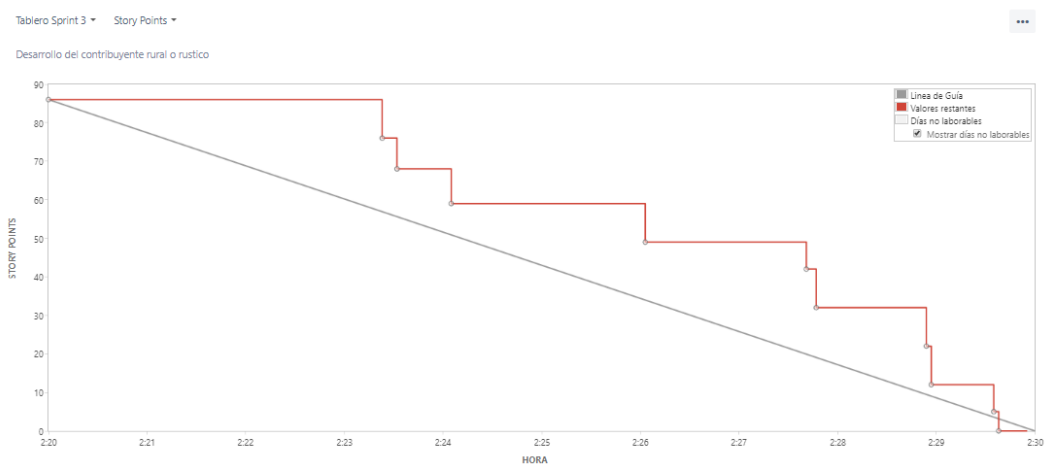
Nota. Fuente elaboración propia, generado con Jira Software.

En la Figura 46 en resumen, se muestra que el avance del sprint 3 con los responsables Yonny Huaranca Ñaupari, Elvis Teneo Morales y Manrique Meneses Yaranga son encargados de realizar el desarrollo del contribuyente rural o rustico desde las 2:20 am a 2:30 am. en total de tiempo que debe transcurrir. En el primer análisis de la Figura 46 y en la Figura 46 se muestra un avance de acumulación de 27 puntos de historia desde 2:20 am hasta 2:24 am. Esta acumulación consta de la finalización de tres incidencias hu17 – agregar un nuevo contribuyente rural o rustica, hu18 – buscador de contribuyente rural o rustica y hu19 – editar/eliminar contribuyente rural o rustica con su responsable Yonny Huaranca Ñaupari con una puntuación de historia de 10, 8 y 9 (Tabla 47).

Figura 47

Trabajo culminado del tablero sprint 3

Gráfica de trabajo pendiente



Nota. Fuente elaboración propia, generado con Jira Software.

El trabajo culminó en 2:30 completando todas las metas que se planificó en este sprint 3, como se muestra en la Figura 47 con este resultado podemos pasar al siguiente sprint 4. Podemos tomar como anecdótico que algún momento del proceso del sprint se puede terminar en un tiempo anterior del estipulado, por lo que para completar el tiempo de sprint se debe testear las tareas hasta fecha que se planificó el sprint 3. En esta parte también se desarrolla el contribuyente rural o rústico.

Tabla 48

Finalizada del sprint 3

Fecha	Incidencia	Tipo de Evento	Detalle de Evento	Story Points		Tiempo Restante Estimado
				Incr.	Descr.	
22/abr/20 2:20 AM	SIS-24 SIS-25 SIS-26 SIS-27 SIS-28 SIS-29 SIS-30 SIS-31 SIS-32 SIS-33	Comienzo de Sprint		10 8 9 10 7 10 10 10 7 5		86
22/abr/20 2:23 AM	SIS-24	Trabajo pendiente	Incidencia completada	10		76
	SIS-25	Trabajo pendiente	Incidencia completada	8		68
22/abr/20 2:24 AM	SIS-26	Trabajo pendiente	Incidencia completada	9		59
22/abr/20 2:26 AM	SIS-27	Trabajo pendiente	Incidencia completada	10		49
22/abr/20 2:27 AM	SIS-28	Trabajo pendiente	Incidencia completada	7		42
	SIS-29	Trabajo pendiente	Incidencia completada	10		32
22/abr/20 2:28 AM	SIS-30	Trabajo pendiente	Incidencia completada	10		22
	SIS-31	Trabajo pendiente	Incidencia completada	10		12
22/abr/20 2:29 AM	SIS-32	Trabajo pendiente	Incidencia completada	7		5
	SIS-33	Trabajo pendiente	Incidencia completada	5		0

Nota. Fuente elaboración propia, generado con Jira Software.

En la Figura 47 y la Tabla 48 se muestra que el sprint 3 se finaliza satisfactoriamente todas las incidencias con un punto de historia de 86, en la fecha establecida de 22/abr/20 2:20 am a 22/abr/30 2:30 am según la simulación, demostrando el desarrollo de la tarea de “contribuyente rural o rústico”, que a la vez que el equipo está comprometido con el proyecto Sistema de Información de Recaudación Tributaria de la Municipalidad Distrital de Santillana para el siguiente sprint 4.

- **SPRINT 4**

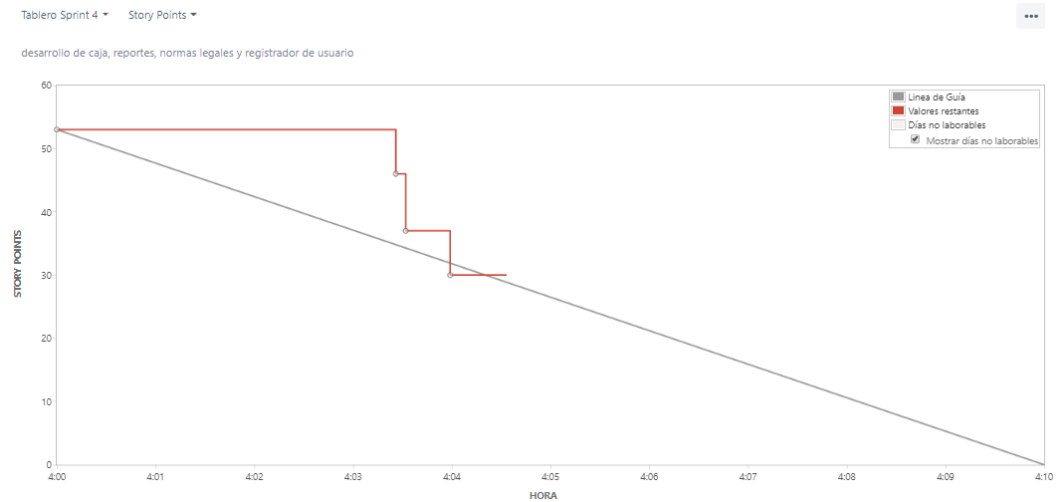
En la Figura 48 se muestra el trabajo pendiente del sprint 4 que consiste en la entrega del desarrollo de sistema de caja, desarrollo de sistema reportes, desarrollo de sistema de la parte norma legal y desarrollo de una parte del sistema

de registro de usuarios. Teniendo en cuenta que la simulación se hace como plan estudio de esta investigación en la fecha actual (Figura 48), en este estudio de investigación se simula con una ponderación de 10 minutos de 4:00 am a 4:10 am.

Figura 48

Trabajo pendiente del tablero sprint 4

Gráfica de trabajo pendiente



Nota. Fuente elaboración propia, generado con Jira Software.

El trabajo pendiente del tablero de sprint 4 es de 4:04 a 4:10, eso el tiempo estipulado que se debe terminar todas las tareas en este sprint 4, para lo cual también observamos la línea de guía de color plomo que es la trayectoria que debe recurrir el sprint 4 como se muestra en la Figura 48 en este parte se desarrolla la caja, reportes, normas legales y registro de usuarios.

Tabla 49

Avance del sprint 4

Fecha	Incidencia	Tipo de Evento	Detalle de Evento	Story Points		Tiempo Restante Estimado
				Incr.	Decr.	
22/abr/20 4:00 AM	SIS-13	Comienzo de Sprint		7		53
	SIS-34			9		
	SIS-35			7		
	SIS-36			10		
	SIS-37			10		
	SIS-38			5		
	SIS-39			5		
22/abr/20 4:03 AM	SIS-13	Trabajo pendiente	Incidencia completada		7	46
	SIS-34	Trabajo pendiente	Incidencia completada		9	37
	SIS-35	Trabajo pendiente	Incidencia completada		7	30

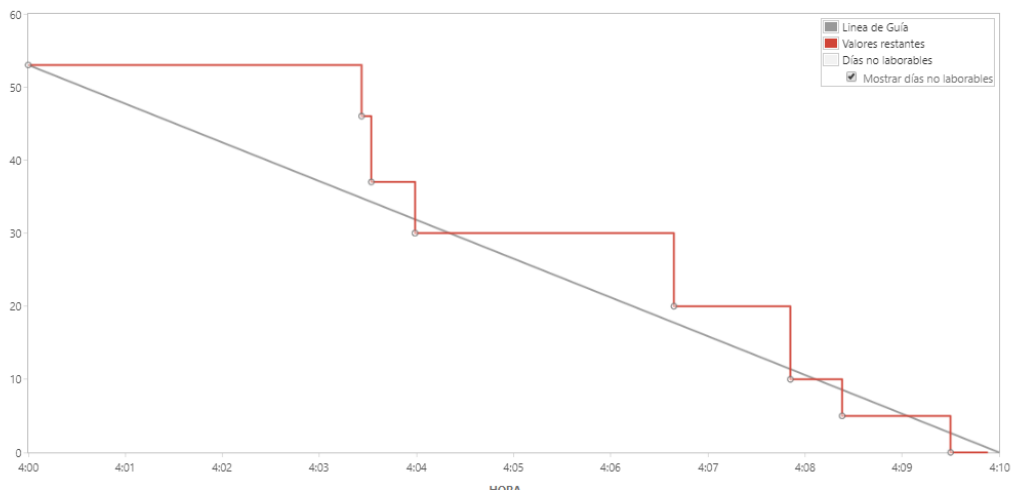
Nota. Fuente elaboración propia, generado con Jira Software.

En la Figura 48 en resumen, se muestra que el avance del sprint 4 con los responsables Yonny Huarancca Ñaupari, Elvis Teneo Morales, Pabel Bautista Acuña y Roberth Arete Palomino son encargados de realizar el desarrollo de sistema de caja, desarrollo de sistema reportes, desarrollo de sistema de la parte norma legal y desarrollo de una parte del sistema de registro de usuarios desde las 4:00 am a 4:10 am. en total de tiempo que debe transcurrir. Pero en primer análisis de la Figura 48 se muestra un avance de acumulación de 23 puntos de historia desde 4:00 am hasta 4:03 am. Esta acumulación consta de la finalización de tres incidencias Hu6 – visualización de usuario, Hu27 – buscar lista de contribuyente y Hu28 – visualización del listado de propiedad con su responsable Yonny Huarancca Ñaupari y Pabel Bautista Acuña con una puntuación de historia de 7, 9 y 7 (Tabla 49) así faltando de realizar 30 puntos.

Figura 49

Trabajo culminado del tablero sprint 4

Gráfica de trabajo pendiente



Nota. Fuente elaboración propia, generado con Jira Software.

Tabla 50

Finalizada del sprint 4

Fecha	Incidencia	Tipo de Evento	Detalle de Evento	Story Points		Tiempo Restante Estimado
				Incr.	Decr.	
2/abr/20 4:00 AM	SIS-13	Comienzo de Sprint		7		53
	SIS-34			9		
	SIS-35			7		
	SIS-36			10		
	SIS-37			10		
	SIS-38			5		
	SIS-39			5		
2/abr/20 4:03 AM	SIS-13	Trabajo pendiente	Incidencia completada		7	46
	SIS-34	Trabajo pendiente	Incidencia completada		9	37
	SIS-35	Trabajo pendiente	Incidencia completada		7	30
2/abr/20 4:06 AM	SIS-36	Trabajo pendiente	Incidencia completada		10	20
2/abr/20 4:07 AM	SIS-37	Trabajo pendiente	Incidencia completada		10	10
2/abr/20 4:08 AM	SIS-38	Trabajo pendiente	Incidencia completada		5	5
2/abr/20 4:09 AM	SIS-39	Trabajo pendiente	Incidencia completada		5	0

Nota. Fuente elaboración propia, generado con Jira Software.

En la Figura 49 y en la Tabla 50 se muestra que el sprint 4 se finaliza satisfactoriamente todas las incidencias con un punto de historia de 53, en la fecha establecida de 22/abr/20 4:00 am a 22/abr/30 4:10 am según la simulación, demostrando el desarrollo de las tareas, que a la vez que el equipo está comprometido con el proyecto Sistema de Información de Recaudación Tributaria de la Municipalidad Distrital de Santillana. Culminando satisfactoriamente todos los sprint.

7.4. FASE N° 4: REVISIÓN DE SPRINT (SPRINT REVIEW)

A. PLANIFICACIÓN DE ENTREGA

Las entregables de cada sprint al culminado todas las incidencias en la tabla de sprint, lo que se hace en esta fase para nuestra investigación del proyecto es mostrar a los encargados de la Municipalidad (cliente) de acuerdo a las reuniones realizados conforme a las metas.

- **PARA SPRINT 1**

En el sprint 1 nos muestra un reporte de todas las incidencias realizadas en el proceso, para lo cual se entrega o se muestra a la municipalidad con las metas que se acordó en la reunión.

Metas del sprint 1:

1. Desarrollo de login

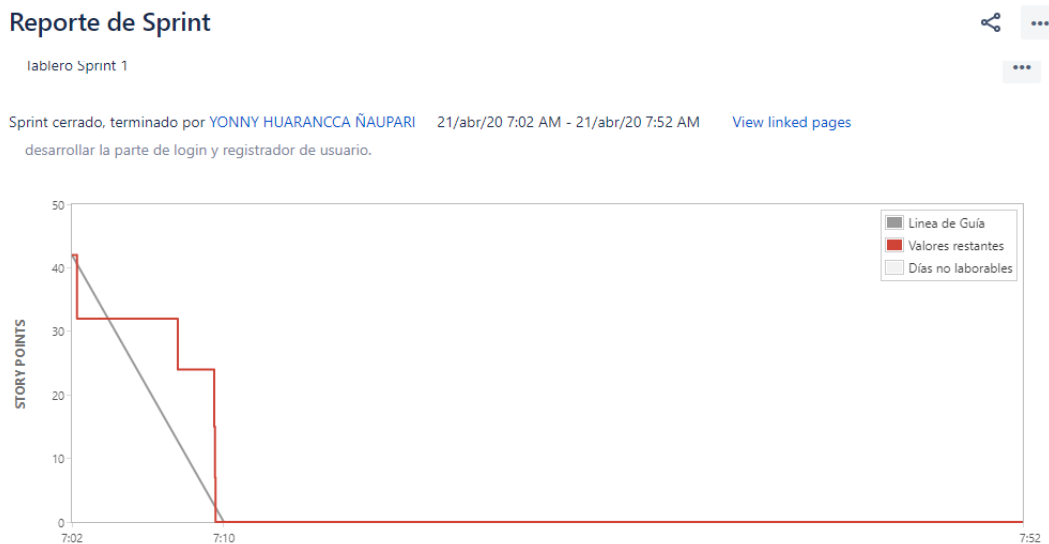
2. Desarrollo registrador de usuario

Comprende que cada trabajo se terminó o se devuelto al backlog en cada sprint, esto ayuda a determinar si el equipo está comprometido con el proyecto o hay demasiada corrupción del alcance.

En la Figura 50 se visualiza la satisfacción del trabajo antes de la fecha establecida 21/abr/20 7:02 am – 21/abr/20 7:10 am, con esto podemos analizar que los trabajadores están bien involucrados, cumpliendo total 40 puntos de historia requerida.

Figura 50

Trabajo culminado del tablero sprint



Nota. Fuente elaboración propia, generado con Jira Software.

Tabla 51

Finalizada del sprint 1

Informe de estado

Incidencias terminadas [Ver en el navegador de incidencias](#)

Clave	Resumen	Tipo de Incidencia	Prioridad	Estado	Story Points (42)
SIS-8	HU01 - LOGUEARSE	Historia	High	FINALIZADA	10
SIS-9	HU02 - SALIR DE LOGIN	Historia	Medium	FINALIZADA	8
SIS-10	HU03 - REGISTRAR USUARIO	Historia	Highest	FINALIZADA	9
SIS-11	HU04 - MODIFICAR USUARIO	Historia	High	FINALIZADA	8
SIS-12	HU05 - ELIMINAR USUARIO	Historia	Low	FINALIZADA	7

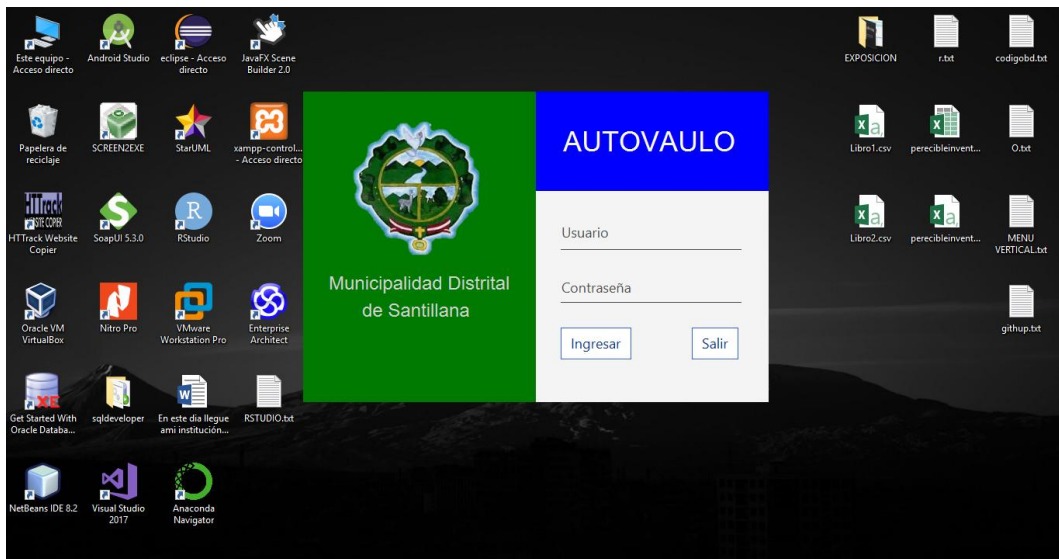
Nota. Fuente elaboración propia, generado con Jira Software.

En el reporte de la Tabla 51 se visualiza todas las incidencias finalizados según la prioridad que se lo asigno, como también la asignación de puntos de historia de cada uno de las incidencias así acumulando un total de 42 puntos que se requería para la finalización.

Tanto las dos metas (login y registrar usuario) se desarrollaron correctamente como se muestra en la Figura 51, Figura 52 y Figura 53.

Figura 51

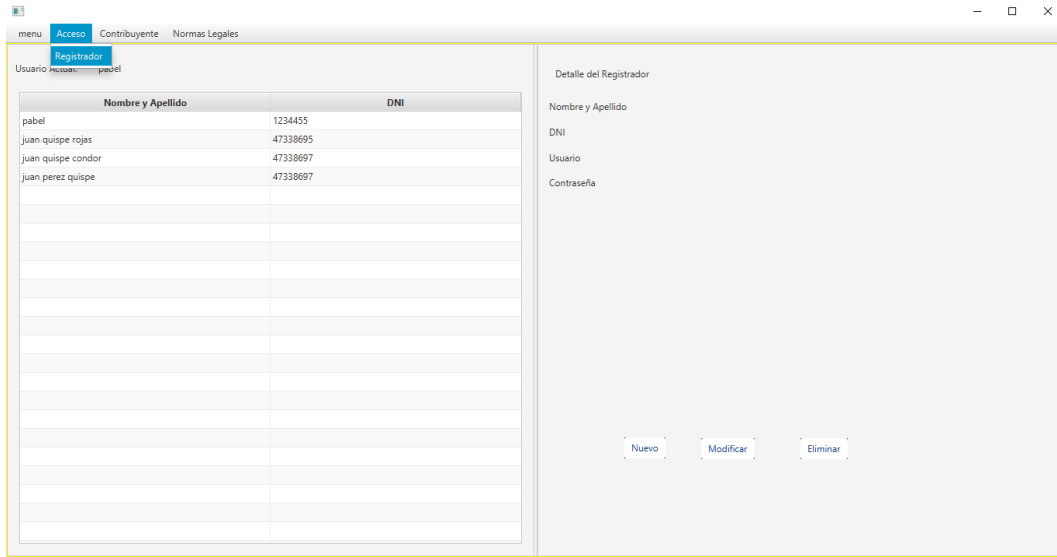
Login



Nota. Fuente elaboración propia, desarrollado con java.

Figura 52

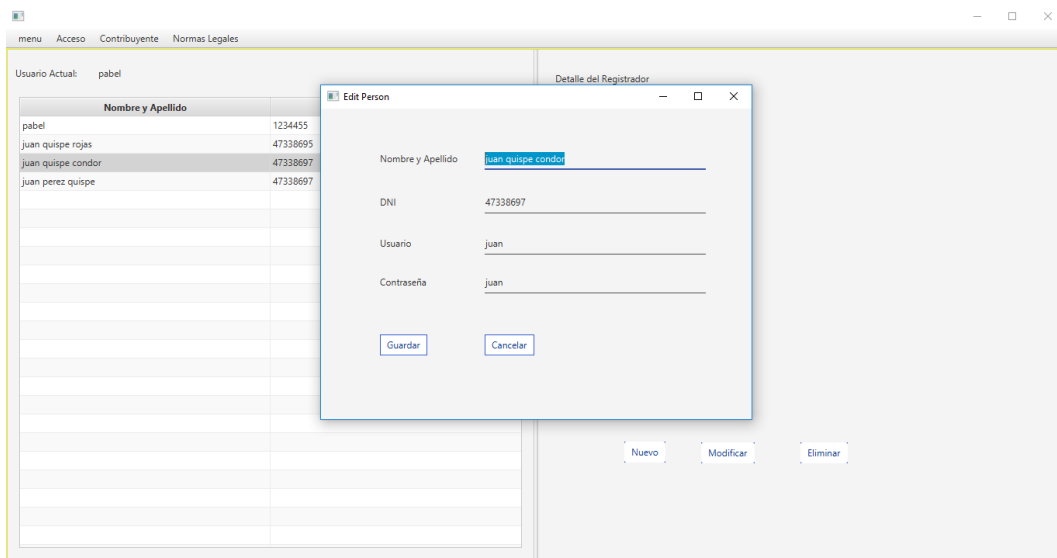
Vista de registrador de usuario



Nota. Fuente elaboración propia, desarrollado con java.

Figura 53

Registrador de usuario



Nota. Fuente elaboración propia, desarrollado con java.

Las entregas o visualizaciones al encargado de la Municipalidad acorde a la meta establecida son Figura 51, Figura 52 y Figura 53 así cumpliendo en la fecha dada de dos semanas 19/08/2019 - 02/09/2019.

- **PARA SPRINT 2**

En el segundo proceso de sprint 2 nos muestra un reporte de todas las incidencias realizadas en el proceso, para lo cual se entrega o se muestra a la municipalidad con las metas que se acordó en la reunión.

Meta del sprint 2:

1. Desarrollo del contribuyente urbano

Comprende que cada trabajo se terminó o se devuelto al backlog en cada sprint, esto ayuda a determinar si el equipo está comprometido con el proyecto o hay demasiada corrupción del alcance.

En la Figura 54 se visualiza la satisfacción del trabajo antes de la fecha establecida 21/abr/20 9:35 pm – 21/abr/20 7:45 pm según la simulación dado, con esto podemos analizar que los trabajadores están bien involucrados, cumpliendo total 86 puntos de historia requerida.

Figura 54

Trabajo culminado del tablero sprint 2



Nota. Fuente elaboración propia, generado con Jira Software.

Tabla 52

Finalizada del sprint 2

Informe de estado

Incidencias terminadas

[Ver en el navegador de incidencias](#)

Clave	Resumen	Tipo de Incidencia	Prioridad	Estado	Story Points (86)
SIS-14	HU07 - AGREGAR NUEVO CONTRIBUYENTE URBANO	Historia	Medium	FINALIZADA	10
SIS-15	HU08 - BUSCADOR DE CONTRIBUYENTE URBANO	Historia	Medium	FINALIZADA	8
SIS-16	HU09 - EDITAR/ELIMINAR CONTRIBUYENTE URBANO	Historia	Highest	FINALIZADA	9
SIS-17	HU10 - ABRIR UN CONTRIBUYENTE URBANO	Historia	Highest	FINALIZADA	10
SIS-18	HU11 - VISUALIZAR ABRIR PERSONA URBANA	Historia	Highest	FINALIZADA	7
SIS-19	HU12 - CREAR PROPIEDAD DE ZONA URBANA	Historia	Highest	FINALIZADA	10
SIS-20	HU13 - CREAR AÑO PARA PREDIO URBANO	Historia	Highest	FINALIZADA	10
SIS-21	HU14 - CREAR NUEVO PREDIO URBANO	Historia	Highest	FINALIZADA	10
SIS-22	HU15 - GENERAR LISTA DE PAGO URBANO	Historia	Highest	FINALIZADA	7
SIS-23	HU16 - CERRAR/MINIMIZAR LA VISTA ABRIR PERSONA URBANA	Historia	Low	FINALIZADA	5

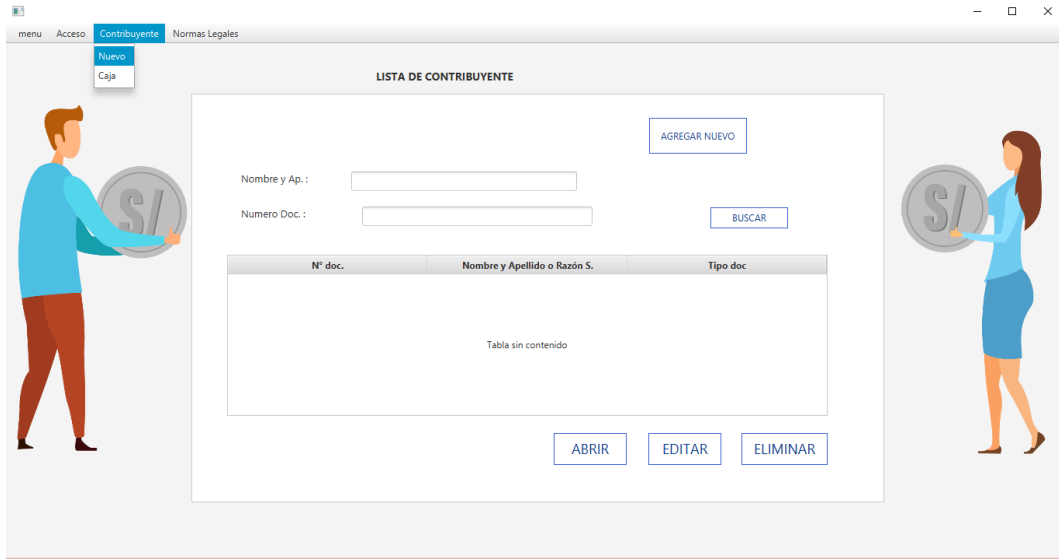
Nota. Fuente elaboración propia, generado con Jira Software.

En el reporte de la Tabla 52 se visualiza todas las incidencias finalizados según la prioridad que se lo asigno, como también la asignación de puntos de historia de cada uno de las incidencias así acumulando un total de 86 puntos que se requería para la finalización.

La meta (desarrollo del contribuyente urbano) se desarrolló correctamente como se muestra en la Figura 55, Figura 56, Figura 57, Figura 58, Figura 59 y Figura 60 para la entrega o visualización correspondiente al encargado de la municipalidad.

Figura 55

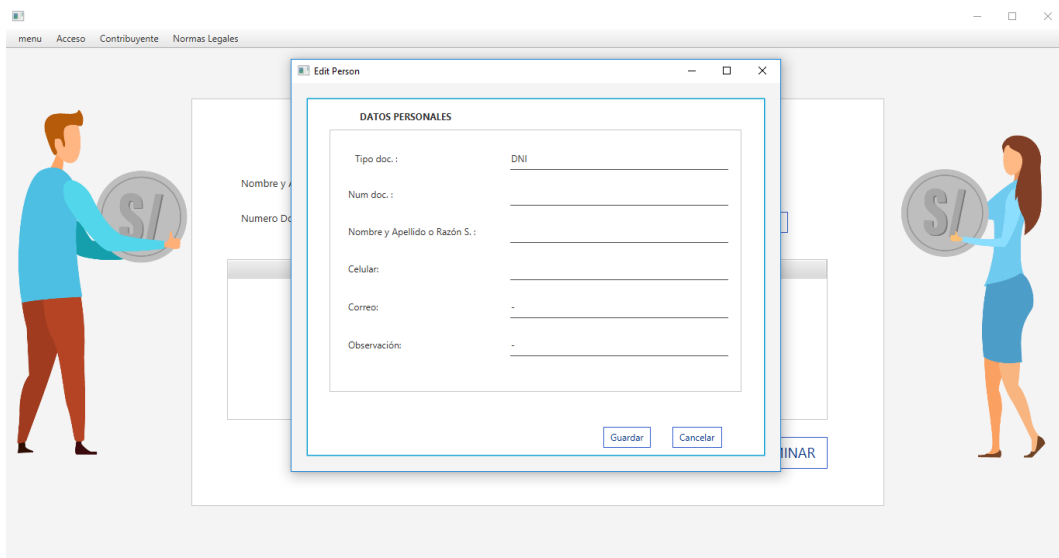
Desarrollo de contribuyente urbana vista n° 1



Nota. Fuente elaboración propia, desarrollado con java.

Figura 56

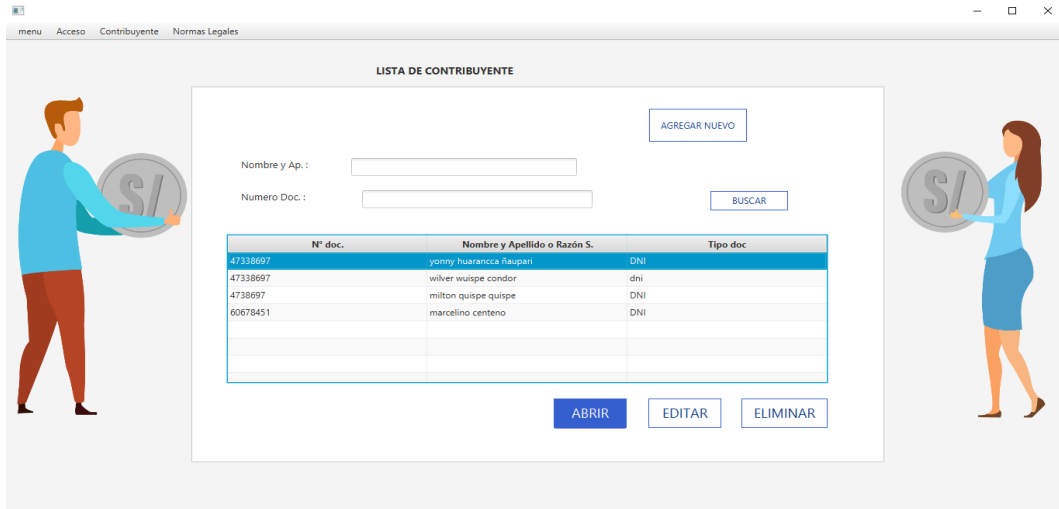
Desarrollo de contribuyente urbana vista n° 2



Nota. Fuente elaboración propia, desarrollado con java.

Figura 57

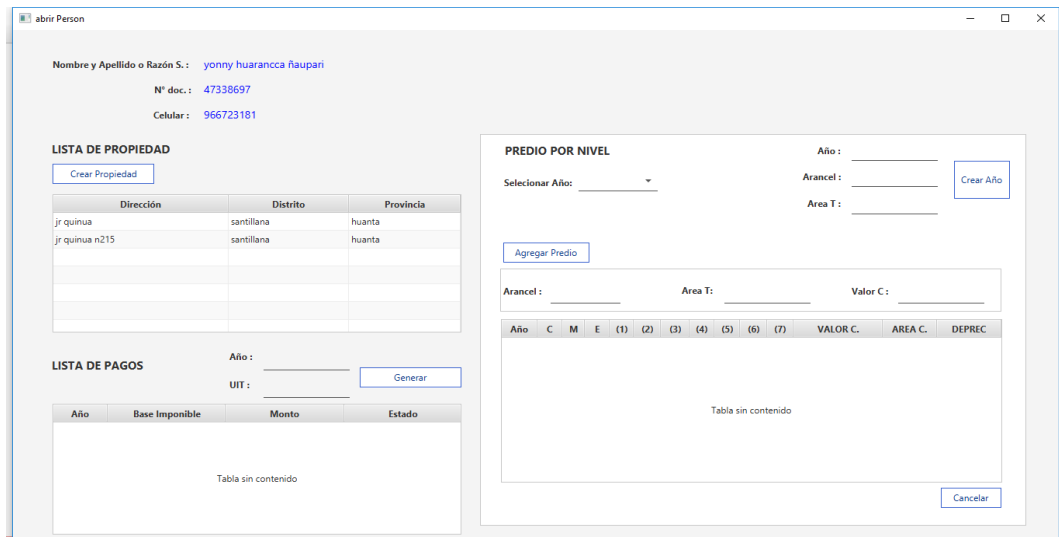
Desarrollo de contribuyente urbana vista n° 3



Nota. Fuente elaboración propia, desarrollado con java.

Figura 58

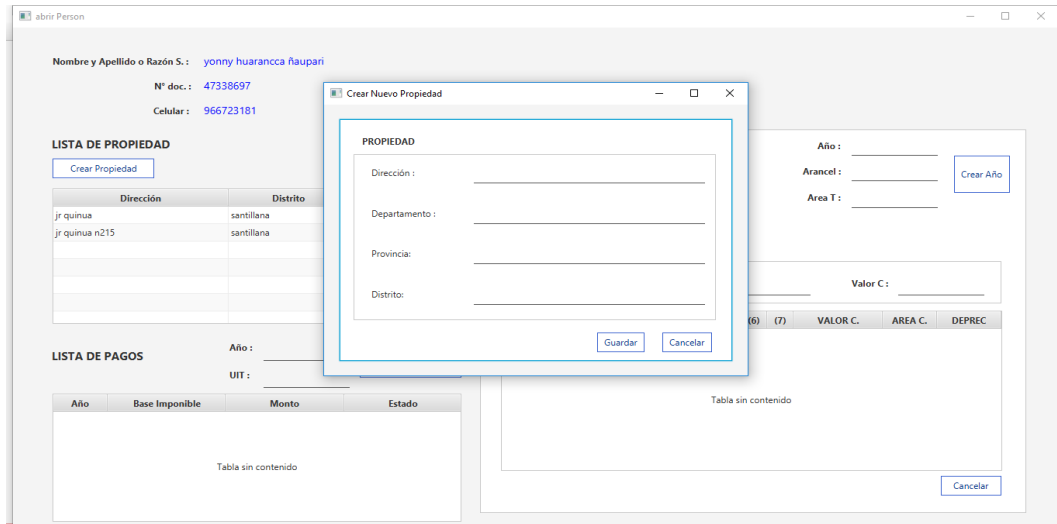
Desarrollo de contribuyente urbana vista n° 4



Nota. Fuente elaboración propia, desarrollado con java.

Figura 59

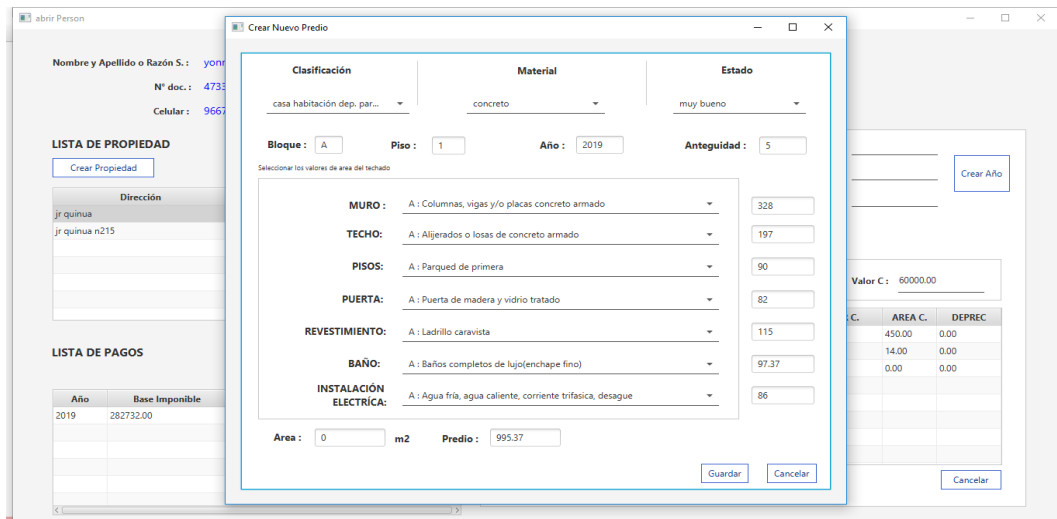
Desarrollo de contribuyente urbana vista n° 5



Nota. Fuente elaboración propia, desarrollado con java.

Figura 60

Desarrollo de contribuyente urbana vista n° 6



Nota. Fuente elaboración propia, desarrollado con java.

Las entregas o visualizaciones a encargado de la Municipalidad acorde a la meta establecida son Figura 55, Figura 56, Figura 57, Figura 58, Figura 59 y Figura 60 así cumpliendo en la fecha dada de dos semanas y 5 días (02/09/2019 - 21/09/2019).

- **PARA SPRINT 3**

En el tercer proceso de sprint 3 nos muestra un reporte de todas las incidencias realizadas en el proceso, para lo cual se entrega o se muestra al cliente (tesorera de la municipalidad) con las metas que se acordó en la reunión.

Meta del sprint 3:

1. Desarrollo del contribuyente rural o rustico

Comprende que cada trabajo se terminó o se devuelto al backlog en cada sprint, esto ayuda a determinar si el equipo está comprometido con el proyecto o hay demasiada corrupción del alcance.

En la Figura 61 se visualiza la satisfacción del trabajo antes de la fecha establecida 22/abr/20 2:20 am – 22/abr/20 2:30 am según la simulación dado, con esto podemos analizar que los trabajadores están bien involucrados, cumpliendo total 86 puntos de historia requerida.

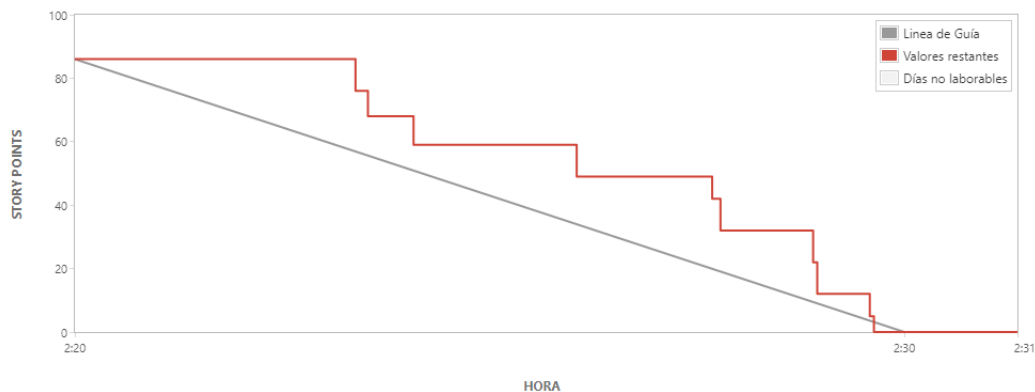
Figura 61

Trabajo culminado del tablero sprint 3

Reporte de Sprint

Tablero Sprint 3 ▾

Sprint cerrado, terminado por [YONNY HUARANCCA ÑAUPARI](#) 22/abr/20 2:20 AM - 22/abr/20 2:31 AM [View linked pages](#)
Desarrollo del contribuyente rural o rustico



Nota. Fuente elaboración propia, generado con Jira Software.

Tabla 53

Finalizada del sprint 3

Informe de estado

Incidencias terminadas

[Ver en el navegador de incidencias](#)

Clave	Resumen	Tipo de Incidencia	Prioridad	Estado	Story Points (86)
SIS-24	HU17 - AGREGAR UN NUEVO CONTRIBUYENTE RURAL O RUSTICO	Historia	Highest	FINALIZADA	10
SIS-25	HU18 - BUSCADOR DE CONTRIBUYENTE RURAL O RUSTICO	Historia	Medium	FINALIZADA	8
SIS-26	HU19 - EDITAR/ELIMINAR CONTRIBUYENTE DE ZONA RURAL O RUSTICO	Historia	Highest	FINALIZADA	9
SIS-27	HU20 - ABRIR UN CONTRIBUYENTE DE DE LA ZONA RURAL O RUSTICO	Historia	High	FINALIZADA	10
SIS-28	HU21 - VISUALIZAR LA VENTANA DE ABRIR PERSONA RURAL O RUSTICO	Historia	Low	FINALIZADA	7
SIS-29	HU22 - CREAR PROPIEDAD DE ZONA RURAL O RUSTICO	Historia	Highest	FINALIZADA	10
SIS-30	HU23 - CREAR AÑO PARA PREDIO RURAL O RUSTICO	Historia	Highest	FINALIZADA	10
SIS-31	HU24 - CREAR NUEVO PREDIO RURAL O RUSTICO	Historia	Highest	FINALIZADA	10
SIS-32	HU25 - LISTA DE PAGO DE ZONA RURAL O RUSTICO	Historia	Medium	FINALIZADA	7
SIS-33	HU26 - CERRAR/MINIMIZAR LA VISTA ABRIR PERSONA RURAL O RUSTICO	Historia	Low	FINALIZADA	5

Nota. Fuente elaboración propia, generado con Jira Software.

En el reporte de la Tabla 53 se visualiza todas las incidencias finalizados según la prioridad que se lo asigno, como también la asignación de puntos de historia de cada uno de las incidencias así acumulando un total de 86 puntos que se requería para la finalización.

La meta (desarrollo del contribuyente rural o rustico) se desarrolló correctamente como se muestra en la Figura 62, Figura 63, Figura 64, Figura 65, Figura 66, Figura 67, Figura 68, Figura 69, Figura 70 y Figura 71 para la entrega o visualización correspondiente al encargado de la municipalidad (cliente).

Figura 62

Desarrollo de contribuyente rural vista n° 1

ABRIR PERSONA - RUSTICO

Nombre y Apellido o Razón S.: POTOSINO QUISPE SEGUNDINA
 N° doc.: 28593910
 Uso del Predio:

LISTA DE PROPIEDAD

Crear Propiedad Eliminar Propiedad Editar Propiedad

Dirección	Lugar / zona / Balle	Carretera
SA: Santillana	balle	--

LISTA DE PAGOS

Año:
 UIT: Generar

Año	Base Imponible	Monto	Estado
Tabla sin contenido			

Valor Terreno:

Agregar Predio Eliminar Predio

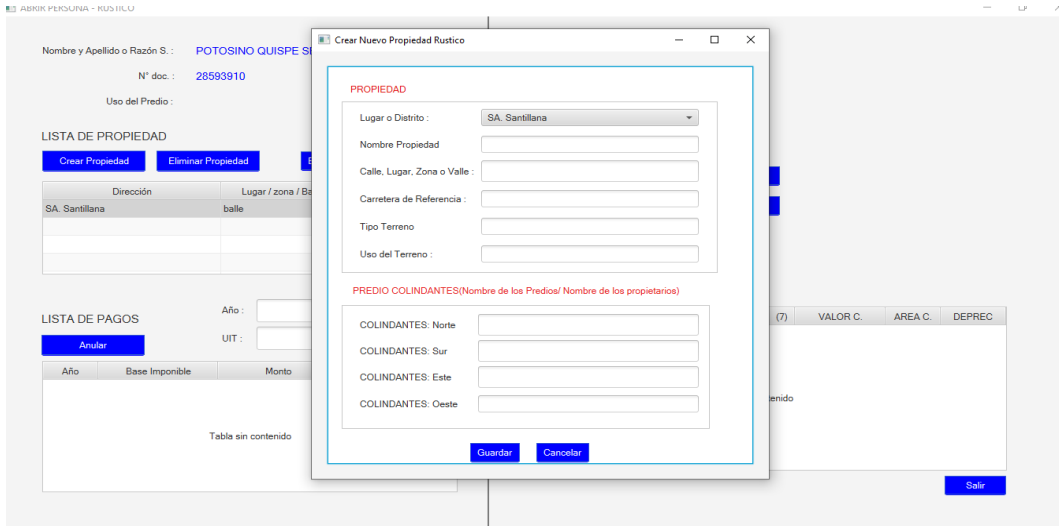
Año	C	M	E	(1)	(2)	(3)	(4)	(5)	(6)	(7)	VALOR C.	AREA C.	DEPREC
Tabla sin contenido													

Salir

Nota. Fuente elaboración propia, desarrollado con java.

Figura 63

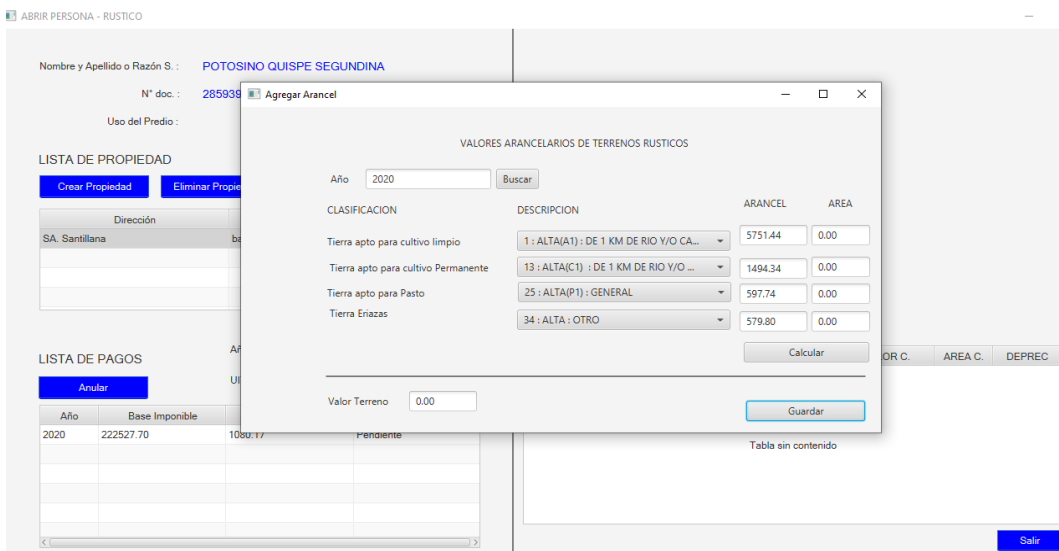
Desarrollo de contribuyente rural vista n° 2



Nota. Fuente elaboración propia, desarrollado con java.

Figura 64

Desarrollo de contribuyente rural vista n° 3



Nota. Fuente elaboración propia, desarrollado con java.

Figura 65

Desarrollo de contribuyente rural vista n° 4

Nota. Fuente elaboración propia, desarrollado con java.

Figura 66

Desarrollo de contribuyente rural vista n° 5

Nota. Fuente elaboración propia, desarrollado con java.

Figura 67

Desarrollo de contribuyente rural vista n° 6

Nombre y Apellido o Razón S. : **POTOSINO QUISPE SEGUNDINA**
 N° doc. : **28593910**
 Uso del Predio :

LISTA DE PROPIEDAD
 Crear Propiedad Eliminar Propiedad Editar Propiedad

Dirección	Lugar / zona / Balle	Carretera
SA. Santillana	balle	--

LISTA DE PAGOS
 Año : 2020 UIT : 4250 Generar
 Anular

Año	Base Imponible	Monto	Estado
2020	222527.70	1080.17	Pendiente

Año: 2020
 Valor Terreno : 3825.81
 Agregar Predio Eliminar Predio

Año	C	M	E	(1)	(2)	(3)	(4)	(5)	(6)	(7)	VALOR C.	AREA C.	DEPREC.
2020	1	1	1	B	C	E	A	A	A	A	736.37	297.00	0.00

Salir

Nota. Fuente elaboración propia, desarrollado con java.

Figura 68

Desarrollo de contribuyente rural vista n° 7

menu Acceso Contribuyente Consultas Arancel Rural Normas Legales

TIERRAS APTAS PARA CULTIVOS LIMPIOS

AÑO:
 DESCRIPCION:
 MONTO:
 CATEGORIA:

NUEVO Agregar Actualizar Eliminar

DESCRIPCION	MONTO	CALIDAD AGROLOGICA	AÑO
DE 1 KM DE RIO Y/O CARRETERAS	5751.44	ALTA(A1)	2020
DE 1 KM DE RIO Y/O CARRETERAS	5080.75	MEDIA(A2)	2020
DE 1 KM DE RIO Y/O CARRETERAS	3586.41	BAJA(A3)	2020
DE 2 KM DE RIO Y/O CARRETERAS	5678.48	ALTA(A1)	2020
DE 2 KM DE RIO Y/O CARRETERAS	4826.71	MEDIA(A2)	2020
DE 2 KM DE RIO Y/O CARRETERAS	3407.09	BAJA(A3)	2020
DE 3 KM DE RIO Y/O CARRETERAS	5379.62	ALTA(A1)	2020

Nota. Fuente elaboración propia, desarrollado con java.

Figura 69

Desarrollo de contribuyente rural vista n° 8

menu Acceso Contribuyente Consultas Arancel Rural Normas Legales

TIERRAS APTAS PARA CULTIVOS PERMANENTES

AÑO:

DESCRIPCION:

MONTO:

CATEGORIA:

DESCRIPCIÓN	MONTO	CALIDAD AGROLOGICA	AÑO
DE 1 KM DE RIO Y/O CARRETERAS	1494.34	ALTA(C1)	2020
DE 1 KM DE RIO Y/O CARRETERAS	1270.19	MEDIA(C2)	2020
DE 1 KM DE RIO Y/O CARRETERAS	896.60	BAJA(C3)	2020
DE 2 KM DE RIO Y/O CARRETERAS	1419.62	ALTA(C1)	2020
DE 2 KM DE RIO Y/O CARRETERAS	1206.68	MEDIA(C2)	2020
DE 2 KM DE RIO Y/O CARRETERAS	851.77	BAJA(C3)	2020
DE 3 KM DE RIO Y/O CARRETERAS	1344.90	ALTA(C1)	2020

Nota. Fuente elaboración propia, desarrollado con java.

Figura 70

Desarrollo de contribuyente rural vista n° 9

menu Acceso Contribuyente Consultas Arancel Rural Normas Legales

TIERRAS APTAS PARA PASTOS

AÑO:

DESCRIPCION:

MONTO:

CATEGORIA:

DESCRIPCIÓN	MONTO	CALIDAD AGROLOGICA	AÑO
GENERAL	597.74	ALTA(P1)	2020
GENERAL	508.07	MEDIA(P2)	2020
GENERAL	358.64	BAJA(P3)	2020

Nota. Fuente elaboración propia, desarrollado con java.

Figura 71

Desarrollo de contribuyente rural vista n° 10

TIERRAS APTAS PARA ERIAZAS

AÑO

DESCRIPCION

MONTO

CATEGORIA

NUEVO **Agregar** **Actualizar** **Eliminar**

DESCRIPCIÓN	MONTO	CALIDAD AGROLOGICA	AÑO
OTRO	579.80	ALTA	2020

Nota. Fuente elaboración propia, desarrollado con java.

Las entregas o visualizaciones a encargado de la Municipalidad (cliente) acorde a la meta establecida son Figura 62, Figura 63, Figura 64, Figura 65, Figura 66, Figura 67, Figura 68, Figura 69, Figura 70 y Figura 71 así cumpliendo en la fecha dada de dos semanas y 3 días (23/sep/19 - 9/oct/19).

- **PARA SPRINT 4**

En el cuarto proceso de sprint 3 nos muestra un reporte de todas las incidencias realizadas en el proceso, para lo cual se entrega o se muestra al cliente (tesorera de la municipalidad) con las metas que se acordó en la reunión.

Meta del sprint 3:

1. Desarrollo de sistema de caja.
2. Desarrollo de sistema reportes,
3. Desarrollo de sistema de la parte norma legal.
4. Desarrollo de una parte del sistema de registro de usuarios.

Comprende que cada trabajo se terminó o se devuelto al backlog en cada sprint, esto ayuda a determinar si el equipo está comprometido con el proyecto o hay demasiada corrupción del alcance.

En la Figura 72 se visualiza la satisfacción del trabajo antes de la fecha establecida 22/abr/20 4:00 am – 22/abr/20 4:10 am según la simulación dado, con

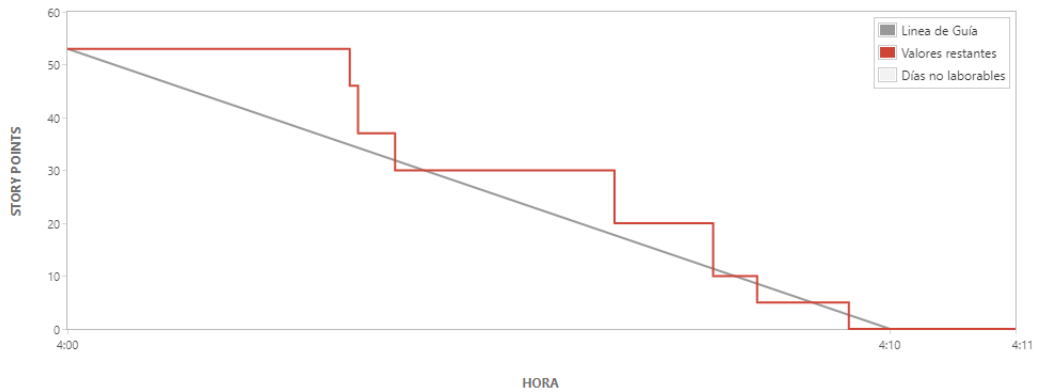
esto podemos analizar que los trabajadores están bien involucrados, cumpliendo total 53 puntos de historia requerida.

Figura 72

Trabajo culminado del tablero sprint 4

Reporte de Sprint

Sprint cerrado, terminado por YONNY HUARANCCA ÑAUPARI 22/abr/20 4:00 AM - 22/abr/20 4:11 AM [View linked pages](#)
 desarrollo de caja, reportes, normas legales y registrador de usuario



Nota. Fuente elaboración propia, generado con Jira Software.

Tabla 54

Finalizada del sprint 4

Informe de estado

Incidencias terminadas [Ver en el navegador de incidencias](#)

Clave	Resumen	Tipo de Incidencia	Prioridad	Estado	Story Points (53)
SIS-13	HU06 - VISUALIZACIÓN DE USUARIO	Historia	Lowest	FINALIZADA	7
SIS-34	HU27 - BUSCAR LISTA DE CONTRIBUYENTE	Historia	High	FINALIZADA	9
SIS-35	HU28 - VISUALIZACIÓN DEL LISTADO DE PROPIEDAD	Historia	Medium	FINALIZADA	7
SIS-36	HU29 - GENERAR PAGO DEL IMPUESTO PREDIAL	Historia	Highest	FINALIZADA	10
SIS-37	HU30 - VISUALIZACIÓN DE REPORTES URBANO Y RURAL	Historia	Highest	FINALIZADA	10
SIS-38	HU31 - VISUALIZACIÓN DE DOCUMENTOS	Historia	High	FINALIZADA	5
SIS-39	HU32 - VISUALIZACIÓN DE AYUDA	Historia	High	FINALIZADA	5

Nota. Fuente elaboración propia, generado con Jira Software.

En el reporte de la Tabla 54 se visualiza todas las incidencias finalizados según la prioridad que se lo asigno, como también la asignación de puntos de historia de cada uno de las incidencias así acumulando un total de 53 puntos que se requería para la finalización.

La meta se desarrolló correctamente como se muestra en la Figura 73, Figura 74, Figura 75, Figura 76 y Figura 77 para la entrega o visualización correspondiente al encargado de la municipalidad (cliente).

Figura 73

Desarrollo de caja vista n° 1

The screenshot displays two side-by-side web panels. The left panel, titled 'LISTA DE CONTRIBUYENTES URBANO', features search fields for 'Nombre' and 'Documento', a 'Buscar' button, and a dropdown menu set to '2: Predio Rural'. Below is a table of contributors with columns for 'Contribuyente' and 'documento de Identidad'. The right panel, titled 'LISTA DE PROPIEDADES URBANA', has fields for 'Direccion' and 'Distrito', a 'Tabla sin contenido' message, and a section for 'LISTA DE PROPIEDADES RURAL' with a 'Ubicacion del Terreno' dropdown. At the bottom right is a 'PAGAR' button.

Contribuyente	documento de Identidad
POTOSINO QUISPE SEGUNDA	28593910
LAPA CCAHUANA ALEJANDRO	28576002
FIGUEROA PARIONA FAUSTINO	28591671
MENDEZ YUPANQUI EDMUNDO	28591780
GAVILAN FIGUEROA JULITA	28592596
VENANCIO GONZALES EZPINOZA LIBIA GARA...	08568295
MENDEZ YUPANQUE TOLENTINO	28591779
ABSALON QUISPE MERINO -SONIA HUAMANI ...	28289689
OZEJO DE RIVAS EFRAINA	28292263
AVILA RODRIGUEZ CLAUDIO	28592403
QUINTANA PADILLA ALEXANDER ATILIO	72500336
QUINTANA PADILLA TANIA GARDENIA	70424009
CALLE RAMOS CLEOFE	70216891

Nota. Fuente elaboración propia, desarrollado con java.

Figura 74

Reporte de boleta

The screenshot shows a 'BOLETA' report from the 'MUNICIPALIDAD DISTRITAL DE SANTILLANA'. It includes fields for 'Apellidos y Nombres o Razón Social' (yonny huaranca flaupari), 'N° DNI o RUC' (47338697), and 'Fecha' (09/06/2019). The report details the 'Derecho de Emisión' (0), 'Base Imponible de PU' (282732.00), 'Impuesto Anual' (2823.13), and 'Impuesto Trimestral' (705.79). A table shows the 'LIQUIDACIÓN DE PAGOS' for four trimesters, all with a status of 'Pagar'. The total amount to be paid is 2823.13 soles.

LIQUIDACIÓN DE PAGOS:	Impuesto	Mora / Interes	Sub Total	Estado Pago	Vencimiento
1er Trimestre	705.79	0	705.79	<input checked="" type="checkbox"/> Pagar	A
2do Trimestre	705.79	0	705.79	<input checked="" type="checkbox"/> Pagar	A
3er Trimestre	705.79	0	705.79	<input checked="" type="checkbox"/> Pagar	A
4to Trimestre	705.79	0	705.79	<input checked="" type="checkbox"/> Pagar	A

Nota. Fuente elaboración propia, desarrollado con java.

Figura 75

Reporte de autoavaluo

predio.pdf - Adobe Reader

Archivo Edición Ver Ventana Ayuda

Herramientas Rellenar y firmar Comentario

DECLARACIÓN JURADA DE AUTOVALUO
T.E.O. de la Ley de Tributación Municipal Aprobado por el Decreto Supremo N° 154-2018-EF

PU AÑO FISCAL 2019
(PREDIO URBANO) N° DD. JJ. -

IDENTIFICACIÓN DEL CONTRIBUYENTE
 NÚMERO DE RUC: ARTILLADO Y VIVEROS S O NAZOR SOCIAL
47130287 sonny Huánuico Flaquez

UBICACIÓN DEL PREDIO
 LUGAR O DISTRITO: SANTISIMA DENOMINACIÓN: C.P. Centro Poblado
 DIRECCIÓN REG.: J. Quilva NÚMERO: - MZ.: - LOTE: -

DATOS RELATIVOS AL PREDIO
 CONDICIÓN DE PROPIEDAD: SOCIEDAD CONJUGAL
 ESTADO DE CONSTRUCCIÓN: TERMINADO TIPO DE PREDIO: PREDIO TOTAL USO DEL PREDIO: Otra habitación. Sencos. Ofic.

DATOS RELATIVOS AL PREDIO PARA EFECTOS DE LA DEPRECIACIÓN
 CLASIFICACIÓN DEL PREDIO: MATERIAL ESTRUCTURAL PRECORSAMENTE ESTADO DE CONSERVACIÓN: MUY BUENO
 LÍNEA INMOBILIARIA: Sencos. Instalaciones. Cables. etc.

DETERMINACIÓN DEL AUTOVALUO (Datos de la Construcción)

N°	UNIDAD	DESCRIPCIÓN DE DESCRIPCIÓN	Valor Unit.	Valor	Valor Unit.	Valor	Valor Unit.	Valor	Valor Unit.	Valor
			(por m ²)		(m ²)	(por m ²)		(m ²)	(por m ²)	
1	A	ESTRUC. (C/O)	494.50	0.00	0.00	494.50	222172.00	0.00	222172.00	
1	A	ESTRUC. (C/O)	861.37	0.00	0.00	861.37	11219.18	0.00	11219.18	
1	A	ESTRUC. (C/O)	995.01	0.00	0.00	995.01	0.00	0.00	0.00	

ÁREA DEL TERRENO: 1500.00 ARANCEL: 40.00 VALOR TOTAL DE LA CONSTRUCCIÓN: 233391.18

REGIMEN DE RAFAECIÓN / EDIFICACIÓN: REGIMEN DE RAFAECIÓN / EDIFICACIÓN VALOR ESTIMADO DE OTRAS INSTALACIONES: 0.00

REGIMEN: (N° EXPEDIV) REGIMEN (Punto de Venta) VALOR TOTAL DEL TERRENO: 60000.00

REGIMEN: Normal VALOR TOTAL DEL TERRENO: 293391.18

Nota. Fuente elaboración propia, desarrollado con java.

Figura 76

Desarrollo de normas legales n° 1

Normas Legales

Documentos

ayuda

NORMAS LEGALES

MINISTERIO DE VIVIENDA, CONSTRUCCIÓN Y SANEAMIENTO

RESOLUCIÓN MINISTERIAL N° 370-2018-VIVIENDA

REGLAMENTO DE LA LEY N° 29554

REGLAMENTO DE LA LEY N° 29554

Artículo	Contenido
1	OBJETIVO
2	ÁMBITO DE APLICACIÓN
3	DEFINICIONES
4	DECLARACIÓN JURADA DE AUTOVALUO
5	VALOR ESTIMADO DE OTRAS INSTALACIONES
6	VALOR TOTAL DEL TERRENO
7	VALOR TOTAL DEL PREDIO
8	VALOR ESTIMADO DE OTRAS INSTALACIONES
9	VALOR TOTAL DEL PREDIO
10	VALOR ESTIMADO DE OTRAS INSTALACIONES
11	VALOR TOTAL DEL PREDIO
12	VALOR ESTIMADO DE OTRAS INSTALACIONES
13	VALOR TOTAL DEL PREDIO
14	VALOR ESTIMADO DE OTRAS INSTALACIONES
15	VALOR TOTAL DEL PREDIO
16	VALOR ESTIMADO DE OTRAS INSTALACIONES
17	VALOR TOTAL DEL PREDIO
18	VALOR ESTIMADO DE OTRAS INSTALACIONES
19	VALOR TOTAL DEL PREDIO
20	VALOR ESTIMADO DE OTRAS INSTALACIONES
21	VALOR TOTAL DEL PREDIO
22	VALOR ESTIMADO DE OTRAS INSTALACIONES
23	VALOR TOTAL DEL PREDIO
24	VALOR ESTIMADO DE OTRAS INSTALACIONES
25	VALOR TOTAL DEL PREDIO
26	VALOR ESTIMADO DE OTRAS INSTALACIONES
27	VALOR TOTAL DEL PREDIO
28	VALOR ESTIMADO DE OTRAS INSTALACIONES
29	VALOR TOTAL DEL PREDIO
30	VALOR ESTIMADO DE OTRAS INSTALACIONES
31	VALOR TOTAL DEL PREDIO
32	VALOR ESTIMADO DE OTRAS INSTALACIONES
33	VALOR TOTAL DEL PREDIO
34	VALOR ESTIMADO DE OTRAS INSTALACIONES
35	VALOR TOTAL DEL PREDIO
36	VALOR ESTIMADO DE OTRAS INSTALACIONES
37	VALOR TOTAL DEL PREDIO
38	VALOR ESTIMADO DE OTRAS INSTALACIONES
39	VALOR TOTAL DEL PREDIO
40	VALOR ESTIMADO DE OTRAS INSTALACIONES
41	VALOR TOTAL DEL PREDIO
42	VALOR ESTIMADO DE OTRAS INSTALACIONES
43	VALOR TOTAL DEL PREDIO
44	VALOR ESTIMADO DE OTRAS INSTALACIONES
45	VALOR TOTAL DEL PREDIO
46	VALOR ESTIMADO DE OTRAS INSTALACIONES
47	VALOR TOTAL DEL PREDIO
48	VALOR ESTIMADO DE OTRAS INSTALACIONES
49	VALOR TOTAL DEL PREDIO
50	VALOR ESTIMADO DE OTRAS INSTALACIONES
51	VALOR TOTAL DEL PREDIO
52	VALOR ESTIMADO DE OTRAS INSTALACIONES
53	VALOR TOTAL DEL PREDIO
54	VALOR ESTIMADO DE OTRAS INSTALACIONES
55	VALOR TOTAL DEL PREDIO
56	VALOR ESTIMADO DE OTRAS INSTALACIONES
57	VALOR TOTAL DEL PREDIO
58	VALOR ESTIMADO DE OTRAS INSTALACIONES
59	VALOR TOTAL DEL PREDIO
60	VALOR ESTIMADO DE OTRAS INSTALACIONES
61	VALOR TOTAL DEL PREDIO
62	VALOR ESTIMADO DE OTRAS INSTALACIONES
63	VALOR TOTAL DEL PREDIO
64	VALOR ESTIMADO DE OTRAS INSTALACIONES
65	VALOR TOTAL DEL PREDIO
66	VALOR ESTIMADO DE OTRAS INSTALACIONES
67	VALOR TOTAL DEL PREDIO
68	VALOR ESTIMADO DE OTRAS INSTALACIONES
69	VALOR TOTAL DEL PREDIO
70	VALOR ESTIMADO DE OTRAS INSTALACIONES
71	VALOR TOTAL DEL PREDIO
72	VALOR ESTIMADO DE OTRAS INSTALACIONES
73	VALOR TOTAL DEL PREDIO
74	VALOR ESTIMADO DE OTRAS INSTALACIONES
75	VALOR TOTAL DEL PREDIO
76	VALOR ESTIMADO DE OTRAS INSTALACIONES
77	VALOR TOTAL DEL PREDIO
78	VALOR ESTIMADO DE OTRAS INSTALACIONES
79	VALOR TOTAL DEL PREDIO
80	VALOR ESTIMADO DE OTRAS INSTALACIONES
81	VALOR TOTAL DEL PREDIO
82	VALOR ESTIMADO DE OTRAS INSTALACIONES
83	VALOR TOTAL DEL PREDIO
84	VALOR ESTIMADO DE OTRAS INSTALACIONES
85	VALOR TOTAL DEL PREDIO
86	VALOR ESTIMADO DE OTRAS INSTALACIONES
87	VALOR TOTAL DEL PREDIO
88	VALOR ESTIMADO DE OTRAS INSTALACIONES
89	VALOR TOTAL DEL PREDIO
90	VALOR ESTIMADO DE OTRAS INSTALACIONES
91	VALOR TOTAL DEL PREDIO
92	VALOR ESTIMADO DE OTRAS INSTALACIONES
93	VALOR TOTAL DEL PREDIO
94	VALOR ESTIMADO DE OTRAS INSTALACIONES
95	VALOR TOTAL DEL PREDIO
96	VALOR ESTIMADO DE OTRAS INSTALACIONES
97	VALOR TOTAL DEL PREDIO
98	VALOR ESTIMADO DE OTRAS INSTALACIONES
99	VALOR TOTAL DEL PREDIO
100	VALOR ESTIMADO DE OTRAS INSTALACIONES

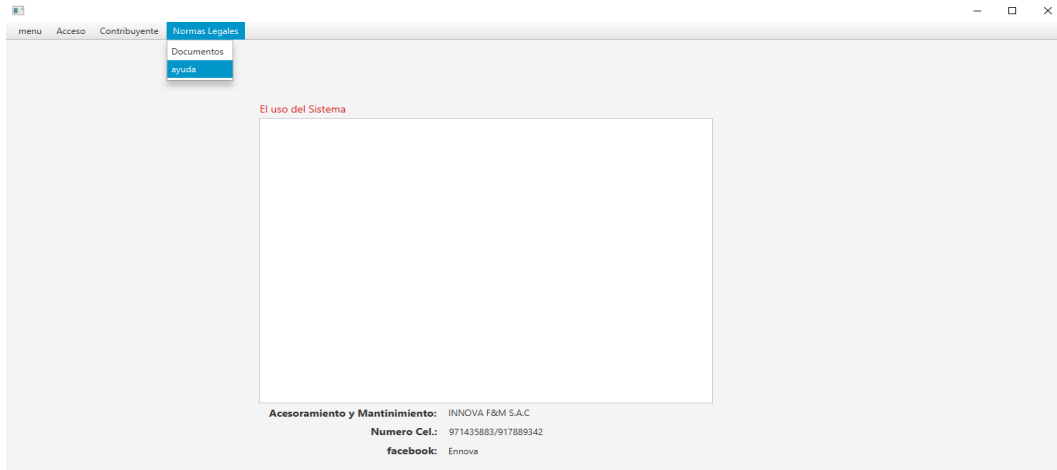
DOCUMENTOS

- * Resolución Ministerial N° 370-2018-VIVIENDA | Gobierno del Perú
<https://www.gob.pe/institucion/vivienda/normas-legales/212965-370-2018-vivienda>
- * Tabla Depreciación-2019 MINISTERIO DE VIVIENDA, CONSTRUCCIÓN Y SANEAMIENTO
https://www.municipal.gob.pe/meta31/TABLA_DEPRECIACION.pdf

Nota. Fuente elaboración propia, desarrollado con java.

Figura 77

Desarrollo de normas legales n° 2



Nota. Fuente elaboración propia, desarrollado con java.

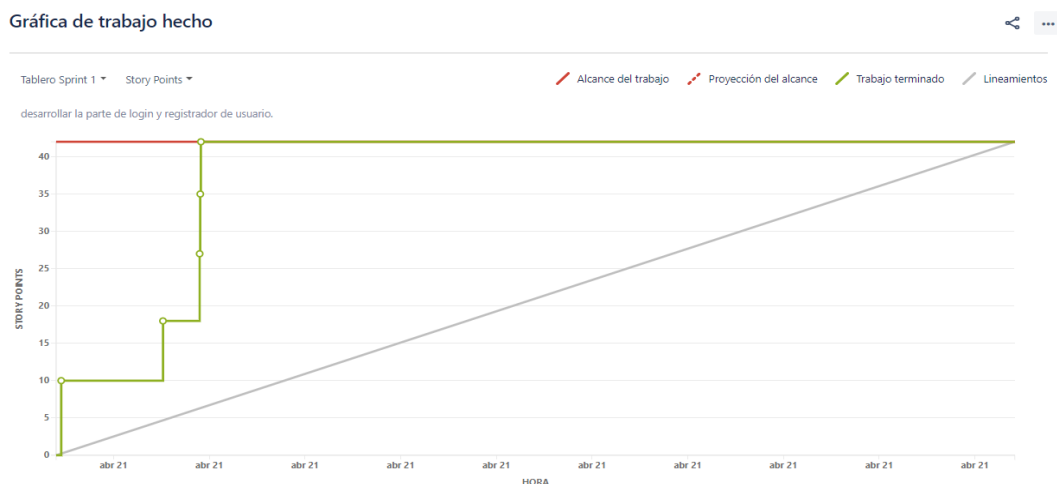
Las entregas o visualizaciones a encargado de la Municipalidad (cliente) acorde a la meta establecida son Figura 73, Figura 74, Figura 75, Figura 76 y Figura 77 así cumpliendo en la fecha dada de dos semanas y 3 días (23/sep/19 - 9/oct/19).

7.5. FASE N° 5: RETROSPECTIVA DE SPRINT (SPRINT RETROSPECTIVE)

En esta fase el equipo de desarrollo, Scrum Master y el propietario de producto se unen para discutir que funciona y que no funciona con Scrum. Por lo que se realiza retrospectiva de cada uno de los cuatro sprint. Donde en la presente investigación sprint 1, sprint 2, sprint 3 y sprint 4 se muestra estado final en lo siguiente:

Figura 78

Trabajo hecho para sprint 1



Nota. Fuente elaboración propia, generado con Jira Software.

Tabla 55

Estado final de las Tareas del Sprint 1

Fecha	Tipo de Evento	Incidencia	Trabajo terminado	Alcance del trabajo
21/04/20 7:02 AM	Sprint iniciado	SIS-8 HU01 - LOGUEARSE SIS-9 HU02 - SALIR DE LOGIN SIS-10 HU03 - REGISTRAR USUARIO SIS-11 HU04 - MODIFICAR USUARIO SIS-12 HU05 - ELIMINAR USUARIO	0	42
21/04/20 7:02 AM	Incidencia terminada	SIS-8 HU01 - LOGUEARSE	0 → 10	42
21/04/20 7:07 AM	Incidencia terminada	SIS-9 HU02 - SALIR DE LOGIN	10 → 18	42
21/04/20 7:09 AM	Incidencia terminada	SIS-10 HU03 - REGISTRAR USUARIO	18 → 27	42
21/04/20 7:09 AM	Incidencia terminada	SIS-11 HU04 - MODIFICAR USUARIO	27 → 35	42
21/04/20 7:09 AM	Incidencia terminada	SIS-12 HU05 - ELIMINAR USUARIO	35 → 42	42
21/04/20 7:52 AM	Sprint finalizado	SIS-8 HU01 - LOGUEARSE SIS-9 HU02 - SALIR DE LOGIN SIS-10 HU03 - REGISTRAR USUARIO SIS-11 HU04 - MODIFICAR USUARIO SIS-12 HU05 - ELIMINAR USUARIO	42	42

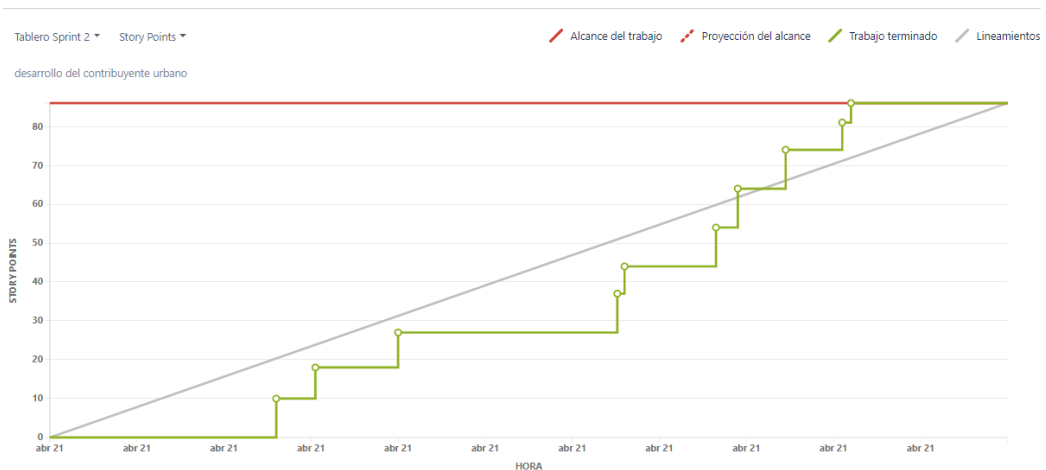
Nota. Fuente elaboración propia, generado con Jira Software.

En la Tabla 55 se muestra la aplicación correcta de Scrum para el desarrollo de parte de login del sistema y registrador de usuarios, permitiendo culminar cada historia, así alcanzando 42 puntos según establecido en el trabajo como se visualiza en la Figura 78 demostrando que no se encuentra ninguna observación en este sprint 1.

Figura 79

Trabajo hecho para sprint 2

Gráfica de trabajo hecho



Nota. Fuente elaboración propia, generado con Jira Software.

Tabla 56

Estado final de las Tareas del Sprint 2

Fecha	Tipo de Evento	Incidencia	Trabajo terminado	Alcance del trabajo
21/04/20 9:35 PM	Sprint iniciado	SIS-14 HU07 - AGREGAR NUEVO CONTRIBUYENTE URBANO SIS-15 HU08 - BUSCADOR DE CONTRIBUYENTE URBANO SIS-16 HU09 - EDITAR/ELIMINAR CONTRIBUYENTE URBANO SIS-17 HU10 - ABRIR UN CONTRIBUYENTE URBANO SIS-18 HU11 - VISUALIZAR ABRIR PERSONA URBANA SIS-19 HU12 - CREAR PROPIEDAD DE ZONA URBANA SIS-20 HU13 - CREAR AÑO PARA PREDIO URBANO SIS-21 HU14 - CREAR NUEVO PREDIO URBANO SIS-22 HU15 - GENERAR LISTA DE PAGO URBANO SIS-23 HU16 - CERRAR/MINIMIZAR LA VISTA ABRIR PERSONA URBANA	0	86
21/04/20 9:37 PM	Incidencia terminada	SIS-14 HU07 - AGREGAR NUEVO CONTRIBUYENTE URBANO	0 - 10	86
21/04/20 9:38 PM	Incidencia terminada	SIS-15 HU08 - BUSCADOR DE CONTRIBUYENTE URBANO	10 - 18	86
21/04/20 9:39 PM	Incidencia terminada	SIS-16 HU09 - EDITAR/ELIMINAR CONTRIBUYENTE URBANO	18 - 27	86
21/04/20 9:41 PM	Incidencia terminada	SIS-17 HU10 - ABRIR UN CONTRIBUYENTE URBANO	27 - 37	86
21/04/20 9:41 PM	Incidencia terminada	SIS-18 HU11 - VISUALIZAR ABRIR PERSONA URBANA	37 - 44	86
21/04/20 9:42 PM	Incidencia terminada	SIS-21 HU14 - CREAR NUEVO PREDIO URBANO	44 - 54	86
21/04/20 9:42 PM	Incidencia terminada	SIS-19 HU12 - CREAR PROPIEDAD DE ZONA URBANA	54 - 64	86
21/04/20 9:43 PM	Incidencia terminada	SIS-20 HU13 - CREAR AÑO PARA PREDIO URBANO	64 - 74	86
21/04/20 9:44 PM	Incidencia terminada	SIS-22 HU15 - GENERAR LISTA DE PAGO URBANO	74 - 81	86
21/04/20 9:44 PM	Incidencia terminada	SIS-23 HU16 - CERRAR/MINIMIZAR LA VISTA ABRIR PERSONA URBANA	81 - 86	86
21/04/20 9:45 PM	Sprint finalizado	SIS-14 HU07 - AGREGAR NUEVO CONTRIBUYENTE URBANO SIS-15 HU08 - BUSCADOR DE CONTRIBUYENTE URBANO SIS-16 HU09 - EDITAR/ELIMINAR CONTRIBUYENTE URBANO SIS-17 HU10 - ABRIR UN CONTRIBUYENTE URBANO SIS-18 HU11 - VISUALIZAR ABRIR PERSONA URBANA SIS-19 HU12 - CREAR PROPIEDAD DE ZONA URBANA SIS-20 HU13 - CREAR AÑO PARA PREDIO URBANO SIS-21 HU14 - CREAR NUEVO PREDIO URBANO SIS-22 HU15 - GENERAR LISTA DE PAGO URBANO SIS-23 HU16 - CERRAR/MINIMIZAR LA VISTA ABRIR PERSONA URBANA	86	86

Nota. Fuente elaboración propia, generado con Jira Software.

En el reporte final del trabajo echo de la Figura 79 y la Tabla 56 se muestra la aplicación correcta de Scrum del sprint 2, permitiendo culminar cada historia, con un puntaje total 86, así desarrollando satisfactoriamente la parte de contribuyente urbano en la fecha establecido. demostrando que no se encuentra ninguna observación en este sprint 2.

Figura 80

Trabajo hecho para sprint 3



Nota. Fuente elaboración propia, generado con Jira Software.

Tabla 57

Estado final de las Tareas del Sprint 3

Fecha	Tipo de Evento	Incidencia	Trabajo terminado	Alcance del trabajo
22/04/20 2:20 AM	Sprint iniciado	SIS-24 HU17 - AGREGAR UN NUEVO CONTRIBUYENTE RURAL O RUSTICO SIS-25 HU18 - BUSCADOR DE CONTRIBUYENTE RURAL O RUSTICO SIS-26 HU19 - EDITAR/ELIMINAR CONTRIBUYENTE DE ZONA RURAL O RUSTICO SIS-27 HU20 - ABRIR UN CONTRIBUYENTE DE DE LA ZONA RURAL O RUSTICO SIS-28 HU21 - VISUALIZAR LA VENTANA DE ABRIR PERSONA RURAL O RUSTICO SIS-29 HU22 - CREAR PROPIEDAD DE ZONA RURAL O RUSTICO SIS-30 HU23 - CREAR AÑO PARA PREDIO RURAL O RUSTICO SIS-31 HU24 - CREAR NUEVO PREDIO RURAL O RUSTICO SIS-32 HU25 - LISTA DE PAGO DE ZONA RURAL O RUSTICO SIS-33 HU26 - CERRAR/MINIMIZAR LA VISTA ABRIR PERSONA RURAL O RUSTICO	0	86
22/04/20 2:22 AM	Incidencia terminada	SIS-24 HU17 - AGREGAR UN NUEVO CONTRIBUYENTE RURAL O RUSTICO	0 - 10	86
22/04/20 2:23 AM	Incidencia terminada	SIS-25 HU18 - BUSCADOR DE CONTRIBUYENTE RURAL O RUSTICO	10 - 18	86
22/04/20 2:24 AM	Incidencia terminada	SIS-26 HU19 - EDITAR/ELIMINAR CONTRIBUYENTE DE ZONA RURAL O RUSTICO	18 - 27	86
22/04/20 2:26 AM	Incidencia terminada	SIS-27 HU20 - ABRIR UN CONTRIBUYENTE DE DE LA ZONA RURAL O RUSTICO	27 - 37	86
22/04/20 2:27 AM	Incidencia terminada	SIS-28 HU21 - VISUALIZAR LA VENTANA DE ABRIR PERSONA RURAL O RUSTICO	37 - 44	86
22/04/20 2:27 AM	Incidencia terminada	SIS-29 HU22 - CREAR PROPIEDAD DE ZONA RURAL O RUSTICO	44 - 54	86
22/04/20 2:28 AM	Incidencia terminada	SIS-30 HU23 - CREAR AÑO PARA PREDIO RURAL O RUSTICO	54 - 64	86
22/04/20 2:28 AM	Incidencia terminada	SIS-31 HU24 - CREAR NUEVO PREDIO RURAL O RUSTICO	64 - 74	86
22/04/20 2:29 AM	Incidencia terminada	SIS-32 HU25 - LISTA DE PAGO DE ZONA RURAL O RUSTICO	74 - 81	86
22/04/20 2:29 AM	Incidencia terminada	SIS-33 HU26 - CERRAR/MINIMIZAR LA VISTA ABRIR PERSONA RURAL O RUSTICO	81 - 86	86
22/04/20 2:31 AM	Sprint finalizado	SIS-24 HU17 - AGREGAR UN NUEVO CONTRIBUYENTE RURAL O RUSTICO SIS-25 HU18 - BUSCADOR DE CONTRIBUYENTE RURAL O RUSTICO SIS-26 HU19 - EDITAR/ELIMINAR CONTRIBUYENTE DE ZONA RURAL O RUSTICO SIS-27 HU20 - ABRIR UN CONTRIBUYENTE DE DE LA ZONA RURAL O RUSTICO SIS-28 HU21 - VISUALIZAR LA VENTANA DE ABRIR PERSONA RURAL O RUSTICO SIS-29 HU22 - CREAR PROPIEDAD DE ZONA RURAL O RUSTICO SIS-30 HU23 - CREAR AÑO PARA PREDIO RURAL O RUSTICO SIS-31 HU24 - CREAR NUEVO PREDIO RURAL O RUSTICO SIS-32 HU25 - LISTA DE PAGO DE ZONA RURAL O RUSTICO SIS-33 HU26 - CERRAR/MINIMIZAR LA VISTA ABRIR PERSONA RURAL O RUSTICO	86	86

Nota. Fuente elaboración propia, generado con Jira Software.

En el reporte final del trabajo echo de la Figura 80 y la Tabla 57 se muestra la aplicación correcta de Scrum del sprint 3 para el desarrollo del contribuyente rural o rustico, permitiendo culminar cada historia, con un puntaje total 86, así desarrollando satisfactoriamente en la fecha establecido del abril de 22. Demostrando que no se encuentra ninguna observación en este sprint 3, corrigiendo los problemas en su ocasión del Scrum diario.

Figura 81

Trabajo hecho para sprint 4



Nota. Fuente elaboración propia, generado con Jira Software.

Tabla 58

Estado final de las Tareas del Sprint 4

Fecha	Tipo de Evento	Incidencia	Trabajo terminado	Alcance del trabajo
22/04/20 4:00 AM	Sprint iniciado	SIS-13 HU06 - VISUALIZACIÓN DE USUARIO SIS-34 HU27 - BUSCAR LISTA DE CONTRIBUYENTE SIS-35 HU28 - VISUALIZACIÓN DEL LISTADO DE PROPIEDAD SIS-36 HU29 - GENERAR PAGO DEL IMPUESTO PREDIAL SIS-37 HU30 - VISUALIZACIÓN DE REPORTES URBANO Y RURAL SIS-38 HU31 - VISUALIZACIÓN DE DOCUMENTOS SIS-39 HU32 - VISUALIZACIÓN DE AYUDA	0	53
22/04/20 4:03 AM	Incidencia terminada	SIS-13 HU06 - VISUALIZACIÓN DE USUARIO	0 – 7	53
22/04/20 4:03 AM	Incidencia terminada	SIS-34 HU27 - BUSCAR LISTA DE CONTRIBUYENTE	7 – 16	53
22/04/20 4:03 AM	Incidencia terminada	SIS-35 HU28 - VISUALIZACIÓN DEL LISTADO DE PROPIEDAD	16 – 23	53
22/04/20 4:06 AM	Incidencia terminada	SIS-36 HU29 - GENERAR PAGO DEL IMPUESTO PREDIAL	23 – 33	53
22/04/20 4:07 AM	Incidencia terminada	SIS-37 HU30 - VISUALIZACIÓN DE REPORTES URBANO Y RURAL	33 – 43	53
22/04/20 4:08 AM	Incidencia terminada	SIS-38 HU31 - VISUALIZACIÓN DE DOCUMENTOS	43 – 48	53
22/04/20 4:09 AM	Incidencia terminada	SIS-39 HU32 - VISUALIZACIÓN DE AYUDA	48 – 53	53
22/04/20 4:11 AM	Sprint finalizado	SIS-13 HU06 - VISUALIZACIÓN DE USUARIO SIS-34 HU27 - BUSCAR LISTA DE CONTRIBUYENTE SIS-35 HU28 - VISUALIZACIÓN DEL LISTADO DE PROPIEDAD SIS-36 HU29 - GENERAR PAGO DEL IMPUESTO PREDIAL SIS-37 HU30 - VISUALIZACIÓN DE REPORTES URBANO Y RURAL SIS-38 HU31 - VISUALIZACIÓN DE DOCUMENTOS SIS-39 HU32 - VISUALIZACIÓN DE AYUDA	53	53

Nota. Fuente elaboración propia, generado con Jira Software.

En el reporte final del trabajo echo de la Figura 81 y la Tabla 58 se muestra la aplicación correcta de Scrum del sprint 4 para el desarrollo de la caja, reporte y normas legales, permitiendo culminar cada historia, con un puntaje total 53, así desarrollando satisfactoriamente en la fecha establecido del abril de 22. Demostrando que no se encuentra ninguna observación en este sprint 4, corrigiendo los problemas en su ocasión del Scrum diario.

Una vez levantado todas y verificad que no se encontró ninguna observación en el proyecto se procede a realizar el cierre del proyecto Tabla 59 con el dueño del proyecto, esta tarea le corresponde generalmente a Scrum Mater (Yonny Huaranca Ñaupari) en coordinación con el dueño del producto (Municipalidad Distrital de Santillana.

7.6. RESULTADOS

En esta sección se especifica los resultados obtenidos al Aplicar el marco Jira Scrum Agile Software que nos permite el desarrollo del Sistema Información de Recaudación Tributaria del impuesto predial de la Municipalidad distrital de Santillana, provincia de Huanta, Perú, en el año 2019.

Tabla 59

Cierre de proyecto con Scrum.

Ítem	Sprint	Responsable	Tareas	Prioridad	Estado	Estado y revisión	Puntuación
1	sprint 1	Yonny H.	hu01 – loguearse	High	Finish	Hecho	10
		Yonny H.	hu02 - salir de login	Medium	Finish	Hecho	8
		Pabel B.	hu03 - registrar usuario	Highest	Finish	Hecho	9
		Pabel B.	hu04 - modificar usuario	High	Finish	Hecho	8
		Pabel B.	hu05 - eliminar usuario	Low	Finish	Hecho	7
2	sprint 2	Roberth A.	hu07 - agregar nuevo contribuyente urbano	Medium	Finish	Hecho	10
		Roberth A.	hu08 - buscador de contribuyente urbano	Medium	Finish	Hecho	8
		Roberth A.	hu09 - editar/eliminar contribuyente urbano	Highest	Finish	Hecho	9
		Roberth A.	hu10 - abrir un contribuyente urbano	Highest	Finish	Hecho	10
		Roberth A.	hu11 - visualizar abrir persona urbana	Highest	Finish	Hecho	7
		Manrique M.	hu12 - crear propiedad de zona urbana	Highest	Finish	Hecho	10
		Manrique M.	hu13 - crear año para predio urbano	Highest	Finish	Hecho	10
		Manrique M.	hu14 - crear nuevo predio urbano	Highest	Finish	Hecho	10
		Manrique M.	hu15 - generar lista de pago urbano	Highest	Finish	Hecho	7
Manrique M.	hu16-cerrar/minimizar la vista abrir persona urbana	Low	Finish	Hecho	5		
3	Sprint 3	Yonny H.	Hu17-agregar un nuevo contribuyente rural o rustico	Low	Finish	Hecho	10
		Yonny H.	Hu18-buscador de contribuyente rural o rustico	Medium	Finish	Hecho	8

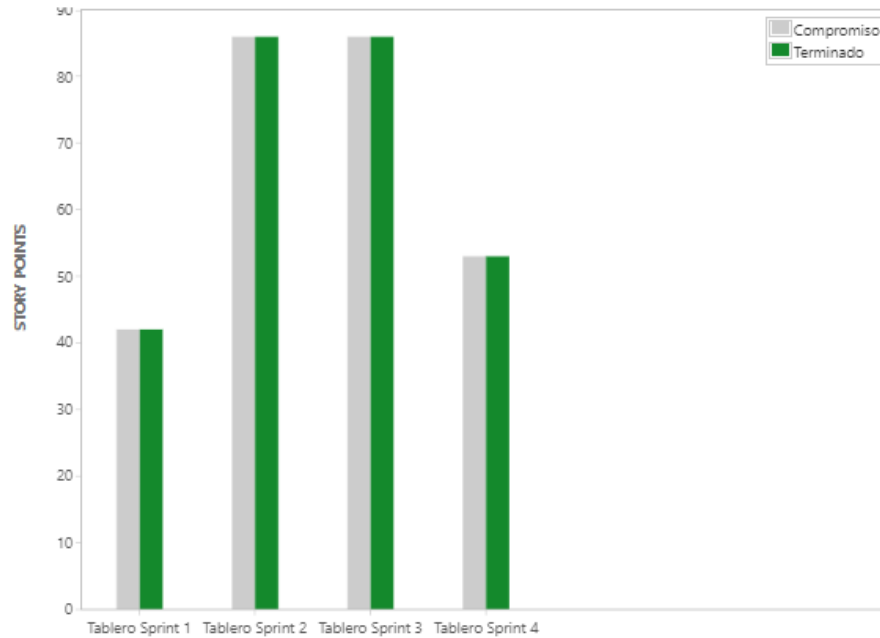
		Yonny H.	Hu19-editar/eliminar contribuyente rural o rustico.	Highest	Finish	Hecho	9
		Elvis T.	Hu20-abrir un contribuyente de la zona rural o rustico.	Highest	Finish	Hecho	10
		Elvis T.	Hu21-visualizar la ventana de abrir persona rural o rustico	Low	Finish	Hecho	7
		Manrique M.	Hu22-crear propiedad de zona rural o rustico	Highest	Finish	Hecho	10
		Manrique M.	Hu23-crear año para predio rural o rustico	Highest	Finish	Hecho	10
		Manrique M.	Hu24-crear nuevo predio rural o rustico	Highest	Finish	Hecho	10
		Manrique M.	Hu25-listar pago de zona rural o rustico	Medium	Finish	Hecho	10
		Manrique M.	Hu26-cerrar/minimizar la vista abrir persona rural o rustico	Low	Finish	Hecho	5
4	Sprin	Yonny H.	Hu06-visualizacion de usuario	Lowest	Finish	Hecho	7
	t 4	Pabel B.	Hu27-buscar lista de contribuyentes	High	Finish	Hecho	9
		Pabel B.	Hu28-Visualizacion del listado de propiedades	Medium	Fenish	Hecho	7
		Pabel B.	Hu29-generar pago del impuesto predial	Medium	Fenish	Hecho	10
		Roberth A.	Hu30-visualizacion de reportes urbanos y rural	Highest	Fenish	Hecho	10
		Elvis T.	Hu31-Visualización de doumentos	High	Fenish	Hecho	5
		Elvis T.	Hu32-visualización de ayuda	High	Fenish	Hecho	5

Nota. Fuente elaboración propia.

La Tabla 59 se muestra el estado actual de las tareas asignados a los responsables, lo cual tienen estado final de “hecho” como se muestra los HUs y su puntuación como: hu1(10), h2 (8), h3 (9), h4 (8), h5 (7), hu7 (10), hu8 (8), hu9 (9), hu10 (10), hu11 (7), hu12 (10), hu13 (10), hu14 (10), hu15 (7), hu16 (5), hu17(10), hu18(8), hu19(9), hu20(10), hu21(7), hu22(10), hu23(10), hu24(10), hu25(10), hu26(5), hu06(7), hu27(9), hu28(7), hu29(10), hu30(10), hu31(5) y hu32(5) obteniendo al cierre del proyecto, cometiendo con la finalización de desarrollar todas las tareas planteadas, cumplimiento con la meta y objetivos trazado la Municipalidad en licitar a la empresa Ennova m&f s.a.c. así validado y obteniéndose una versión terminada del producto (producto esperado).

Figura 82

Velocidad de punto de historia



Nota. Fuente elaboración propia, generado con Jira Software.

Tabla 60

Sprint de velocidad.

Sprint	Compromiso	Terminado
Tablero Sprint 1	42	42
Tablero Sprint 2	86	86
Tablero Sprint 3	86	86
Tablero Sprint 4	53	53

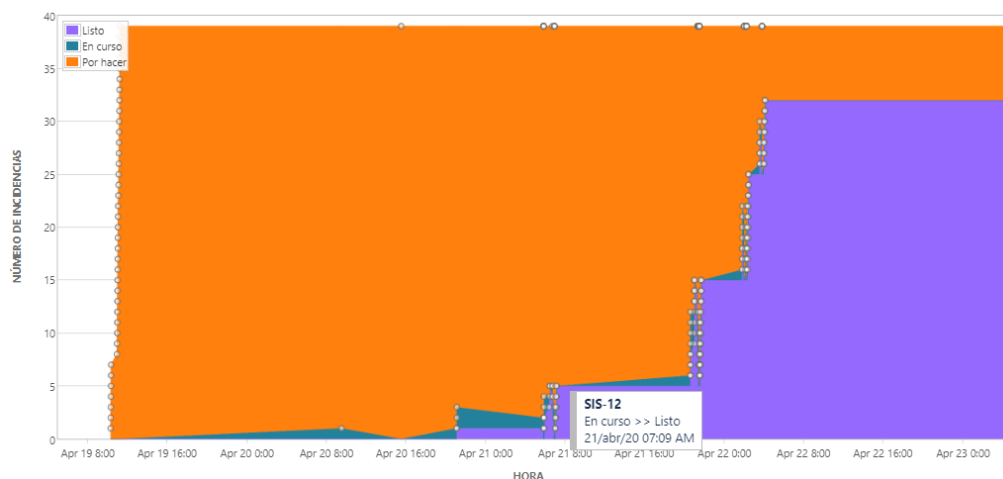
Nota. Fuente elaboración propia, generado con Jira Software.

En la Figura 82 y la Tabla 60 se puede visualizar la cantidad de trabajo realizado en cada sprint como sabemos la suma del sprint del tablero sprint 1 es 42, sprint del tablero 2 es 86, sprint 3 es de 86 y sprint 4 es de 53 teniendo como resultado ayudar a la determinación de la velocidad del equipo y la forma de la utilización del marco Jira Scrum Agile Software en el proyecto de Sistema de Información de Recaudación Tributaria del impuesto predial de la Municipalidad, llegamos a conocer la buena gestión de las expectativas del cliente, ya que el cliente

o Producto Owner (Marcelino Centeno representante de la Municipalidad) interactúa con Scrum Master (Yonny Huaranca) y el equipo de trabajo (Manrique Meneses, Pavel Bautista y Roberth Arete) en las 2 reuniones de la planificación de Sprint (Fase 2) para definir la fecha de inicio y fecha de final y las metas trazadas, también en la fase 4 revisión de sprint (Sprint Review) se reúnen el cliente y Scrum Master para las entregas de cada sprint culminado y en la fase 5 de Retrospectiva de Sprint también se reúnen para discutir que funciona y que no funciona con Scrum para el inicio de la nueva iteración o el cierre del proyecto. Terminando todo lo requerido del cliente (Municipalidad) como se muestra en la Figura 82 con puntuaciones completados de 42 y 86.

Figura 83

Flujo acumulado



Nota. Fuente elaboración propia, generado con Jira Software.

La utilización de Jira Scrum Agile Software para la gestión de los equipos sean auto dirigidos o auto organizados, pues mejora de gran manera principalmente en el compromiso de cada integrante del proyecto Scrum Master y equipo de trabajo (Scrum Team), como se muestra en la Figura 83. El sis-12 es el HU05 del sprint 1 que termino satisfactoriamente sin que haya ningún incidencias o bugs, los bugs puede influir por muchos factores (incorporación de tareas no planificadas, HUs no culminados, renuncia de personal, etc.), es un tema de investigación, pero en este estudio para el proyecto no se da por mismo hecho de que el equipo es auto dirigido y auto organizado, como consecuencia el equipo termino el primer sprint con total

de 42 puntos lo necesario para culminar. Por otra parte, que cada integrante de Scrum Team se auto organiza con la finalidad de tener una responsabilidad de tareas, que a la vez cada tarea tiene una cantidad de puntuación de historia, al finalizar cada tarea se suman un total de 42 y 86 puntos de historia que se proyectó en la Figura 82.

7.7. ANÁLISIS DE RESULTADOS

Basándose en la correcta ejecución del proyecto Sistema de Información de Recaudación Tributaria de la Municipalidad Distrital de Santillana (actualmente en uso), en todo el Capítulo VII de análisis y resultado, aplicando Jira Scrum Agile, se concluye que es un marco de trabajo de gran facilidad en la organización y gestión del proyecto.

CAPÍTULO VIII

CONCLUSIONES Y RECOMENDACIONES

8.1. CONCLUSIONES

En este trabajo se aplicó el marco Jira Scrum Agile Software que nos permite el desarrollo del Sistema Información de Recaudación Tributaria del impuesto predial de la Municipalidad distrital de Santillana, Huanta, Perú, 2019. Lo más importante de aplicar el marco de Jira Scrum Agile Software fue el cumplimiento de todos los parámetros y estándares para el desarrollo de Sistema Información de Recaudación Tributaria, porque facilito construir un sistema hibrido para la Municipalidad distrital de Santillana. Lo que más ayudo aplicar el marco Jira Scrum Agile Software fue la adaptabilidad al proyecto del sistema de recaudación tributaria del impuesto predial de la Municipalidad distrital de Santillana creando una cultura de colaboración, porque une a todo el equipo como al cliente en torno al objetivo y genera una entrega iterativa en cada finalización de sprint. Lo más difícil de aplicar el marco Jira Scrum Agil Software fue prevenir las incidencias o bugs, porque pudieron ocasionar si no hubiera habido una planificación detallado como en la reunión para realización de la pila producto con el producto Owner y en los sprint.

De acuerdo al primer objetivo específico se aplicó el marco Jira Scrum Agile Software para la realización de la gestión regular de las expectativas del cliente que nos permita el desarrollo del Sistema Información de Recaudación Tributaria del impuesto predial de la Municipalidad distrital de Santillana, Huanta, Perú, 2019. Lo más importante de aplicar el marco Jira Scrum Agile Software fue afirmar la gestión regular de las expectativas del cliente, porque es un beneficio para el proyecto en todas las reuniones de las iteraciones de sprint como en la reunión para la pila de producto, planificación de sprint, revisión de sprint y retrospectiva de sprint.

Para el segundo objetivo específico se aplicó el marco Jira Scrum Agile Software para la gestión de los equipos auto dirigidos y auto organizados que nos permitan el desarrollo del Sistema Información de Recaudación Tributaria del impuesto predial de la Municipalidad distrital de Santillana, Huanta, Perú, 2019. Lo más importante de aplicar el marco Jira Scrum Agile Software fue afirmar que la

gestión de los equipos es auto dirigidos y auto organizados, porque es un beneficio para Scrum Master, Scrum Team y Producto Owner, para la mejora del enfoque y la organización del proyecto, evitando los bugs que pudieron ocasionar en el proceso del proyecto del desarrollo del sistema.

En el desarrollo del Sistema de Información de Recaudación tributaria se puede afirmar que el marco de Jira Scrum es un enfoque de roles, actividades y artefactos, permite de gran facilidad de organizar y gestionar el trabajo de Sistema de Información de Recaudación tributaria de la Municipalidad Distrital de Santillana, ya que está basado en los valores de honestidad en el avance del proyecto, apertura, coraje, respeto, confianza, empoderarían y colaboración, con las personas como el Producto Owner, Scrum Master y Scrum Team. El Scrum Team debe cumplir en las fechas establecidas con la puntuación, el Scrum Master es el encargado del auto dirigido en el proyecto teniendo en cuenta como líder del equipo, tanto organizando y supervisando en el tablero de sprint. Tanto así evitando una incidencia o bug al término del proyecto y el aprendizaje del Jira Scrum Agile, cumpliendo satisfactoriamente el proyecto como muestra en el Figura 83 sin bugs.

Este trabajo de investigación demostró la satisfacción de las expectativas de los clientes y los equipos auto dirigidos y auto organizados. También sirva para la utilización de Jira Scrum Agile a muchas empresas para el desarrollo de los sistemas del gobierno peruano o de otros.

8.2. RECOMENDACIONES

- a) Se recomienda realizar la investigación de los proyectos de sistemas de recaudación tributaria de las Municipales para zonas de la costa y sierra, utilizando el marco de Jira Scrum Agile.
- b) Se sugiere a las empresas de tecnología utilizar esta nueva vanguardia de Jira Scrum a sus proyectos por su buena organización y gestión. Las herramientas de Jira Scrum que une a los equipos entorno a un solo objetivo y fomenta la entrega iterativa e incremental.

- c) Se recomienda analizar los Tableros Scrum y Tablero Kanban de las funcionalidades para el desarrollo de software de Jira.

- d) Se recomienda la reutilización del proyecto a otras Municipalidades Distritales de zona sierra, teniendo en cuenta la falta de accesibilidad de internet.

BIBLIOGRAFÍA

- Adkins, L. (2010). *Coaching Agile Teams: A Companion for ScrumMasters, Agile Coaches, and Project Managers in Transition*. Addison-Wesley Professional.
- Alicia, j. (2016). *Desarrollo de un Sistema de Información Tributaria*. Universidad Oberta de Catalunya.
- Álvarez, M. (2014). *Que es MVC: Teoría de la Programación Orientado de Objetos*. Obtenido de <https://desarrolloweb.com/articulos/que-es-mvc.html>
- Atlassian. (2017). *Jira Software. La herramienta de desarrollo de software líder de los equipos ágiles*.
- Bahit, E. (2012). *Scrum y Extreme Programming. Argentina, Buenos Aires*.
- Balaji, V. (2019). *Introducing Maven. A Build Tool for Today's Java Developers. 2nd ed.* Salt Lake City, UT, USA. doi:ISBN 978-1-4842-5409-7e-ISBN 978-1-4842-5410-3
- Barraza, O., Krol, F., Meléndez, L., & Velásquez, M. (2006). *Introducción a la programación orientados a objetos (1era ed.)*. México: Pearson educación.
- Buyto. (2012). *Diferencia entre aplicación web y desktop (o de escritorio)*. Obtenido de <http://www.buyto.es/general-diseno-web/diferencias-entre-aplicaciones-web-y-aplicaciones-desktop>
- Camacho, H., Castañeda, R., & Vargas, I. (2013). *Manual del curso: java Básico*. (México, Ed.) Universidad Nacional Autónoma de México.
- Catalani, E. (2007). *Arquitectura: modelo, vista y contro*. Obtenido de <https://exequielc.wordpress.com/2007/08/20/arquitectura-modelovistacontrolador/>
- Córdova, M. (2003). *Estadística Descriptiva e Inferencial*. Lima, Perú: Moshera S.R.L.
- David, H. (2018). *Creating and Running an Agile Project in JIRA*. doi:ISBN: 9781788835695.
- David, H. (2018). *Hands-On. Agile Software Development With JIRA*. doi:ISBN: 9781789532125.
- Díaz, R. (2009). *Las metodologías ágiles como garantía de calidad de software*.

- Gilfillan, I. (2003). *La biblia de MySQL*. Madrid, España: Anaya Multimedia, S.A.
- Gómez. (2019). *Java 2020 curso de programación (1ra ed.)*. Bogotá, Colombia: Lobo Gris - 2020.
- Gutiérrez, F. (2014). *Desarrollo de una aplicación web con spring framework para un gestor de un recetario. Proyecto de fin grado. Universidad Politécnica de Madrid*. Madrid, España.
- java. (s.f.). *¿Qué es java?* Obtenido de https://www.java.com/es/about/whatis_java.jsp?bucket_value=desktop-chrome77-windows10-64bit&in_query=no
- Javier, E. (2017). *Kanban y Scrumbar Orientados a Proyectos de Tecnología de la Información*. São Luís, Maranhão, Brasil.: Universitario Aeronáutico.
- Joyanes, L., & Zahonero, M. (2011). *Programación java (1era ed.): Algoritmo, programación orientado a objetos e interfaz gráfica de usuario*. México: Pearson educación.
- Karenyy, B. (2009). *Metodologías de desarrollo para la Aplicación Web*. Universidad de Cienfuegos.
- Ken, S., & Jeff, S. (2013). *La Guía de Scrum. La Guía Definitiva de Scrum: Las Reglas del Juego*.
- Kenneth, S. (2012). *Essential Scrum. A Practical Guide To The Most Popular Agile Process*. doi:ISBN 978-0-13-704329-3
- Llacctahuaman, j. (2015). *Sistema integral para mejorar la calidad de información en la recaudación tributaria de la municipalidad distrital de el tambo*. Universidad Nacional del Centro del Perú.
- López, P. (2004). *Población muestra y muestreo*. Obtenido de http://www.scielo.org.bo/scielo.php?script=sci_arttext&pid=S1815-02762004000100012
- M, X. (2016). *Implantación de scrum en la empresa con el soporte de la herramienta jira v2. Ingeniería Técnica en Informática de Gestión*.
- Marco normativa. (2015). *Manuales para la mejora de la recaudación del impuesto predial*. (F. (MEF)., Ed.) Perú.
- Mendoza. (2019). *¿Qué es el autovaluo en Perú?* Obtenido de <https://rebajatuscuentas.com/pe/blog/que-es-el-autovaluo-en-peru>.

- Monroy, S. (2008). *Estadística Descriptiva*. México.
- Osmandi, G. (2017). *¿Qué es javaFX?* Obtenido de <https://medium.com/@osmandi/qu%C3%A9-es-javafx-41cfc327bdf2>.
- Perú. (1993). *Decreto legislativo n° 776. Ley tributo municipal*. Perú.: el peruano.
- Pilar, R., & Juan, G. (2008). *Estudio de la Aplicación de Metodologías Ágiles para la Evolución de Productos Software*. Universidad Politécnica de Madrid.
- Rios, R. (2017). *Determinación de la recaudación de los impuestos municipales*. Pontificia Universidad Católica del Perú.
- Rubin, K. (2013). *Essential Scrum. USA: Pearson Educa*.
- Saneamiento, M. d. (1993). *Reglamento nacional de tasación del Perú. Resolución ministerial n° 127-2007-vivienda*. Perú.
- Schwaber, K., & Mike, B. (2001). *Agile Software Development with Scrum*. Prentice Hall.
- Schwaber, K., & Sutherland, J. (2017). *The Scrum Guide*.
- Sergio, j. (2009). *Aplicaciones de Escritorio vs Aplicaciones Web, ¿hay diferencia en el desarrollo?* Obtenido de <https://geeks.ms/sergiotarrillo/2009/01/14/aplicaciones-de-escritorio-vs-aplicaciones-web-hay-diferencia-en-el-desarrollo/>
- Silverschatz, A., Korth, H., & Sudarshan, S. (2006). *Fundamentos de Bases de Datos*. Madrid, España: MacGraw-Hill.
- Sommerville, I. (2010). *Ingeniería de software. España, Madrid: Pearson Education Limited*. España.
- Verónica, C. (2015). *Implementación de un sistema de información para la recaudación del impuesto de alcabala 'sipria' de la municipalidad provincial de puno*. (U. N. Altiplano, Ed.) Puno.
- Vila, J. (2014). *Dreamweaver vs. MVC (Modelo-Vista-Controlador o Framework)*. Obtenido de <https://blog.ayzweb.com/tutorial/dreamweaver-vs-mvc-modelo-vista-controlador-o-framework>
- Villanueva Ojeda, A. (2008). *Análisis, Diseño e Implementación de un DataWarehouse de Soporte de Decisiones para un Hospital del Sistema de Salud Público*. Lima.

- Yucra, B. (2015). *La gestión tributaria municipal y propuesta para el mejoramiento de la recaudación impuesto predial en Municipalidad Distrital de Cayma 2015*. Universidad Nacional de San Agustín de Arequipa.
- Zevala. (2019). *Informe legal. reglas para el pago del impuesto predial y arbitrios 2019*. Perú.: Cámara de comercio de lima.

ANEXOS

ANEXO 1. GUÍA DE OBSERVACIÓN DE EVALUACIÓN DEL SISTEMA “SYSAUTOVALUO” Y EL SISTEMA QUE SE VA CONSTRUIR

EVALUACIÓN	ESTANDAR	
	Sysautovaluo (lo que se tenía el sistema)	lo que se tendrá al construir el sistema
CODIGO FUENTE		
el código fuente se tiene		
el código fuente se puede mejorar		
el código fuente cuenta con estándares de seguridad		
tiene documentación o manuales		
BASE DE DATOS		
la Base de Datos se tiene		
se puede mejorar la Base de Datos		
cumple con las estándares de Desarrollo		
se tiene Usuario y contraseña al acceso al Base de Datos		
se puede recuperar la base de datos de todo los registros y pagos de la población		
el Base de Datos está desarrollado en una versión actual de un gestor de Base de datos		
ESTANDARES DE NORMA		
Resolución Ministerial N° 370-2018-Vivienda		
Decreto supremo N° 153-2004-EF Valores totales de predio		
resoluciones ministeriales N° 126-2007-Vivienda y N° 266-2012-Vivienda de la tabla de depreciación por antigüedad y estado de conservación según el material estructural predominante		
IMPUESTO PREDIAL		
Registro de Declaraciones Juradas		
Predios Urbanos y Rústicos		
Cálculo de Base Imponible		
Mantenimiento de Tablas para el cálculo		
Cálculo y recalcado automático		
Información confiable de los contribuyentes y su Predios		
ARBITRIOS MUNICIPALES		
Determinación de Arbitrios Municipales		
Calculo por fecha de declaración		
Mantenimiento de Tablas para el cálculo		
Cálculo y recalcado automático		

FISCALIZACIÓN		
Registro de la Base de Datos de la Fiscalización ejecutada sobre determinados contribuyentes		
Cálculo automático obteniendo las diferencias en las Cuentas Corrientes		
Resoluciones de Determinación		

ANEXO 2. FICHA DE INCIDENCIAS (HISTORIA DE USUARIOS)

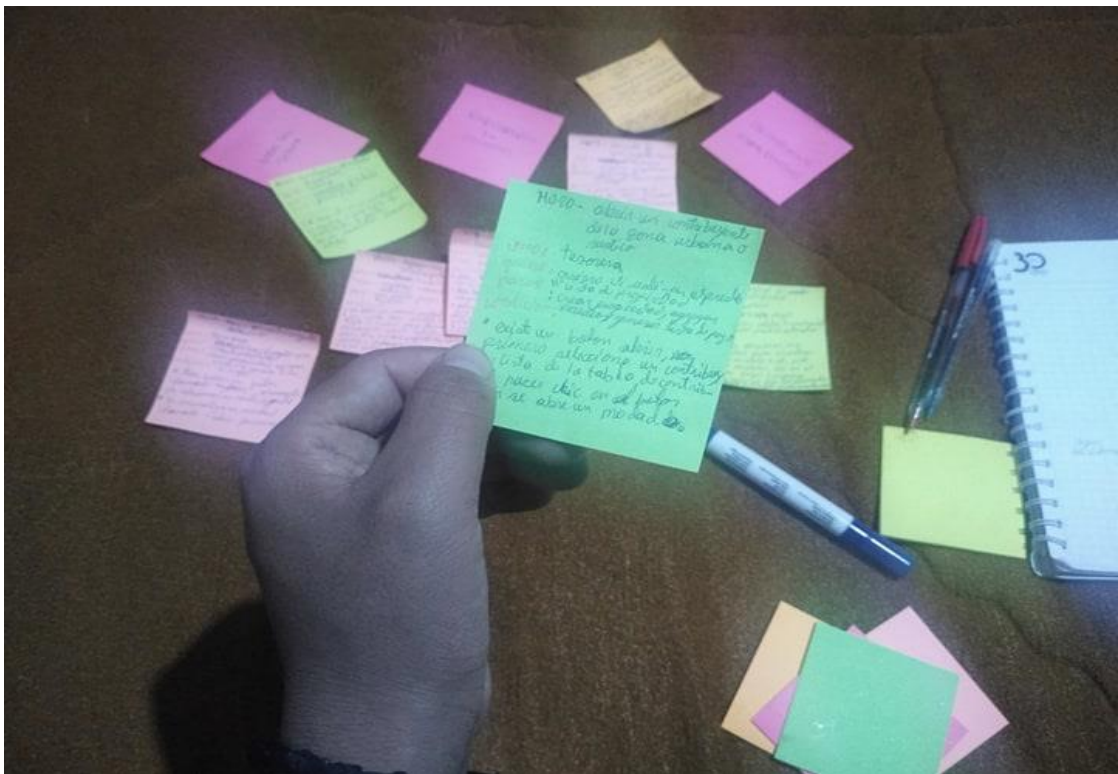
[SIS-__] HU__ – HISTORIA DE USUARIO Creada: __/__/__ Actualizada: __/__/__	
Estado:	
Proyecto:	
Componentes:	
Versiones afectadas:	
Versiones corregidas:	

Tipo:		Prioridad:	
Informador:		Responsable:	
Resolución:		Votos:	
Etiquetas:			
Estimación Restante:			
Tiempo Trabajado:			
Estimación original:			

Epic Link:	
Sprint:	

Descripción
<p>como:</p> <p>quiero:</p> <p>para:</p> <p>condición:</p>

ANEXO 3. FOTOGRAFÍAS DE LISTA DE ACTIVIDADES (HISTORIAL DE USUARIO) PARA LA ELABORACIÓN DE SCRUM.



ANEXO 4. CUADRO DE VALORES UNITARIOS



ANEXO I CUADRO DE VALORES UNITARIOS OFICIALES DE EDIFICACIONES PARA LA SIERRA AL 31 DE OCTUBRE DE 2018

VALORES POR PARTIDAS EN SOLES POR METRO CUADRADO DE AREA TECHADA						
E S T R U C T U R A S			A C A B A D O S			I N S T A L A C I O N E S E L É C T R I C A S Y S A N I T A R I A S (7)
MUROS Y COLUMNAS (1)	TECHOS (2)	PISOS (3)	PUERTAS Y VENTANAS (4)	REVESTI- MIENTOS (5)	BAÑOS (6)	
A ESTRUCTURAS LAMINARES CURVADAS DE CONCRETO ARMADO QUE INCLUYEN EN UNA SOLA ARMADURA LA CIMENTACIÓN Y EL TECHO, PARA ESTE CASO NO SE CONSIDERA LOS VALORES DE LA COLUMNA N°2	LOSA O ALIGERADO DE CONCRETO ARMADO CON LUCES MAYORES DE 6 M. CON SOBRECARGA MAYOR A 300 KG/M2	MÁRMOL IMPORTADO, PIEDRAS NATURALES IMPORTADAS, PORCELANATO.	ALUMINIO PESADO CON PERFILES ESPECIALES MADERA FINA ORNAMENTAL (CAOBA, CEDRO O PINO SELECTO) VIDRIO INSULADO. (1)	MÁRMOL IMPORTADO, MADERA FINA (CAOBA O SIMILAR) BALDOSA ACÚSTICO EN TECHO O SIMILAR.	BAÑOS COMPLETOS (7) DE LUJO IMPORTADO CON ENCHAPE FINO (MÁRMOL O SIMILAR)	AIRE ACONDICIONADO, ILUMINACIÓN ESPECIAL, VENTILACIÓN FORZADA, SIST. HIDRONEUMÁTICO, AGUA CALIENTE Y FRÍA, INTERCOMUNICADOR, ALARMAS, ASCENSOR, SISTEMA BOMBEO DE AGUA Y DESAGÜE.(5) TELÉFONO.
551.27	286.64	203.39	217.57	274.55	97.37	346.98
B COLUMNAS, VIGAS Y/O PLACAS DE CONCRETO ARMADO Y/O METÁLICAS.	ALIGERADOS O LOSAS DE CONCRETO ARMADO INCLINADAS	MÁRMOL NACIONAL O RECONSTITUIDO, PARQUET FINO (OLIVO, CHONTA O SIMILAR), CERÁMICA IMPORTADA MADERA FINA.	ALUMINIO O MADERA FINA (CAOBA O SIMILAR) DE DISEÑO ESPECIAL, VIDRIO TRATADO POLARIZADO (2) Y CURVADO, LAMINADO O TEMPLADO	MÁRMOL NACIONAL, MADERA FINA (CAOBA O SIMILAR) ENCHAPES EN TECHOS.	BAÑOS COMPLETOS (7) IMPORTADOS CON MAYÓLICA O CERÁMICO DECORATIVO IMPORTADO.	SISTEMA DE BOMBEO DE AGUA POTABLE, ASCENSOR TELÉFONO, AGUA CALIENTE Y FRÍA.
327.97	197.07	169.59	192.54	219.28	69.56	204.09
C PLACAS DE CONCRETO E=10 A 15 CM.ALBAÑILERÍA ARMADA, LADRILLO O SIMILAR CON COLUMNAS Y VIGAS DE AMARRE DE CONCRETO ARMADO	ALIGERADO O LOSAS DE CONCRETO ARMADO HORIZONTALES.	MADERA FINA MACHIHEMBRADA TERRAZO.	ALUMINIO O MADERA FINA (CAOBA O SIMILAR) VIDRIO TRATADO POLARIZADO. (2) LAMINADO O TEMPLADO	SUPERFICIE CARAVISTA OBTENIDA MEDIANTE ENCOFRADO ESPECIAL, ENCHAPE EN TECHOS.	BAÑOS COMPLETOS (7) NACIONALES CON MAYÓLICA O CERÁMICO NACIONAL DE COLOR.	IGUAL AL PUNTO "B" SIN ASCENSOR.
237.95	137.90	109.74	140.48	181.49	45.41	151.86
D LADRILLO, SILLAR O SIMILAR, SIN ELEMENTOS DE CONCRETO ARMADO, DRYWALL O SIMILAR INCLUYE TECHO (6)	CALAMINA METÁLICA FIBROCEMENTO SOBRE VIGUERÍA METÁLICA.	PARQUET DE 1era. , LAJAS, CERÁMICA NACIONAL, LOSETA VENECIANA 40x40, PISO LAMINADO.	VENTANAS DE ALUMINIO PUERTAS DE MADERA SELECTA, VIDRIO TRATADO TRANSPARENTE (3)	ENCHAPE DE MADERA O LAMINADOS, PIEDRA O MATERIAL VITRIFICADO.	BAÑOS COMPLETOS (7) NACIONALES CON MAYÓLICA BLANCA.	AGUA FRÍA, AGUA CALIENTE, CORRIENTE TRIFÁSICA, TELÉFONO.
219.79	93.36	89.98	82.39	138.82	27.76	86.05
E ADOBE, TAPIAL O QUINCHA	MADERA CON MATERIAL IMPERMEABILIZANTE.	PARQUET DE 2da. LOSETA VENECIANA 30x30 LAJAS DE CEMENTO CON CANTO RODADO.	VENTANAS DE FIERRO PUERTAS DE MADERA SELECTA (CAOBA O SIMILAR) VIDRIO SIMPLE TRANSPARENTE (4)	SUPERFICIE DE LADRILLO CARAVISTA.	BAÑOS CON MAYÓLICA BLANCA PARCIAL.	AGUA FRÍA, AGUA CALIENTE, CORRIENTE MONOFÁSICA, TELÉFONO.
172.54	42.86	74.42	62.94	115.49	13.62	47.89
F MADERA (ESTORAQUE, PUMAQUIRO, HUAYRURO, MACHINGA, CATAHUA AMARILLA, COPAIBA, DIABLO FUERTE, TORNILLO O SIMILARES) DRY WALL O SIMILAR (SIN TECHO)	CALAMINA METÁLICA FIBROCEMENTO O TEJA SOBRE VIGUERÍA DE MADERA CORRIENTE.	LOSETA CORRIENTE, CANTO RODADO. ALFOMBRA	VENTANAS DE FIERRO O ALUMINIO INDUSTRIAL, PUERTAS CONTRAPLACADAS DE MADERA (CEDRO O SIMILAR), PUERTAS MATERIAL MDF o HDF. VIDRIO SIMPLE TRANSPARENTE (4)	TARRAJEO FROTACHADO Y/O YESO MOLDURADO, PINTURA LAVABLE.	BAÑOS BLANCOS SIN MAYÓLICA.	AGUA FRÍA, CORRIENTE MONOFÁSICA, TELÉFONO
107.59	34.24	60.78	48.67	68.86	11.58	31.13
G PIRCADO CON MEZCLA DE BARRO.	SIN TECHO	LOSETA VINILICA, CEMENTO BRUÑADO COLOREADO. TAPIZÓN.	MADERA CORRIENTE CON MARCOS EN PUERTAS Y VENTANAS DE PVC O MADERA CORRIENTE	ESTUCADO DE YESO Y/O BARRO, PINTURA AL TEMPLE O AGUA.	SANITARIOS BÁSICOS DE LOSA DE 2da, FIERRO FUNDIDO O GRANITO.	AGUA FRÍA, CORRIENTE MONOFÁSICA SIN EMPOTRAR.
63.39	0.00	45.46	28.68	51.16	7.96	18.34
H		CEMENTO PULIDO, LADRILLO CORRIENTE, ENTABLADO CORRIENTE.	MADERA RÚSTICA.	PINTADO EN LADRILLO RÚSTICO, PLACA DE CONCRETO O SIMILAR.	SIN APARATOS SANITARIOS.	SIN INSTALACIÓN ELÉCTRICA NI SANITARIA.
.....	24.56	14.34	20.46	0.00	0.00
I		TIERRA COMPACTADA	SIN PUERTAS NI VENTANAS.	SIN REVESTIMIENTOS EN LADRILLO, ADOBE O SIMILAR.		
.....	5.40	0.00	0.00

EN EDIFICIOS AUMENTAR EL VALOR POR M2 EN 5% A PARTIR DEL 5 PISO

EL VALOR UNITARIO POR M2 PARA UNA EDIFICACIÓN DETERMINADA SE OBTIENE SUMANDO LOS VALORES SELECCIONADOS DE UNA DE LAS 7 COLUMNAS DEL CUADRO, DE ACUERDO A SUS CARACTERÍSTICAS PREDOMINANTES. LA DEMARCAÇÃO TERRITORIAL CONSIGNADA ES DE USO EXCLUSIVO PARA LA APLICACIÓN DEL PRESENTE CUADRO. ABARCA LAS LOCALIDADES UBICADAS EN LA FAJA LONGITUDINAL DEL TERRITORIO LIMITADA, AL NORTE POR LA FRONTERA CON ECUADOR, AL SUR POR LA FRONTERA CON CHILE Y BOLIVIA. AL OESTE POR LA CURVA DE NIVEL DE 2000 m.s.n.m QUE LA SEPARA DE LA COSTA ESTE, POR UNA CURVA DE NIVEL QUE LA SEPARA DE LA SELVA, QUE PARTIENDO DE LA FRONTERA CON EL ECUADOR, CONTINÚA HASTA SU CONFLUENCIA CON EL RÍO NOVA, AFLUENTE DEL SAN ALEJANDRO, EN DONDE ASCIENDE HASTA LA COTA 2000 Y CONTINÚA POR ESTA HACIA EL SUR HASTA SU CONFLUENCIA CON EL RÍO SANABENI, AFLUENTE DEL ENE, DE ESTE PUNTO BAJA HASTA LA COTA 1500 Y CONTINÚA HASTA LA FRONTERA CON BOLIVIA.

- (1) REFERIDO AL DOBLE VIDRIADO HERMÉTICO, CON PROPIEDADES DE AISLAMIENTO TÉRMICO Y ACÚSTICO.
- (2) REFERIDO AL VIDRIO QUE RECIBE TRATAMIENTO PARA INCREMENTAR SU RESISTENCIA MECÁNICA Y PROPIEDADES DE AISLAMIENTO ACÚSTICO Y TÉRMICO, SON COLOREADOS EN SU MASA PERMITIENDO LA VISIBILIDAD ENTRE 14% Y 83%.
- (3) REFERIDO AL VIDRIO QUE RECIBE TRATAMIENTO PARA INCREMENTAR SU RESISTENCIA MECÁNICA Y PROPIEDADES DE AISLAMIENTO ACÚSTICO Y TÉRMICO, PERMITEN LA VISIBILIDAD ENTRE 75% Y 92%.
- (4) REFERIDO AL VIDRIO PRIMARIO SIN TRATAMIENTO, PERMITEN LA TRANSMISIÓN DE LA VISIBILIDAD ENTRE 75% Y 92%.
- (5) SISTEMA DE BOMBEO DE AGUA Y DESAGÜE, REFERIDO A INSTALACIONES INTERIORES SUBTERRÁNEAS (CISTERNAS, TANQUES SÉPTICOS) Y AÉREAS (TANQUES ELEVADOS) QUE FORMAN PARTE INTEGRANTE DE LA EDIFICACIÓN
- (6) PARA ESTE CASO NO SE CONSIDERA LA COLUMNA N° 2
- (7) SE CONSIDERA COMO MÍNIMO LAVATORIO, INODORO Y DUCHA O TINA.

ANEXO 5. TABLA DE DEPRECIACIÓN

TABLA N° 4
**PORCENTAJES PARA EL CÁLCULO DE LA DEPRECIACIÓN POR ANTIGÜEDAD Y
 ESTADO DE CONSERVACIÓN SEGÚN EL MATERIAL ESTRUCTURAL PREDOMINANTE
 PARA EDIFICACIONES DE SALUD, CINES, INDUSTRIAS, EDIFICACIONES
 DE USO EDUCATIVO, TALLERES.**

Antigüedad (en años)	Material Estructural Predominante	ESTADO DE CONSERVACIÓN			
		Muy Bueno %	Bueno %	Regular %	Malo %
Hasta 5 Años	Concreto	0	5	20	59
	Ladrillo	0	12	24	63
	Adobe	9	21	34	69
Hasta 10 Años	Concreto	3	10	22	61
	Ladrillo	5	16	28	68
	Adobe	14	26	39	74
Hasta 15 Años	Concreto	6	13	25	64
	Ladrillo	9	20	32	72
	Adobe	19	30	44	79
Hasta 20 Años	Concreto	9	16	27	67
	Ladrillo	13	24	36	77
	Adobe	24	35	49	84
Hasta 25 Años	Concreto	12	18	30	70
	Ladrillo	17	28	40	81
	Adobe	29	40	52	89
Hasta 30 Años	Concreto	15	20	32	72
	Ladrillo	21	32	44	83
	Adobe	34	45	59	*
Hasta 35 Años	Concreto	18	23	34	75
	Ladrillo	25	36	48	*
	Adobe	39	50	64	*
Hasta 40 Años	Concreto	21	26	37	77
	Ladrillo	29	40	52	*
	Adobe	44	54	69	*
Hasta 45 Años	Concreto	24	29	39	80
	Ladrillo	33	44	56	*
	Adobe	49	59	74	*
Hasta 50 Años	Concreto	27	32	42	*
	Ladrillo	37	48	60	*
	Adobe	54	64	79	*
Más de 50 Años	Concreto	30	35	44	*
	Ladrillo	41	52	64	*
	Adobe	60	70	84	*

* El perito deberá estimar los porcentajes no tabulados.

NOTA: En el caso de la calificación del estado de conservación muy malo, el perito establecerá a su criterio el porcentaje de depreciación.

TABLA N° 3
PORCENTAJES PARA EL CÁLCULO DE LA DEPRECIACIÓN POR ANTIGÜEDAD
Y ESTADO DE CONSERVACIÓN SEGÚN EL MATERIAL ESTRUCTURAL
PREDOMINANTE PARA EDIFICIOS - OFICINAS

Antigüedad (en años)	Material Estructural Predominante	ESTADO DE CONSERVACIÓN			
		Muy Bueno %	Bueno %	Regular %	Malo %
Hasta 5 Años	Concreto	0	5	10	55
	Ladrillo	0	8	20	60
	Adobe	9	19	34	69
Hasta 10 Años	Concreto	3	8	13	58
	Ladrillo	5	13	25	65
	Adobe	14	24	39	74
Hasta 15 Años	Concreto	6	11	16	61
	Ladrillo	9	17	29	69
	Adobe	19	29	44	79
Hasta 20 Años	Concreto	9	14	19	64
	Ladrillo	13	21	33	73
	Adobe	24	34	49	84
Hasta 25 Años	Concreto	12	17	22	67
	Ladrillo	17	25	37	77
	Adobe	29	39	54	89
Hasta 30 Años	Concreto	15	20	25	70
	Ladrillo	21	29	41	81
	Adobe	34	44	59	*
Hasta 35 Años	Concreto	18	23	28	73
	Ladrillo	25	33	45	85
	Adobe	39	49	64	*
Hasta 40 Años	Concreto	21	26	31	76
	Ladrillo	29	37	49	89
	Adobe	44	54	69	*
Hasta 45 Años	Concreto	24	29	34	79
	Ladrillo	33	41	53	*
	Adobe	49	59	74	*
Hasta 50 Años	Concreto	27	32	37	82
	Ladrillo	37	45	57	*
	Adobe	54	64	79	*
Más de 50 Años	Concreto	30	35	40	85
	Ladrillo	41	49	61	*
	Adobe	59	69	84	*

* El perito deberá estimar los porcentajes no tabulados.

NOTA: En el caso de la calificación del estado de conservación muy malo, el perito establecerá a su criterio el porcentaje de depreciación.

TABLA Nº 2
PORCENTAJES PARA EL CÁLCULO DE LA DEPRECIACIÓN POR ANTIGÜEDAD Y
ESTADO DE CONSERVACIÓN SEGÚN EL MATERIAL ESTRUCTURAL PREDOMINANTE
PARA TIENDAS, DEPÓSITOS, CENTROS DE RECREACIÓN o ESPARCIMIENTO,
CLUBES SOCIALES o INSTITUCIONES

Antigüedad (en años)	Material Estructural Predominante	ESTADO DE CONSERVACIÓN			
		Muy Bueno %	Bueno %	Regular %	Malo %
Hasta 5 Años	Concreto	0	5	10	55
	Ladrillo	0	8	20	60
	Adobe	7	17	32	67
Hasta 10 Años	Concreto	2	7	12	57
	Ladrillo	4	12	24	64
	Adobe	12	22	37	72
Hasta 15 Años	Concreto	5	10	15	60
	Ladrillo	8	16	28	68
	Adobe	17	27	42	77
Hasta 20 Años	Concreto	8	13	18	63
	Ladrillo	12	20	32	72
	Adobe	22	32	47	82
Hasta 25 Años	Concreto	11	16	21	66
	Ladrillo	16	24	36	76
	Adobe	27	37	52	87
Hasta 30 Años	Concreto	14	19	24	69
	Ladrillo	20	28	40	80
	Adobe	32	42	57	*
Hasta 35 Años	Concreto	17	22	27	72
	Ladrillo	24	32	44	84
	Adobe	37	47	62	*
Hasta 40 Años	Concreto	20	25	30	75
	Ladrillo	28	36	48	88
	Adobe	42	52	67	*
Hasta 45 Años	Concreto	23	28	33	78
	Ladrillo	32	40	52	*
	Adobe	47	57	72	*
Hasta 50 Años	Concreto	26	31	36	81
	Ladrillo	36	44	56	*
	Adobe	52	62	77	*
Más de 50 Años	Concreto	29	34	39	84
	Ladrillo	40	48	60	*
	Adobe	57	67	82	*

* El perito deberá estimar los porcentajes no tabulados.

NOTA: En el caso de la calificación del estado de conservación muy malo, el perito establecerá a su criterio el porcentaje de depreciación.

TABLA Nº 2
PORCENTAJES PARA EL CÁLCULO DE LA DEPRECIACIÓN POR ANTIGÜEDAD Y
ESTADO DE CONSERVACIÓN SEGÚN EL MATERIAL ESTRUCTURAL PREDOMINANTE
PARA TIENDAS, DEPÓSITOS, CENTROS DE RECREACIÓN o ESPARCIMIENTO,
CLUBES SOCIALES o INSTITUCIONES

Antigüedad (en años)	Material Estructural Predominante	ESTADO DE CONSERVACIÓN			
		Muy Bueno %	Bueno %	Regular %	Malo %
Hasta 5 Años	Concreto	0	5	10	55
	Ladrillo	0	8	20	60
	Adobe	7	17	32	67
Hasta 10 Años	Concreto	2	7	12	57
	Ladrillo	4	12	24	64
	Adobe	12	22	37	72
Hasta 15 Años	Concreto	5	10	15	60
	Ladrillo	8	16	28	68
	Adobe	17	27	42	77
Hasta 20 Años	Concreto	8	13	18	63
	Ladrillo	12	20	32	72
	Adobe	22	32	47	82
Hasta 25 Años	Concreto	11	16	21	66
	Ladrillo	16	24	36	76
	Adobe	27	37	52	87
Hasta 30 Años	Concreto	14	19	24	69
	Ladrillo	20	28	40	80
	Adobe	32	42	57	*
Hasta 35 Años	Concreto	17	22	27	72
	Ladrillo	24	32	44	84
	Adobe	37	47	62	*
Hasta 40 Años	Concreto	20	25	30	75
	Ladrillo	28	36	48	88
	Adobe	42	52	67	*
Hasta 45 Años	Concreto	23	28	33	78
	Ladrillo	32	40	52	*
	Adobe	47	57	72	*
Hasta 50 Años	Concreto	26	31	36	81
	Ladrillo	36	44	56	*
	Adobe	52	62	77	*
Más de 50 Años	Concreto	29	34	39	84
	Ladrillo	40	48	60	*
	Adobe	57	67	82	*

* El perito deberá estimar los porcentajes no tabulados.

NOTA: En el caso de la calificación del estado de conservación muy malo, el perito establecerá a su criterio el porcentaje de depreciación.

TABLA N° 1
PORCENTAJES PARA EL CÁLCULO DE LA DEPRECIACIÓN POR ANTIGÜEDAD Y
ESTADO DE CONSERVACIÓN SEGÚN EL MATERIAL ESTRUCTURAL PREDOMINANTE
PARA CASAS HABITACIÓN Y DEPARTAMENTOS PARA VIVIENDAS
INCLUIDO LOS UBICADOS EN EDIFICIOS

Antigüedad (en años)	Material Estructural Predominante	ESTADO DE CONSERVACIÓN			
		Muy Bueno %	Bueno %	Regular %	Malo %
Hasta 5 Años	Concreto	0	5	10	55
	Ladrillo	0	8	20	60
	Adobe	5	15	30	65
Hasta 10 Años	Concreto	0	5	10	55
	Ladrillo	3	11	23	63
	Adobe	10	20	35	70
Hasta 15 Años	Concreto	3	8	13	58
	Ladrillo	6	14	26	66
	Adobe	15	25	40	75
Hasta 20 Años	Concreto	6	11	16	61
	Ladrillo	9	17	29	69
	Adobe	20	30	45	80
Hasta 25 Años	Concreto	9	14	19	64
	Ladrillo	12	20	32	72
	Adobe	25	35	50	85
Hasta 30 Años	Concreto	12	17	22	67
	Ladrillo	15	23	35	75
	Adobe	30	40	55	90
Hasta 35 Años	Concreto	15	20	25	70
	Ladrillo	18	26	38	78
	Adobe	35	45	60	*
Hasta 40 Años	Concreto	18	23	28	73
	Ladrillo	21	29	41	81
	Adobe	40	50	65	*
Hasta 45 Años	Concreto	21	26	31	76
	Ladrillo	24	32	44	84
	Adobe	45	55	70	*
Hasta 50 Años	Concreto	24	29	34	79
	Ladrillo	27	35	47	87
	Adobe	50	60	75	*
Más de 50 Años	Concreto	27	32	37	82
	Ladrillo	30	38	50	90
	Adobe	55	65	80	*

* El perito deberá estimar los porcentajes no tabulados.

NOTA: En el caso de la calificación del estado de conservación muy malo, el perito establecerá a su criterio el porcentaje de depreciación.

ANEXO 6. AVANCES DEL TRABAJO EN EL TABLERO DE SPRINT DE JIRA SCRUM

Proyectos / SistemaRT / Tablero SIS

Tablero Sprint 1

☆ 0 días restantes Terminar sprint

desarrollar la parte de login y registrador de usuario.




Tablero Sprint 2

☆ 0 días restantes Terminar sprint

desarrollo del contribuyente urbano



ANEXO 7. BOLETA Y DECLARACIÓN JURADA DE AUTOVALUO



MUNICIPALIDAD DISTRITAL DE SANTILLANA
OFICINA DE RECAUDACIÓN Y TESORERÍA
Dirección: Plaza Principal SIN - San José de Sece
Concepto de Pago al Impuesto Predial

R.U.C. 20143157483
BOLETA
N° 00000010

SELLO DE RECEPCIÓN

DECLARACIÓN JURADA DE AUTOVALUO
T.U.O. de la Ley de Tributación Municipal Aprobado por el Decreto Supremo N° 156-20-I-F

PU
(PREDIO URBANO)

MUNICIPALIDAD DISTRITAL DE SANTILLANA
R.U.C. 20143157483

CODIGO DE PREDIO 00000010-PU **AÑO FISCAL** 2019 **Nº DD. J.J.** 00000010

IDENTIFICACIÓN DEL CONTRIBUYENTE
NÚMERO DNI o RUC 280593910 **APELLIDOS Y NOMBRES O RAZÓN SOCIAL** POTOSINO QUISEP SEGUNDA(E)RROR

UBICACIÓN DEL PREDIO
LUGAR O DISTRITO SANTILLANA **DENOMINACIÓN** CP.
DIRECCIÓN REF. SAN JOSE DE SECE **NÚMERO** SIN **Mz.** D **LOTE** 2

DATOS RELATIVOS AL PREDIO
CONDICIÓN DE PROPIEDAD SOCIEDAD CONYUGAL

ESTADO DE CONSTRUCCIÓN TERMINADO **TIPO DE PREDIO** PREDIO TOTAL **USO DEL PREDIO** casa habitación dep. para vivienda

DATOS RELATIVOS AL PREDIO PARA EFECTOS DE LA DEPRECIACIÓN
CLASIFICACIÓN DEL PREDIO casa habitación dep. para vivienda **MATERIAL ESTRUCTURAL PREDOMINANTE** adobe, rustico **ESTADO DE CONSERVACIÓN** malo

DETERMINACIÓN DEL AUTOVALUO (Datos de la Construcción)																		
Nº Pto	VALORES UNITARIOS DE DEPRECIACIÓN			Inces. Depreciación %	Valor de Depreciación	Área Const. m2	Valor Área Depreciada	Áreas Const. m2	Valor de la Construcción									
	Anti	M	P															
A. 1.22		E	F	H	I	G												
					265.32	0.00	172.46	65.00	10612.86	0.00	3714.40							
TOTAL A PAGAR S/ 57.91																		

ÁREA DEL TERRENO(m2): 1802.80 **APANCEL S/** 14.00 **VALOR TOTAL DE LA CONSTRUCCIÓN:** 3714.48

REGIMEN DE INAFECTACIÓN / EXONERACIÓN Normal **VALOR ESTIMADO DE OTRAS INSTALACIONES:** 0.00

REGIMEN IN° EXPED | N° RESOL | Fecha(Desde-Hasta) **VALOR TOTAL DEL TERRENO:** 25239.20

TOTAL DE AUTOVALUO: 28953.68

8 AUGUST 2019

MUNICIPALIDAD DISTRITAL DE SANTILLANA
OFICINA DE RECAUDACIÓN Y TESORERÍA
Dirección: Plaza Principal SIN - San José de Sece
Concepto de Pago al Impuesto Predial

PREDIO URBANO
ANO 2019

Nº DNI o RUC 280593910 **Fecha** 08/08/2019

Dirección Domiciliaria CP. SAN JOSE DE SECE N° SIN Mz. D Lt. 2

***UIT Actual S/** 3950.00 **Total Años a Declarar** 1

Derecho de Emisión 0 **Años (Desde - Hasta)** 2019

Base Imponible de PU 28953.68 **Impuesto Anual** 57.91 **Impuesto Trimestral** 14.48

LIQUIDACIÓN DE PAGOS:	Impuesto	Mora / Interes	Sub Total	Estado Pago	Vencimiento
1er Trimestre	14.48	0	14.48	<input checked="" type="checkbox"/> Pagar	A
2do Trimestre	14.48	0	14.48	<input checked="" type="checkbox"/> Pagar	A
3er Trimestre	14.48	0	14.48	<input checked="" type="checkbox"/> Pagar	A
4to Trimestre	14.48	0	14.48	<input checked="" type="checkbox"/> Pagar	A

TOTAL A PAGAR S/ 57.91

SON: CINCUENTA Y SIETE 91/100 SOLES.

ANEXO 8. CÓDIGO FUENTE

a. CONTROLADOR

Class Cabrirtc

```
/*
 * To change this license header, choose License Headers in
 * Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package control;

import dao.DCaja;
import dao.DPropiedad;
import dao.DaoPredio;
import dao.DaoValorpropiedad;
import java.io.IOException;
import java.math.BigDecimal;
import java.net.URL;
import java.time.Instant;
import java.util.Date;
import java.util.Iterator;
import java.util.List;
import java.util.ResourceBundle;
import javafx.collections.FXCollections;
import javafx.collections.ObservableList;
import javafx.fxml.FXML;
import javafx.fxml.FXMLLoader;
import javafx.fxml.Initializable;
import javafx.scene.Scene;
import javafx.scene.control.Alert;
import javafx.scene.control.ComboBox;
import javafx.scene.control.Label;
import javafx.scene.control.TableColumn;
import javafx.scene.control.TableView;
import javafx.scene.control.TextField;
import javafx.scene.layout.AnchorPane;
import javafx.stage.Modality;
import javafx.stage.Stage;
import modelo.Caja;
import modelo.Predio;
import modelo.Propiedad;
import modelo.Proprietario;
import modelo.Valorpropiedad;
import mysql.DaoPredioM;

/**
 *
 * @author YONNY
 */
public class Cabrirtc implements Initializable {

    @FXML
    public Label pnom;
    @FXML
    public Label pnum;
    @FXML
```

```

public Label pcel;
@FXML
private TableView<Propiedad> listap;
@FXML
private TableColumn<Propiedad, String> c1;
@FXML
private TableColumn<Propiedad, String> c2;
@FXML
private TableColumn<Propiedad, String> c3;

public ObservableList<Propiedad> ls =
FXCollections.observableArrayList();

/**
 * tabla dos*
 */
@FXML
TextField tp1, tp2, tp3;
@FXML
TextField tp11, tp21, tp31;
@FXML
TextField tp111, tp211;
@FXML
ComboBox cp1;
@FXML
private TableView<Predio> listapredio;
@FXML
private TableColumn<Predio, String> pca;
@FXML
private TableColumn<Predio, String> c;
@FXML
private TableColumn<Predio, String> m;
@FXML
private TableColumn<Predio, String> e;
@FXML
private TableColumn<Predio, String> pc1;
@FXML
private TableColumn<Predio, String> pc2;
@FXML
private TableColumn<Predio, String> pc3;
@FXML
private TableColumn<Predio, String> pc4;
@FXML
private TableColumn<Predio, String> pc5;
@FXML
private TableColumn<Predio, String> pc6;
@FXML
private TableColumn<Predio, String> pc7;
@FXML
private TableColumn<Predio, String> pcv1;//valor de
construccion
@FXML
private TableColumn<Predio, String> pcv2;//area de
construccion
@FXML
private TableColumn<Predio, String> pcv3;//deprec

public ObservableList<Predio> lss =
FXCollections.observableArrayList();

```

```

/**
 * tabla dos*
 */
@FXML
private TableView<Caja> lista3;
@FXML
private TableColumn<Caja, String> a1;
@FXML
private TableColumn<Caja, String> a2;
@FXML
private TableColumn<Caja, String> a3;
@FXML
private TableColumn<Caja, String> a4;

public ObservableList<Caja> ls3 =
FXCollections.observableArrayList();

@Override
public void initialize(URL location, ResourceBundle
resources) {

}

public void datap(Propietario pr) {
    ls.clear();
    try {

listap.getSelectionModel().selectedItemProperty().addListener((o
bservable, oldValue, po) -> load(po));

        DPropiedad reg1 = new DPropiedad();
        List l = reg1.all(pr);

        for (Iterator iterator = l.iterator();
iterator.hasNext();) {
            Propiedad next = (Propiedad) iterator.next();
            ls.add(next);
        }
        c1.setCellValueFactory(da ->
da.getValue().direccion());
        c2.setCellValueFactory(da ->
da.getValue().distrito());
        c3.setCellValueFactory(da ->
da.getValue().provincia());
        listap.setItems(ls);
    } catch (NullPointerException er) {

    }

}

public void load(Propiedad po) {
    anos(po);
    caja(po);
    datapredio(po);
}

public void anos(Propiedad po) {
    cp1.getItems().clear();

```

```

    try {
        DPropiedad dp = new DPropiedad();
        dp.anos(po).stream().map((h) -> {
            cp1.getItems().addAll(h.getAno());
            return h;
        }).forEachOrdered((_item) -> {
            cp1.getSelectionModel().select(0);
        });
    } catch (NullPointerException er) {
    }
}

@FXML
public void Predios() {
    Propiedad pr =
listap.getSelectionModel().getSelectedItem();
    datapredio(pr);
}

public void datapredio(Propiedad pr) {
    lss.clear();
    try {
        DPropiedad dp = new DPropiedad();

        List<Valorpropiedad> p =
dp.valor(Integer.parseInt(cp1.getValue().toString()), pr);
        if (!p.isEmpty()) {
            tp1.setText(p.get(0).getArancel().toString());
            tp2.setText(p.get(0).getAreat().toString());
            tp3.setText(p.get(0).getValort().toString());
        } else {
            tp1.setText("");
            tp2.setText("");
            tp3.setText("");
        }

        List l =
dp.predios(Integer.parseInt(cp1.getValue().toString()), pr);
        for (Iterator iterator = l.iterator();
iterator.hasNext();) {
            Predio next = (Predio) iterator.next();
            lss.add(next);
        }
        listapredio.setItems(lss);
        pca.setCellValueFactory(da -> da.getValue().ano());
        c.setCellValueFactory(da ->
da.getValue().Clasificacion());
        m.setCellValueFactory(da ->
da.getValue().Material());
        e.setCellValueFactory(da -> da.getValue().Estado());
        pc1.setCellValueFactory(da -> da.getValue().muro());
        pc2.setCellValueFactory(da ->
da.getValue().techo());
        pc3.setCellValueFactory(da ->
da.getValue().pisos());
        pc4.setCellValueFactory(da ->
da.getValue().puerta());
        pc5.setCellValueFactory(da ->
da.getValue().revestimiento());
    }
}

```

```

        pc6.setCellValueFactory(da -> da.getValue().bano());
        pc7.setCellValueFactory(da ->
da.getValue().instalacion());
        pcv1.setCellValueFactory(da ->
da.getValue().vconstruccion());
        pcv2.setCellValueFactory(da ->
da.getValue().aconstruccion());
        pcv3.setCellValueFactory(da ->
da.getValue().dconstruccion());
    } catch (NullPointerException er) {
        tp1.setText("");
        tp2.setText("");
        tp3.setText("");
    }
}

public void caja(Propiedad p) {
    ls3.clear();
    try {
        DCaja reg = new DCaja();
        List l = reg.all(p.getIdprop());
        for (Iterator iterator = l.iterator();
iterator.hasNext();) {
            Caja next = (Caja) iterator.next();
            ls3.add(next);
        }
        a1.setCellValueFactory(da -> da.getValue().ano());
        a2.setCellValueFactory(da -> da.getValue().base());
        a3.setCellValueFactory(da -> da.getValue().monto());
        a4.setCellValueFactory(da ->
da.getValue().estado());
        lista3.setItems(ls3);
    } catch (NullPointerException er) {
        System.out.println(er);
    }
}

//modal de crear predio
@FXML
private void crearp() {
    Propiedad re = new Propiedad();
    boolean okClicked = EditDialog(re);
    if (okClicked) {
        datap(reg);
    }
}

@FXML
private void MPro() {

    try {
        Propiedad pr =
listap.getSelectionModel().getSelectedItem();
        boolean okClicked = EditDialog(pr);
        if (okClicked) {
            datap(reg);
        }
    } catch (Exception e) {
        Alert al = new Alert(Alert.AlertType.ERROR);

```

```

        al.setHeaderText("Error al Modificar");
        al.setContentText("Seleccione una fila");
        al.showAndWait();
        System.out.println(e);
    }
}

@FXML
private void EAno() {
    try {
        int a = Integer.parseInt(cp1.getValue().toString());
        Propiedad pr =
listap.getSelectionModel().getSelectedItem();
        DPropiedad dp = new DPropiedad();
        dp.EAno(pr, a);
        anos(pr);

        Alert al = new Alert(Alert.AlertType.INFORMATION);
        al.setHeaderText("Eliminado Correctamente");
        al.setContentText("Año: " + String.valueOf(a));
        al.showAndWait();

    } catch (Exception e) {
        Alert al = new Alert(Alert.AlertType.ERROR);
        al.setHeaderText("Error al Eliminar");
        al.setContentText("Seleccione un Año");
        al.showAndWait();
    }
}

@FXML
private void DPro() {
    try {
        Alert al = new Alert(Alert.AlertType.CONFIRMATION);
        al.setHeaderText("Eliminar Pago");
        al.setContentText("¿Está seguro al Eliminar Pago?");
        al.showAndWait();
        if
(al.getResult().getButtonData().isDefaultButton()) {
            int id =
lista3.getSelectionModel().getSelectedIndex();
            DCaja dp = new DCaja();
            Caja pr =
lista3.getSelectionModel().getSelectedItem();
            if (!pr.isEstado()) {
                dp.eliminar(pr);
                lista3.getItems().remove(id);
            } else {
                Alert la = new Alert(Alert.AlertType.ERROR);
                la.setHeaderText("Error al Eliminar");
                la.setContentText("El Predio ya fue
Pagado");
                la.showAndWait();
            }
        }
    } catch (Exception e) {
        Alert al = new Alert(Alert.AlertType.ERROR);

```

```

        al.setHeaderText("Error al Eliminar");
        al.setContentText("Seleccione una fila o El Predio
ya fue Pagado");
        al.showAndWait();
        System.out.println(e);
    }

}

@FXML
private void EPredio() {
    try {
        Alert al = new Alert(Alert.AlertType.CONFIRMATION);
        al.setHeaderText("Eliminar Predio");
        al.setContentText("¿Está seguro al Eliminar
Predio?");
        al.showAndWait();
        if
(al.getResult().getButtonData().isDefaultButton()) {
            DaoPredioM dp = new DaoPredioM();
            Predio pr =
listapredio.getSelectionModel().getSelectedItem();
            dp.EPredio(pr);
            Predios();

            Alert a = new
Alert(Alert.AlertType.INFORMATION);
            a.setHeaderText("Eliminado Correctamente");
            a.setContentText("Año: " +
String.valueOf(pr.getAno()));
            a.showAndWait();
        }
    } catch (Exception e) {
        Alert al = new Alert(Alert.AlertType.ERROR);
        al.setHeaderText("Error al Eliminar");
        al.setContentText("Seleccione una fila del Predio");
        al.showAndWait();
        System.out.println(e);
    }
}

@FXML
private void EPropiedad() {
    try {
        Alert al = new Alert(Alert.AlertType.CONFIRMATION);
        al.setHeaderText("Eliminar Propiedad");
        al.setContentText("¿Está seguro al Eliminar
Propiedad?");
        al.showAndWait();
        if
(al.getResult().getButtonData().isDefaultButton()) {
            DaoPredioM dp = new DaoPredioM();
            Propiedad pr =
listap.getSelectionModel().getSelectedItem();
            dp.EPropiedad(pr);
            datap(reg);

            Alert a = new
Alert(Alert.AlertType.INFORMATION);

```

```

        a.setHeaderText("Eliminado Correctamente");
        a.setContentText("Propiedad: " +
pr.getDireccion());
        a.showAndWait();
    }
} catch (Exception e) {
    Alert al = new Alert(Alert.AlertType.ERROR);
    al.setHeaderText("Error al Eliminar");
    al.setContentText("Seleccione una fila del Predio");
    al.showAndWait();
    System.out.println(e);
}
}

public boolean EditDialog(Propiedad re) {
    try {
        FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("/view/VCrearPropiedad
.fxml"));

        AnchorPane page = (AnchorPane) loader.load();
        Stage prop = new Stage();
        prop.setTitle("Crear Nuevo Propiedad");
        prop.initModality(Modality.WINDOW_MODAL);
        Scene scene = new Scene(page);
        prop.setScene(scene);
        CCrearPropiedad con = loader.getController();
        con.setDialogStage(prop);
        con.setPerson(re, reg);
        prop.showAndWait();
        return con.isOk();
    } catch (IOException er) {
        System.out.println(er);
        return false;
    }
}

/*modal para agregar predio*/
@FXML
private void crearpredio() {
    try {
        Predio re = new Predio();
        boolean okClicked = EditDialogpre(re);
        if (okClicked) {
            Propiedad pr =
listap.getSelectionModel().getSelectedItem();
            datapredio(pr);
        }
    } catch (NullPointerException er) {
        System.out.println(er);
    }
}

public boolean EditDialogpre(Predio rep) {
    try {
        Propiedad pr =
listap.getSelectionModel().getSelectedItem();
        FXMLLoader loader = new FXMLLoader();

```



```

loader.setLocation(getClass().getResource("/view/VCrearPredio.fxml"));

AnchorPane page = (AnchorPane) loader.load();
Stage pred = new Stage();
pred.setTitle("Crear Nuevo Predio");
pred.initModality(Modality.WINDOW_MODAL);
Scene scene = new Scene(page);
pred.setScene(scene);
CCrearPredio con = loader.getController();
con.setDialogStage(pred);
con.setPerson(pr, cp1.getValue().toString());
pred.showAndWait();
return con.isOk();
} catch (IOException er) {
    System.out.println(er);
    return false;
}
}

@FXML
public void Crear() {
    try {
        Propiedad pr =
listap.getSelectionModel().getSelectedItem();
        DPropiedad d = new DPropiedad();
        if (d.existe(pr, Integer.parseInt(tp11.getText())))
{
            DaoValorpropiedad dp = new DaoValorpropiedad();
            CPredio cp = new CPredio();
            dp.save(new Valorpropiedad(pr,
Integer.parseInt(tp11.getText()), new
BigDecimal(tp21.getText()), new BigDecimal(tp31.getText()),
cp.vter(new BigDecimal(tp21.getText()), new
BigDecimal(tp31.getText()))));
            anos(pr);
            Alert al = new
Alert(Alert.AlertType.INFORMATION);
            al.setHeaderText("Creado Correctamente");
            al.setContentText("Año: " + tp11.getText());
            al.showAndWait();
        } else {
            Alert al = new Alert(Alert.AlertType.ERROR);
            al.setHeaderText("Error al Crear");
            al.setContentText("Este Año ya Existe");
            al.showAndWait();
        }
    } catch (Exception ex) {
        Alert al = new Alert(Alert.AlertType.ERROR);
        al.setHeaderText("Error al Crear");
        al.setContentText("Seleccctione una Propiedad y los
Párametros AÑO, ARANCEL, AREA no deben estar vacios");
        al.showAndWait();
    }
}

@FXML
public void Generar() {
    try {

```

```

        Propiedad pr =
listap.getSelectionModel().getSelectedItem();
        DPropiedad d = new DPropiedad();
        if (!d.existe(pr, Integer.parseInt(tp111.getText()))
&& d.existeC(pr, Integer.parseInt(tp111.getText()))) {
            CCPredio pe = new CCPredio();
            DCaja dc = new DCaja();
            int a = Integer.parseInt(tp111.getText());
            BigDecimal uit = new
BigDecimal(tp211.getText());
            Caja cc = new Caja(pr.getIdprop(), a,
pe.predio(pr, a), pe.alicuota(pe.predio(pr, a), uit), false,
uit, Date.from(Instant.now()));
            dc.save(cc);
            caja(pr);
        } else {
            Alert al = new Alert(Alert.AlertType.ERROR);
            al.setHeaderText("Error al Crear");
            al.setContentText("Este Año no Existe o La
Factura ya fue Generada");
            al.showAndWait();
        }
    } catch (Exception e) {
        Alert al = new Alert(Alert.AlertType.ERROR);
        al.setHeaderText("Error al Crear");
        al.setContentText("AÑO, UIT no deben estar vacios");
        al.showAndWait();
    }
}

public Stage stage;
public Propietario reg;
public boolean isOk = false;

void setDialogStage(Stage dialogStage) {
    stage = dialogStage;
}

void setPerson(Propietario cont) {
    reg = cont;
    pnom.setText(cont.getNombre());
    pnum.setText(cont.getNumdoc());
    pcel.setText(cont.getCelular());
    datap(cont);
}

public boolean isOk() {
    return isOk;
}

@FXML
public void cancelar() {
    stage.close();
}
}

```

class CabrirtcR

```
/*
 * To change this license header, choose License Headers in
 * Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package control;

import dao.DCaja;
import dao.DPropiedad;
import dao.DaoPredio;
import dao.DaoValorpropiedad;
import java.io.IOException;
import java.math.BigDecimal;
import java.net.URL;
import java.time.Instant;
import java.util.Date;
import java.util.Iterator;
import java.util.List;
import java.util.ResourceBundle;
import javafx.collections.FXCollections;
import javafx.collections.ObservableList;
import javafx.fxml.FXML;
import javafx.fxml.FXMLLoader;
import javafx.fxml.Initializable;
import javafx.scene.Scene;
import javafx.scene.control.Alert;
import javafx.scene.control.ComboBox;
import javafx.scene.control.Label;
import javafx.scene.control.TableColumn;
import javafx.scene.control.TableView;
import javafx.scene.control.TextField;
import javafx.scene.layout.AnchorPane;
import javafx.stage.Modality;
import javafx.stage.Stage;
import modelo.Caja;
import modelo.Predio;
import modelo.Propiedad;
import modelo.Propietario;
import modelo.Valorpropiedad;
import mysql.DaoPredioM;

/**
 *
 * @author YONNY
 */
public class CabrirtcR implements Initializable {

    @FXML
    public Label pnomo;
    @FXML
    public Label pnomo;
    @FXML
    public Label pcel;
    @FXML
    private TableView<Propiedad> listap;
    @FXML
```

```

private TableColumn<Propiedad, String> c1;
@FXML
private TableColumn<Propiedad, String> c2;
@FXML
private TableColumn<Propiedad, String> c3;

public ObservableList<Propiedad> ls =
FXCollections.observableArrayList();

/**
 * tabla dos*
 */
@FXML
TextField tp1, tp2, tp3;
@FXML
TextField tp11, tp21, tp31;
@FXML
TextField tp111, tp211;
@FXML
ComboBox cp1;
@FXML
private TableView<Predio> listapredio;
@FXML
private TableColumn<Predio, String> pca;
@FXML
private TableColumn<Predio, String> c;
@FXML
private TableColumn<Predio, String> m;
@FXML
private TableColumn<Predio, String> e;
@FXML
private TableColumn<Predio, String> pc1;
@FXML
private TableColumn<Predio, String> pc2;
@FXML
private TableColumn<Predio, String> pc3;
@FXML
private TableColumn<Predio, String> pc4;
@FXML
private TableColumn<Predio, String> pc5;
@FXML
private TableColumn<Predio, String> pc6;
@FXML
private TableColumn<Predio, String> pc7;
@FXML
private TableColumn<Predio, String> pcv1;//valor de
construccion
@FXML
private TableColumn<Predio, String> pcv2;//area de
contruccion
@FXML
private TableColumn<Predio, String> pcv3;//deprec

public ObservableList<Predio> lss =
FXCollections.observableArrayList();

/**
 * tabla dos*
 */

```

```

@FXML
private TableView<Caja> lista3;
@FXML
private TableColumn<Caja, String> a1;
@FXML
private TableColumn<Caja, String> a2;
@FXML
private TableColumn<Caja, String> a3;
@FXML
private TableColumn<Caja, String> a4;

public ObservableList<Caja> ls3 =
FXCollections.observableArrayList();

@Override
public void initialize(URL location, ResourceBundle
resources) {

}

public void datap(Propietario pr) {
    ls.clear();
    try {

listap.getSelectionModel().selectedItemProperty().addListener((o
bservable, oldValue, po) -> load(po));

        DPropiedad reg1 = new DPropiedad();
        List l = reg1.all(pr);

        for (Iterator iterator = l.iterator();
iterator.hasNext();) {
            Propiedad next = (Propiedad) iterator.next();
            ls.add(next);
        }
        c1.setCellValueFactory(da ->
da.getValue().direccion());
        c2.setCellValueFactory(da ->
da.getValue().distrito());
        c3.setCellValueFactory(da ->
da.getValue().provincia());
        listap.setItems(ls);
    } catch (NullPointerException er) {

    }

}

public void load(Propiedad po) {
    anos(po);
    caja(po);
    datapredio(po);
}

public void anos(Propiedad po) {
    cp1.getItems().clear();
    try {
        DPropiedad dp = new DPropiedad();
        dp.anos(po).stream().map((h) -> {
            cp1.getItems().addAll(h.getAno());
        });
    }
}

```

```

        return h;
    }).forEachOrdered((_item) -> {
        cp1.getSelectionModel().select(0);
    });
} catch (NullPointerException er) {
}
}

@FXML
public void Predios() {
    Propiedad pr =
listap.getSelectionModel().getSelectedItem();
    datapredio(pr);
}

public void datapredio(Propiedad pr) {
    lss.clear();
    try {
        DPropiedad dp = new DPropiedad();

        List<Valorpropiedad> p =
dp.valor(Integer.parseInt(cp1.getValue().toString()), pr);
        if (!p.isEmpty()) {
            tp1.setText(p.get(0).getArancel().toString());
            tp2.setText(p.get(0).getAreat().toString());
            tp3.setText(p.get(0).getValort().toString());
        } else {
            tp1.setText("");
            tp2.setText("");
            tp3.setText("");
        }
    }

    List l =
dp.predios(Integer.parseInt(cp1.getValue().toString()), pr);
    for (Iterator iterator = l.iterator();
iterator.hasNext();) {
        Predio next = (Predio) iterator.next();
        lss.add(next);
    }
    listapredio.setItems(lss);
    pca.setCellValueFactory(da -> da.getValue().ano());
    c.setCellValueFactory(da ->
da.getValue().Clasificacion());
    m.setCellValueFactory(da ->
da.getValue().Material());
    e.setCellValueFactory(da -> da.getValue().Estado());
    pc1.setCellValueFactory(da -> da.getValue().muro());
    pc2.setCellValueFactory(da ->
da.getValue().techo());
    pc3.setCellValueFactory(da ->
da.getValue().pisos());
    pc4.setCellValueFactory(da ->
da.getValue().puerta());
    pc5.setCellValueFactory(da ->
da.getValue().revestimiento());
    pc6.setCellValueFactory(da -> da.getValue().bano());
    pc7.setCellValueFactory(da ->
da.getValue().instalacion());
}
}

```

```

        pcv1.setCellValueFactory(da ->
da.getValue().vconstruccion());
        pcv2.setCellValueFactory(da ->
da.getValue().aconstruccion());
        pcv3.setCellValueFactory(da ->
da.getValue().dconstruccion());
    } catch (NullPointerException er) {
        tp1.setText("");
        tp2.setText("");
        tp3.setText("");
    }
}

public void caja(Propiedad p) {
    ls3.clear();
    try {
        DCaja reg = new DCaja();
        List l = reg.all(p.getIdprop());
        for (Iterator iterator = l.iterator();
iterator.hasNext();) {
            Caja next = (Caja) iterator.next();
            ls3.add(next);
        }
        a1.setCellValueFactory(da -> da.getValue().ano());
        a2.setCellValueFactory(da -> da.getValue().base());
        a3.setCellValueFactory(da -> da.getValue().monto());
        a4.setCellValueFactory(da ->
da.getValue().estado());
        lista3.setItems(ls3);
    } catch (NullPointerException er) {
        System.out.println(er);
    }
}

//modal de crear prOPiEDAD
@FXML
private void crearp() {
    Propiedad re = new Propiedad();
    boolean okClicked = EditDialog(re);
    if (okClicked) {
        datap(reg);
    }
}

@FXML
private void MPro() {

    try {
        Propiedad pr =
listap.getSelectionModel().getSelectedItem();
        boolean okClicked = EditDialog(pr);
        if (okClicked) {
            datap(reg);
        }
    } catch (Exception e) {
        Alert al = new Alert(Alert.AlertType.ERROR);
        al.setHeaderText("Error al Modificar");
        al.setContentText("Seleccione una fila");
        al.showAndWait();
    }
}

```

```

        System.out.println(e);
    }
}

@FXML
private void EAno() {

    try {
        int a = Integer.parseInt(cp1.getValue().toString());
        Propiedad pr =
listap.getSelectionModel().getSelectedItem();
        DPropiedad dp = new DPropiedad();
        dp.EAno(pr, a);
        anos(pr);

        Alert al = new Alert(Alert.AlertType.INFORMATION);
        al.setHeaderText("Eliminado Correctamente");
        al.setContentText("Año: " + String.valueOf(a));
        al.showAndWait();

    } catch (Exception e) {
        Alert al = new Alert(Alert.AlertType.ERROR);
        al.setHeaderText("Error al Eliminar");
        al.setContentText("Seleccione un Año");
        al.showAndWait();
    }
}

@FXML
private void DPro() {
    try {
        Alert al = new Alert(Alert.AlertType.CONFIRMATION);
        al.setHeaderText("Eliminar Pago");
        al.setContentText("¿Está seguro al Eliminar Pago?");
        al.showAndWait();
        if
(al.getResult().getButtonData().isDefaultButton()) {
            int id =
lista3.getSelectionModel().getSelectedIndex();
            DCaja dp = new DCaja();
            Caja pr =
lista3.getSelectionModel().getSelectedItem();
            if (!pr.isEstado()) {
                dp.eliminar(pr);
                lista3.getItems().remove(id);
            } else {
                Alert la = new Alert(Alert.AlertType.ERROR);
                la.setHeaderText("Error al Eliminar");
                la.setContentText("El Predio ya fue
Pagado");
                la.showAndWait();
            }
        }
    } catch (Exception e) {
        Alert al = new Alert(Alert.AlertType.ERROR);
        al.setHeaderText("Error al Eliminar");
        al.setContentText("Seleccione una fila o El Predio
ya fue Pagado");
    }
}

```



```

        al.showAndWait();
        System.out.println(e);
    }

}

@FXML
private void EPredio() {
    try {
        Alert al = new Alert(Alert.AlertType.CONFIRMATION);
        al.setHeaderText("Eliminar Predio");
        al.setContentText("¿Está seguro al Eliminar
Predio?");
        al.showAndWait();
        if
(al.getResult().getButtonData().isDefaultButton()) {
            DaoPredioM dp = new DaoPredioM();
            Predio pr =
listapredio.getSelectionModel().getSelectedItem();
            dp.EPredio(pr);
            Predios();

            Alert a = new
Alert(Alert.AlertType.INFORMATION);
            a.setHeaderText("Eliminado Correctamente");
            a.setContentText("Año: " +
String.valueOf(pr.getAno()));
            a.showAndWait();
        }
    } catch (Exception e) {
        Alert al = new Alert(Alert.AlertType.ERROR);
        al.setHeaderText("Error al Eliminar");
        al.setContentText("Seleccione una fila del Predio");
        al.showAndWait();
        System.out.println(e);
    }
}

@FXML
private void EPropiedad() {
    try {
        Alert al = new Alert(Alert.AlertType.CONFIRMATION);
        al.setHeaderText("Eliminar Propiedad");
        al.setContentText("¿Está seguro al Eliminar
Propiedad?");
        al.showAndWait();
        if
(al.getResult().getButtonData().isDefaultButton()) {
            DaoPredioM dp = new DaoPredioM();
            Propiedad pr =
listap.getModel().getSelectedItem();
            dp.EPropiedad(pr);
            datap(reg);

            Alert a = new
Alert(Alert.AlertType.INFORMATION);
            a.setHeaderText("Eliminado Correctamente");
            a.setContentText("Propiedad: " +
pr.getDireccion());

```

```

        a.showAndWait();
    }
} catch (Exception e) {
    Alert al = new Alert(Alert.AlertType.ERROR);
    al.setHeaderText("Error al Eliminar");
    al.setContentText("Seleccione una fila del Predio");
    al.showAndWait();
    System.out.println(e);
}
}

public boolean EditDialog(Propiedad re) {
    try {
        FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("/view/VCrearPropiedad
R.fxml"));
        AnchorPane page = (AnchorPane) loader.load();
        Stage prop = new Stage();
        prop.setTitle("Crear Nuevo Propiedad - Rustico");
        prop.initModality(Modality.WINDOW_MODAL);
        Scene scene = new Scene(page);
        prop.setScene(scene);
        CCrearPropiedadR con = loader.getController();
        con.setDialogStage(prop);
        con.setPerson(re, reg);
        prop.showAndWait();
        return con.isOk();
    } catch (IOException er) {
        System.out.println(er);
        return false;
    }
}

/*modal para agregar predio*/
@FXML
private void crearpredio() {
    try {
        Predio re = new Predio();
        boolean okClicked = EditDialogpre(re);
        if (okClicked) {
            Propiedad pr =
listap.getSelectionModel().getSelectedItem();
            datapredio(pr);
        }
    } catch (NullPointerException er) {
        System.out.println(er);
    }
}

public boolean EditDialogpre(Predio rep) {
    try {
        Propiedad pr =
listap.getSelectionModel().getSelectedItem();
        FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("/view/VCrearPredioR.f
xml"));
        AnchorPane page = (AnchorPane) loader.load();

```

```

        Stage pred = new Stage();
        pred.setTitle("Crear Nuevo Predio - Rustico");
        pred.initModality(Modality.WINDOW_MODAL);
        Scene scene = new Scene(page);
        pred.setScene(scene);
        CCrearPredioR con = loader.getController();
        con.setDialogStage(pred);
        con.setPerson(pr, cp1.getValue().toString());
        pred.showAndWait();
        return con.isOk();
    } catch (IOException er) {
        System.out.println(er);
        return false;
    }
}

@FXML
public void Crear() {
    try {
        Propiedad pr =
listap.getSelectionModel().getSelectedItem();
        DPropiedad d = new DPropiedad();
        if (d.existe(pr, Integer.parseInt(tp11.getText())))
{
            DaoValorpropiedad dp = new DaoValorpropiedad();
            CPredio cp = new CPredio();
            dp.save(new Valorpropiedad(pr,
Integer.parseInt(tp11.getText()), new
BigDecimal(tp21.getText()), new BigDecimal(tp31.getText()),
cp.vter(new BigDecimal(tp21.getText()), new
BigDecimal(tp31.getText()))));
            anos(pr);
            Alert al = new
Alert(Alert.AlertType.INFORMATION);
            al.setHeaderText("Creado Correctamente");
            al.setContentText("Año: " + tp11.getText());
            al.showAndWait();
        } else {
            Alert al = new Alert(Alert.AlertType.ERROR);
            al.setHeaderText("Error al Crear");
            al.setContentText("Este Año ya Existe");
            al.showAndWait();
        }
    } catch (Exception ex) {
        Alert al = new Alert(Alert.AlertType.ERROR);
        al.setHeaderText("Error al Crear");
        al.setContentText("Seleccione una Propiedad y los
Parámetros AÑO, ARANCEL, AREA no deben estar vacios");
        al.showAndWait();
    }
}

@FXML
public void Generar() {
    try {
        Propiedad pr =
listap.getSelectionModel().getSelectedItem();
        DPropiedad d = new DPropiedad();

```

```

        if (!d.existe(pr, Integer.parseInt(tp111.getText()))
&& d.existeC(pr, Integer.parseInt(tp111.getText()))) {
            CCPredio pe = new CCPredio();
            DCaja dc = new DCaja();
            int a = Integer.parseInt(tp111.getText());
            BigDecimal uit = new
BigDecimal(tp211.getText());
            Caja cc = new Caja(pr.getIdprop(), a,
pe.predio(pr, a), pe.alicuota(pe.predio(pr, a), uit), false,
uit, Date.from(Instant.now()));
            dc.save(cc);
            caja(pr);
        } else {
            Alert al = new Alert(Alert.AlertType.ERROR);
            al.setHeaderText("Error al Crear");
            al.setContentText("Este Año no Existe o La
Factura ya fue Generada");
            al.showAndWait();
        }
    } catch (Exception e) {
        Alert al = new Alert(Alert.AlertType.ERROR);
        al.setHeaderText("Error al Crear");
        al.setContentText("AÑO, UIT no deben estar vacios");
        al.showAndWait();
    }
}

public Stage stage;
public Propietario reg;
public boolean isOk = false;

void setDialogStage(Stage dialogStage) {
    stage = dialogStage;
}

void setPerson(Propietario cont) {
    reg = cont;
    pnom.setText(cont.getNombre());
    pnum.setText(cont.getNumdoc());
    pcel.setText(cont.getCelular());
    datap(cont);
}

public boolean isOk() {
    return isOk;
}

@FXML
public void cancelar() {
    stage.close();
}

/*modal para agregar arancel*/

@FXML
private void creararancel() {
    try {
        Predio re = new Predio();

```

```

        boolean okClicked = EditDialogara(re);
        if (okClicked) {
            Propiedad pr =
listap.getSelectionModel().getSelectedItem();
            datapredio(pr);
        }
    } catch (NullPointerException er) {
        System.out.println(er);
    }
}

public boolean EditDialogara(Predio rep) {
    try {
        Propiedad pr =
listap.getSelectionModel().getSelectedItem();
        FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("/view/VCrearArancelR.
FXML"));

        AnchorPane page = (AnchorPane) loader.load();
        Stage pred = new Stage();
        pred.setTitle("Crear Nuevo Arancel");
        pred.initModality(Modality.WINDOW_MODAL);
        Scene scene = new Scene(page);
        pred.setScene(scene);
        CCrearArancelR con = loader.getController();
        con.setDialogStage(pred);
        con.setPerson(pr, cp1.getValue().toString());
        pred.showAndWait();
        return con.isOk();
    } catch (IOException er) {
        System.out.println(er);
        return false;
    }
}
}
}

```

class Cayuda

```

/*
 * To change this license header, choose License Headers in
Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package control;

import java.net.URL;
import java.util.ResourceBundle;
import javafx.fxml.Initializable;

/**
 *
 * @author YONNY
 */
public class Cayuda implements Initializable{

```

```

        public void initialize(URL location, ResourceBundle
resources) {

        }
}

```

class CCaja

```

/*
 * To change this license header, choose License Headers in
Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package control;

import dao.DCaja;
import dao.DPropiedad;
import dao.DaoPropietario;
import java.io.IOException;
import java.net.URL;
import java.util.ArrayList;
import java.util.Iterator;
import java.util.List;
import java.util.ResourceBundle;
import javafx.collections.FXCollections;
import javafx.collections.ObservableList;
import javafx.fxml.FXML;
import javafx.fxml.Initializable;
import javafx.scene.control.Alert;
import javafx.scene.control.TableColumn;
import javafx.scene.control.TableView;
import javafx.scene.control.TextField;
import javax.swing.JOptionPane;
import modelo.Caja;
import modelo.Propietario;
import modelo.Propiedad;

/**
 * FXML Controller class
 *
 * @author Manrrique
 */
public class CCaja implements Initializable {

    @FXML
    public TextField n;
    @FXML
    public TextField d;

    @FXML
    private TableView<Propietario> lista;
    @FXML
    private TableColumn<Propietario, String> c1;
    @FXML
    private TableColumn<Propietario, String> c2;

```

```

    public ObservableList<Propietario> ls =
FXCollections.observableArrayList();

    @FXML
    private TableView<Propiedad> lista2;
    @FXML
    private TableColumn<Propiedad, String> p1;
    @FXML
    private TableColumn<Propiedad, String> p2;
    public ObservableList<Propiedad> ls2 =
FXCollections.observableArrayList();

    @FXML
    private TableView<Caja> lista3;
    @FXML
    private TableColumn<Caja, String> a1;
    @FXML
    private TableColumn<Caja, String> a2;
    @FXML
    private TableColumn<Caja, String> a3;
    @FXML
    private TableColumn<Caja, String> a4;
    public ObservableList<Caja> ls3 =
FXCollections.observableArrayList();

    @Override
    public void initialize(URL url, ResourceBundle rb) {
        // TODO
    }

    @FXML
    public void Buscar() {
        data(n.getText(), d.getText());
    }

    @FXML
    public void Pagar() {
        Alert a = new Alert(Alert.AlertType.CONFIRMATION);
        a.setHeaderText("Pagar Predio");
        a.setContentText("Una vez pagado no se podrá
modificar");
        a.showAndWait();
        if (a.getResult().getButtonData().isDefaultButton()) {
            pagado();
        }
    }

    public void data(String nom, String ndoc) {
        ls.clear();
        try {

lista.getSelectionModel().selectedItemProperty().addListener((ob
servable, oldValue, p) -> prop(p));

            DaoPropietario reg = new DaoPropietario();
            List l = new ArrayList();
            if (!ndoc.equals("")) {
                l = (List) reg.data(ndoc);
            } else if (!nom.equals("")) {

```

```

        l = (List) reg.nombre(nom);
    } else if (ndoc.equals("") && nom.equals("")) {
        l = reg.all();
    }
    if (l.isEmpty()) {
        JOptionPane.showMessageDialog(null, "No Hay
Resultados");
    }

    for (Iterator iterator = l.iterator();
iterator.hasNext();) {
        Propietario next = (Propietario)
iterator.next();
        ls.add(next);
    }
    c1.setCellValueFactory(da -> da.getValue().nom());
    c2.setCellValueFactory(da ->
da.getValue().numdoc());
    lista.setItems(ls);
    } catch (NullPointerException e) {

    }
}

public void prop(Propietario p) {
    ls2.clear();
    try {

lista2.getSelectionModel().selectedItemProperty().addListener((o
bservable, oldValue, po) -> caja(po));

        DPropiedad reg = new DPropiedad();
        List l = reg.all(p);

        for (Iterator iterator = l.iterator();
iterator.hasNext();) {
            Propiedad next = (Propiedad) iterator.next();
            ls2.add(next);
        }
        p1.setCellValueFactory(da ->
da.getValue().direccion());
        p2.setCellValueFactory(da -> da.getValue().dis());
        lista2.setItems(ls2);
    } catch (NullPointerException e) {

    }
}

public void caja(Propiedad p) {
    ls3.clear();
    try {
        DCaja reg = new DCaja();
        List l = reg.all(p.getIdprop());

        for (Iterator iterator = l.iterator();
iterator.hasNext();) {
            Caja next = (Caja) iterator.next();
            ls3.add(next);
        }
    }
}

```



```

        a1.setCellValueFactory(da -> da.getValue().ano());
        a2.setCellValueFactory(da -> da.getValue().base());
        a3.setCellValueFactory(da -> da.getValue().monto());
        a4.setCellValueFactory(da ->
da.getValue().estado());
        lista3.setItems(ls3);
    } catch (NullPointerException e) {

    }

}

public void pagado() {
    try {
        DCaja dc = new DCaja();
        Propietario pr =
lista.getSelectionModel().getSelectedItem();
        Propiedad pd =
lista2.getSelectionModel().getSelectedItem();
        Caja c =
lista3.getSelectionModel().getSelectedItem();
        c.setEstado(true);
        dc.save(c);

        //IPU ip = new IPU();
        IDJ ip2 = new IDJ();
        DPropiedad dp = new DPropiedad();
        //int i = ip.IP(c, pr, pd);
        int i = ip2.IP(dp.predios(c.getAno(), pd), pr, pd,
dp.valor(c.getAno(), pd).get(0), c);
        if (i == 1) {
            Alert a = new
Alert(Alert.AlertType.INFORMATION);
            a.setHeaderText("Predio Pagado");
            a.setContentText("PDF Generado Correctamente");
            a.showAndWait();
        }
        caja(pd);
        try {
            Runtime.getRuntime().exec("rundll32
url.dll,FileProtocolHandler " + "D://Autovaluo/predio.pdf");
        } catch (IOException e) {
            System.out.println(e);
            Alert a = new Alert(Alert.AlertType.ERROR);
            a.setHeaderText("PDF Cagado");
            a.setContentText("Su Puto PDF esta abierto o no
existe la ruta");
            a.showAndWait();
        }
    } catch (Exception e) {
        System.out.println(e);
        Alert a = new Alert(Alert.AlertType.ERROR);
        a.setHeaderText("PDF Cagado");
        a.setContentText("Su Puto PDF esta abierto o no
existe la ruta");
        a.showAndWait();
    }
}
}
}

```

class CContribuyente

```
/*
 * To change this license header, choose License Headers in
 Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package control;

import dao.DaoPropietario;
import java.io.IOException;
import java.net.URL;
import java.util.ArrayList;
import java.util.Iterator;
import java.util.List;
import java.util.ResourceBundle;
import javafx.collections.FXCollections;
import javafx.collections.ObservableList;
import javafx.fxml.FXML;
import javafx.fxml.FXMLLoader;
import javafx.fxml.Initializable;
import javafx.scene.Scene;
import javafx.scene.control.Alert;
import javafx.scene.control.TableColumn;
import javafx.scene.control.TableView;
import javafx.scene.control.TextField;
import javafx.scene.layout.AnchorPane;
import javafx.stage.Modality;
import javafx.stage.Stage;
import javax.swing.JOptionPane;
import modelo.Propietario;

/**
 *
 * @author YONNY
 */
public class CContribuyente implements Initializable {

    @FXML
    TextField CTFnombre, CTFnumdoc;

    public static Propietario propiet;

    @FXML
    private TableView<Propietario> CTVtablacont;

    @FXML
    private TableColumn<Propietario, String> TCnum;

    @FXML
    private TableColumn<Propietario, String> TCnomap;

    @FXML
    private TableColumn<Propietario, String> TCtipodoc;

    public ObservableList<Propietario> ls =
FXCollections.observableArrayList();
```

```

@Override
public void initialize(URL location, ResourceBundle
resources) {
    CData(CTFnombre.getText(), CTFnumdoc.getText());
}

@FXML
public void Buscar() throws IOException {
    CData(CTFnombre.getText(), CTFnumdoc.getText());
}

@FXML
public void Abrir() throws IOException {
    Propietario na =
CTVtablacont.getSelectionModel().getSelectedItem();
    if (na != null) {
        boolean okClicked = modal(na);
        if (okClicked) {
            CData(CTFnombre.getText(), CTFnumdoc.getText());
        }
    } else {
        JOptionPane.showMessageDialog(null, "Seleccione un
Propietario");
    }
}

/**
 * metodo mostrar en tabla
 *
 *
 * @param nom
 * @param ndoc
 */
public void CData(String nom, String ndoc) {
    System.out.println(ndoc);
    ls.clear();
    DaoPropietario reg1 = new DaoPropietario();

    List l = new ArrayList();
    if (!ndoc.equals("")) {
        l = (List) reg1.data(ndoc);
    } else if (!nom.equals("")) {
        l = (List) reg1.nombre(nom);
    } else if (ndoc.equals("") && ndoc.equals("")) {
        l = reg1.all();
    }
    if (l.isEmpty()) {
        JOptionPane.showMessageDialog(null, "No Hay
Resultados");
    }

    for (Iterator iterator = l.iterator();
iterator.hasNext();) {
        Propietario next = (Propietario) iterator.next();
        ls.add(next);
    }
    CTVtablacont.setItems(ls);
    TCnum.setCellValueFactory(da -> da.getValue().numdoc());
    TCnomap.setCellValueFactory(da -> da.getValue().nom());
}

```

```

        TCTipodoc.setCellValueFactory(da ->
da.getValue().tipdoc());
    }

    public boolean modal(modelo.Propietario pro) {
        try {
            FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("/view/VMCabrir.fxml")
);

            AnchorPane page = (AnchorPane) loader.load();
            Stage stage = new Stage();
            stage.setTitle("Abrir Person");
            stage.initModality(Modality.WINDOW_MODAL);
            Scene scene = new Scene(page);
            stage.setScene(scene);
            Cabrirtc con = loader.getController();
            con.setDialogStage(stage);
            con.setPerson(pro);
            stage.showAndWait();
            return con.isOk();
        } catch (Exception e) {
            System.out.println(e);
            return false;
        }

    }

    @FXML
    private void Delete() {
        try {
            Alert al = new Alert(Alert.AlertType.CONFIRMATION);
            al.setHeaderText("Eliminar Contribuyente");
            al.setContentText("¿Está seguro al Eliminar
Contribuyente?");
            al.showAndWait();
            if
(al.getResult().getButtonData().isDefaultButton()) {
                int id =
CTVtablacont.getSelectionModel().getSelectedIndex();
                DaoPropietario reg = new DaoPropietario();
                Propietario us =
CTVtablacont.getSelectionModel().getSelectedItem();
                reg.eliminar(us);
                CTVtablacont.getItems().remove(id);
            }
        } catch (Exception e) {
            Alert al = new Alert(Alert.AlertType.ERROR);
            al.setHeaderText("Error al Eliminar");
            al.setContentText("Seleccione una fila o El
propietario Tiene Propiedades");
            al.showAndWait();
            System.out.println(e);
        }

    }

    @FXML
    private void New() {
        Propietario re = new Propietario();
    }

```

```

        boolean okClicked = EditDialog(re);
        if (okClicked) {
            CData(CTFnombre.getText(), CTFnumdoc.getText());
        }
    }

    public boolean EditDialog(Propietario re) {
        try {
            FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("/view/VMCagregar.fxml
"));

            AnchorPane page = (AnchorPane) loader.load();
            Stage stage = new Stage();
            stage.setTitle("Edit Person");
            stage.initModality(Modality.WINDOW_MODAL);
            Scene scene = new Scene(page);
            stage.setScene(scene);
            CMCagregar con = loader.getController();
            con.setDialogStage(stage);
            con.setPerson(re);
            stage.showAndWait();
            return con.isOk();
        } catch (Exception e) {
            System.out.println(e);
            return false;
        }
    }

    @FXML
    private void Edit() {
        Propietario id =
CTVtablacont.getSelectionModel().getSelectedItem();
        if (id != null) {
            boolean okClicked = EditDialog(id);
            if (okClicked) {
                CData(CTFnombre.getText(), CTFnumdoc.getText());
            }

        } else {
            Alert al = new Alert(Alert.AlertType.ERROR);
            al.setHeaderText("Error al Modificar");
            al.setContentText("Seleccione una fila");
            al.showAndWait();
        }
    }
}
}

```

class CContribuyenteR

```

/*
 * To change this license header, choose License Headers in
Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package control;

```

```

import dao.DaoPropietario;
import java.io.IOException;
import java.net.URL;
import java.util.ArrayList;
import java.util.Iterator;
import java.util.List;
import java.util.ResourceBundle;
import javafx.collections.FXCollections;
import javafx.collections.ObservableList;
import javafx.fxml.FXML;
import javafx.fxml.FXMLLoader;
import javafx.fxml.Initializable;
import javafx.scene.Scene;
import javafx.scene.control.Alert;
import javafx.scene.control.TableColumn;
import javafx.scene.control.TableView;
import javafx.scene.control.TextField;
import javafx.scene.layout.AnchorPane;
import javafx.stage.Modality;
import javafx.stage.Stage;
import javax.swing.JOptionPane;
import modelo.Propietario;

/**
 *
 * @author YONNY
 */
public class CContribuyenteR implements Initializable {

    @FXML
    TextField CTFnombre, CTFnumdoc;

    public static Propietario propiet;

    @FXML
    private TableView<Propietario> CTVtablacont;

    @FXML
    private TableColumn<Propietario, String> TCnum;

    @FXML
    private TableColumn<Propietario, String> TCnomap;

    @FXML
    private TableColumn<Propietario, String> TCtipodoc;

    public ObservableList<Propietario> ls =
FXCollections.observableArrayList();

    @Override
    public void initialize(URL location, ResourceBundle
resources) {
        CData (CTFnombre.getText (), CTFnumdoc.getText ());
    }

    @FXML
    public void Buscar() throws IOException {
        CData (CTFnombre.getText (), CTFnumdoc.getText ());
    }

```

```

}

@FXML
public void Abrir() throws IOException {
    Propietario na =
CTVtablacont.getSelectionModel().getSelectedItem();
    if (na != null) {
        boolean okClicked = modal(na);
        if (okClicked) {
            CData(CTFnombre.getText(), CTFnumdoc.getText());
        }
    } else {
        JOptionPane.showMessageDialog(null, "Seleccione un
Propietario");
    }
}

/**
 * metodo mostrar en tabla
 *
 *
 * @param nom
 * @param ndoc
 */
public void CData(String nom, String ndoc) {
    System.out.println(ndoc);
    ls.clear();
    DaoPropietario reg1 = new DaoPropietario();

    List l = new ArrayList();
    if (!ndoc.equals("")) {
        l = (List) reg1.data(ndoc);
    } else if (!nom.equals("")) {
        l = (List) reg1.nombre(nom);
    } else if (ndoc.equals("") && ndoc.equals("")) {
        l = reg1.all();
    }
    if (l.isEmpty()) {
        JOptionPane.showMessageDialog(null, "No Hay
Resultados");
    }

    for (Iterator iterator = l.iterator();
iterator.hasNext();) {
        Propietario next = (Propietario) iterator.next();
        ls.add(next);
    }
    CTVtablacont.setItems(ls);
    TCnum.setCellValueFactory(da -> da.getValue().numdoc());
    TCnomap.setCellValueFactory(da -> da.getValue().nom());
    TCTipodoc.setCellValueFactory(da ->
da.getValue().tipdoc());
}

public boolean modal(modelo.Propietario pro) {
    try {
        FXMLLoader loader = new FXMLLoader();

```

```

loader.setLocation(getClass().getResource("/view/VMCabrirR.fxml"
));

    AnchorPane page = (AnchorPane) loader.load();
    Stage stage = new Stage();
    stage.setTitle("ABRIR PERSONA - RUSTICO");
    stage.initModality(Modality.WINDOW_MODAL);
    Scene scene = new Scene(page);
    stage.setScene(scene);
    CabrirtcR con = loader.getController();
    con.setDialogStage(stage);
    con.setPerson(pro);
    stage.showAndWait();
    return con.isOk();
} catch (Exception e) {
    System.out.println(e);
    return false;
}

}

@FXML
private void Delete() {
    try {
        Alert al = new Alert(Alert.AlertType.CONFIRMATION);
        al.setHeaderText("Eliminar Contribuyente");
        al.setContentText("¿Está seguro al Eliminar
Contribuyente?");
        al.showAndWait();
        if
(al.getResult().getButtonData().isDefaultButton()) {
            int id =
CTVtablacont.getSelectionModel().getSelectedIndex();
            DaoPropietario reg = new DaoPropietario();
            Propietario us =
CTVtablacont.getSelectionModel().getSelectedItem();
            reg.eliminar(us);
            CTVtablacont.getItems().remove(id);
        }
    } catch (Exception e) {
        Alert al = new Alert(Alert.AlertType.ERROR);
        al.setHeaderText("Error al Eliminar");
        al.setContentText("Seleccione una fila o El
propietario Tiene Propiedades");
        al.showAndWait();
        System.out.println(e);
    }
}

@FXML
private void New() {
    Propietario re = new Propietario();
    boolean okClicked = EditDialog(re);
    if (okClicked) {
        CData(CTFnombre.getText(), CTFnumdoc.getText());
    }
}

public boolean EditDialog(Propietario re) {

```



```

        try {
            FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("/view/VMCagregarR.fxml
1"));

            AnchorPane page = (AnchorPane) loader.load();
            Stage stage = new Stage();
            stage.setTitle("Edit Person");
            stage.initModality(Modality.WINDOW_MODAL);
            Scene scene = new Scene(page);
            stage.setScene(scene);
            CMCagregarR con = loader.getController();
            con.setDialogStage(stage);
            con.setPerson(re);
            stage.showAndWait();
            return con.isOk();
        } catch (Exception e) {
            System.out.println(e);
            return false;
        }
    }

@FXML
private void Edit() {
    Propietario id =
CTVtablacont.getSelectionModel().getSelectedItem();
    if (id != null) {
        boolean okClicked = EditDialog(id);
        if (okClicked) {
            CData(CTFnombre.getText(), CTFnumdoc.getText());
        }

    } else {
        Alert al = new Alert(Alert.AlertType.ERROR);
        al.setHeaderText("Error al Modificar");
        al.setContentText("Seleccione una fila");
        al.showAndWait();
    }
}
}
}

```

class CCPredio

```

/*
 * To change this license header, choose License Headers in
Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package control;

import dao.DPropiedad;
import java.math.BigDecimal;
import java.math.RoundingMode;
import java.util.List;
import modelo.Predio;
import modelo.Propiedad;

```

```

import modelo.Valorpropiedad;

/**
 *
 * @author Manrrique
 */
public class CCPredio {

    public BigDecimal predio(Propiedad p, int a) {
        BigDecimal m = new BigDecimal(0);
        DPropiedad dp = new DPropiedad();
        Propiedad pr = dp.data(p);
        List<Predio> lp = dp.predios(a, pr);
        List<Valorpropiedad> lv = dp.valor(a, pr);
        for (Predio predio : lp) {
            BigDecimal pu =
predio.getArea().multiply(predio.getValorconst()).multiply(new
BigDecimal(100).subtract(predio.getDeprec()));
            pu = pu.divide(new BigDecimal(100));
            m = m.add(pu);
        }
        for (Valorpropiedad valorpropiedad : lv) {
            m = m.add(valorpropiedad.getValort());
            break;
        }
        return m.setScale(2, RoundingMode.CEILING);
    }

    public BigDecimal alicuota(BigDecimal t, BigDecimal uit) {
        BigDecimal m = new BigDecimal(0);
        if (t.intValue() >= uit.multiply(new
BigDecimal(15)).intValue()) {
            t = t.subtract(uit.multiply(new BigDecimal(15)));
            m = m.add(uit.multiply(new
BigDecimal(15).multiply(new BigDecimal(0.002))));
            if (t.intValue() >= uit.multiply(new
BigDecimal(45)).intValue()) {
                t = t.subtract(uit.multiply(new
BigDecimal(45)));
                m = m.add(uit.multiply(new
BigDecimal(45).multiply(new BigDecimal(0.006))));
                if (t.intValue() > 0) {
                    m = m.add(t.multiply(new BigDecimal(0.01)));
                }
            } else {
                m = m.add(t.multiply(new BigDecimal(0.006)));
            }
        } else {
            m = m.add(t.multiply(new BigDecimal(0.002)));
        }
        return m.setScale(2, RoundingMode.CEILING);
    }
}
}

```

class CCrearArancelR

```

/*

```

```

* To change this license header, choose License Headers in
Project Properties.
* To change this template file, choose Tools | Templates
* and open the template in the editor.
*/
package control;

import dao.DBano;
import dao.DaoConservacion;
import dao.DaoHabitacion;
import dao.DaoInstalaciones;
import dao.DaoMaterialMuro;
import dao.DaoMatpredominante;
import dao.DaoPiso;
import dao.DaoPredio;
import dao.DaoPuerta;
import dao.DaoRevestimiento;
import dao.DaoTecho;
import java.io.IOException;
import java.math.BigDecimal;
import java.net.URL;
import java.util.ResourceBundle;
import javafx.fxml.FXML;
import javafx.fxml.Initializable;
import javafx.scene.control.ComboBox;
import javafx.scene.control.TextField;
import javafx.scene.input.KeyEvent;
import javafx.stage.Stage;
import modelo.Bano;
import modelo.Instalaciones;
import modelo.Materialmuro;
import modelo.Piso;
import modelo.Predio;
import modelo.Propiedad;
import modelo.Puerta;
import modelo.Revestimiento;
import modelo.Techo;

/**
 *
 * @author YONNY
 */
public class CCrearArancelR implements Initializable {

    @FXML
    ComboBox ch1, ch2, ch3, cv1, cv2, cv3, cv4, cv5, cv6, cv7;
    @FXML
    TextField th1, th2, th3, th4;
    @FXML
    TextField tv1, tv2, tv3, tv4, tv5, tv6, tv7;
    @FXML
    TextField th5, th6;

    public Stage stage;
    public Predio reg;
    public boolean isOk = false;
    Propiedad pro;

    public BigDecimal predio;

```

```

public BigDecimal depreciacion;

@Override
public void initialize(URL location, ResourceBundle
resources) {

    DaoHabitacion dh = new DaoHabitacion();
    dh.all().stream().map(h -> {
        ch1.getItems().addAll(h.getDescripcion());
        return h;
    }).forEachOrdered((_item) -> {
        ch1.getSelectionModel().select(0);
    });

    DaoMatpredominante dm = new DaoMatpredominante();
    dm.all().stream().map(h -> {
        ch2.getItems().addAll(h.getDescripcion());
        return h;
    }).forEachOrdered((_item) -> {
        ch2.getSelectionModel().select(0);
    });

    DaoConservacion dc = new DaoConservacion();
    dc.all().stream().map(h -> {
        ch3.getItems().addAll(h.getDescripcion());
        return h;
    }).forEachOrdered((_item) -> {
        ch3.getSelectionModel().select(0);
    });
    /*--*/
    DaoMaterialMuro dmm = new DaoMaterialMuro();
    dmm.all().stream().map(h -> {
        cv1.getItems().addAll(h.getLetra() + " : " +
h.getDescripcion());
        return h;
    }).forEachOrdered((_item) -> {
        cv1.getSelectionModel().select(0);
    });

    DaoTecho dt = new DaoTecho();
    dt.all().stream().map(h -> {
        cv2.getItems().addAll(h.getLetra() + " : " +
h.getDescripcion());
        return h;
    }).forEachOrdered((_item) -> {
        cv2.getSelectionModel().select(0);
    });

    DaoPiso dp = new DaoPiso();
    dp.all().stream().map(h -> {
        cv3.getItems().addAll(h.getLetra() + " : " +
h.getDescripcion());
        return h;
    }).forEachOrdered((_item) -> {
        cv3.getSelectionModel().select(0);
    });

    DaoPuerta dpt = new DaoPuerta();
    dpt.all().stream().map(h -> {

```

```

        cv4.getItems().addAll(h.getLetra() + " : " +
h.getDescripcion());
        return h;
    }).forEachOrdered((_item) -> {
        cv4.getSelectionModel().select(0);
    });

    DaoRevestimiento dr = new DaoRevestimiento();
    dr.all().stream().map((h) -> {
        cv5.getItems().addAll(h.getLetra() + " : " +
h.getDescripcion());
        return h;
    }).forEachOrdered((_item) -> {
        cv5.getSelectionModel().select(0);
    });

    DBano db = new DBano();
    db.all().stream().map((h) -> {
        cv6.getItems().addAll(h.getLetra() + " : " +
h.getDescripcion());
        return h;
    }).forEachOrdered((_item) -> {
        cv6.getSelectionModel().select(0);
    });

    DaoInstalaciones di = new DaoInstalaciones();
    di.all().stream().map((h) -> {
        cv7.getItems().addAll(h.getLetra() + " : " +
h.getDescripcion());
        return h;
    }).forEachOrdered((_item) -> {
        cv7.getSelectionModel().select(0);
    });

    Muro();
    Piso();
    Puerta();
    Revestimiento();
    Techo();
    Bano();
    Electrica();
    Deprecia();
}

public void setPerson(Propiedad pr, String a) {
    th3.setText(a);
    pro = pr;
}

@FXML
public void Enter() {

}

@FXML
public void Save() {
    DaoMaterialMuro dl = new DaoMaterialMuro();
    String r1 = (String) cv1.getValue();
    r1 = r1.substring(0, 1);
}

```

```

Materialmuro i1 = d1.data(r1);

DaoTecho d2 = new DaoTecho();
String r2 = (String) cv2.getValue();
r2 = r2.substring(0, 1);
Techo i2 = d2.data(r2);

DaoPiso d3 = new DaoPiso();
String r3 = (String) cv3.getValue();
r3 = r3.substring(0, 1);
Piso i3 = d3.data(r3);

DaoPuerta d4 = new DaoPuerta();
String r4 = (String) cv4.getValue();
r4 = r4.substring(0, 1);
Puerta i4 = d4.data(r4);

DaoRevestimiento d5 = new DaoRevestimiento();
String r5 = (String) cv5.getValue();
r5 = r5.substring(0, 1);
Revestimiento i5 = d5.data(r5);

DBano d6 = new DBano();
String r6 = (String) cv6.getValue();
r6 = r6.substring(0, 1);
Bano i6 = d6.data(r6);

DaoInstalaciones d7 = new DaoInstalaciones();
String r7 = (String) cv7.getValue();
r7 = r7.substring(0, 1);
Instalaciones i7 = d7.data(r7);

DaoConservacion c1 = new DaoConservacion();
DaoHabitacion c2 = new DaoHabitacion();
DaoMatpredominante c3 = new DaoMatpredominante();
Predio dp = new Predio(i6,

c1.data(ch3.getSelectionModel().getSelectedIndex() + 1),
c2.data(ch1.getSelectionModel().getSelectedIndex() + 1),
    i7, i1,

c3.data(ch2.getSelectionModel().getSelectedIndex() + 1),
    i3, pro, i4, i5, i2,
Integer.parseInt(th3.getText()), th1.getText() + " " +
th2.getText(), th4.getText(),
    predio, new BigDecimal(th5.getText()),
depreciacion);
    DaoPredio c4 = new DaoPredio();
    c4.save(dp);
    isOk = true;
    stage.close();
}

@FXML
public void Salir() {
    stage.close();
}

```

```

void setDialogStage(Stage dialogStage) {
    stage = dialogStage;
}

boolean isOk() {
    return isOk;
}

@FXML
public void Muro() {
    DaoMaterialMuro di = new DaoMaterialMuro();
    String r = (String) cv1.getValue();
    r = r.substring(0, 1);
    Materialmuro i = di.data(r);
    tv1.setText(i.getMonto().toString());
    Sumar();
}

@FXML
public void Techo() {
    DaoTecho di = new DaoTecho();
    String r = (String) cv2.getValue();
    r = r.substring(0, 1);
    Techo i = di.data(r);
    tv2.setText(i.getMonto().toString());
    Sumar();
}

@FXML
public void Piso() {
    DaoPiso di = new DaoPiso();
    String r = (String) cv3.getValue();
    r = r.substring(0, 1);
    Piso i = di.data(r);
    tv3.setText(i.getMonto().toString());
    Sumar();
}

@FXML
public void Puerta() {
    DaoPuerta di = new DaoPuerta();
    String r = (String) cv4.getValue();
    r = r.substring(0, 1);
    Puerta i = di.data(r);
    tv4.setText(String.valueOf(i.getMonto()));
    Sumar();
}

@FXML
public void Revestimiento() {
    DaoRevestimiento di = new DaoRevestimiento();
    String r = (String) cv5.getValue();
    r = r.substring(0, 1);
    Revestimiento i = di.data(r);
    tv5.setText(i.getMonto().toString());
    Sumar();
}

@FXML

```

```

public void Bano() {
    DBano di = new DBano();
    String r = (String) cv6.getValue();
    r = r.substring(0, 1);
    Bano i = di.data(r);
    tv6.setText(i.getMonto().toString());
    Sumar();
}

@FXML
public void Electrica() {
    DaoInstalaciones di = new DaoInstalaciones();
    String r = (String) cv7.getValue();
    r = r.substring(0, 1);
    Instalaciones i = di.data(r);
    tv7.setText(i.getMonto().toString());
    Sumar();
}

@FXML
public void Deprecia() {
    CPredio cp = new CPredio();
    depreciacion =
cp.vdep(ch1.getSelectionModel().getSelectedIndex() + 1,
        antiguedad(th4.getText()),
        ch2.getSelectionModel().getSelectedIndex() + 1,
        ch3.getSelectionModel().getSelectedIndex() + 1);
    System.out.println("Deprecia : " +
depreciacion.toString());
}

public int antiguedad(String an) {
    int a = 5;
    if (Integer.parseInt(an) <= 5) {
        a = 5;
    } else if (Integer.parseInt(an) <= 10) {
        a = 10;
    } else if (Integer.parseInt(an) <= 15) {
        a = 15;
    } else if (Integer.parseInt(an) <= 20) {
        a = 20;
    } else if (Integer.parseInt(an) <= 25) {
        a = 25;
    } else if (Integer.parseInt(an) <= 30) {
        a = 30;
    } else if (Integer.parseInt(an) <= 35) {
        a = 35;
    } else if (Integer.parseInt(an) <= 40) {
        a = 40;
    } else if (Integer.parseInt(an) <= 45) {
        a = 45;
    } else if (Integer.parseInt(an) > 45) {
        a = 50;
    }
    return a;
}

public void Sumar() {
    CPredio cp = new CPredio();
}

```



```

        BigDecimal[] s = new BigDecimal[7];
        s[0] = new BigDecimal(tv1.getText());
        s[1] = new BigDecimal(tv2.getText());
        s[2] = new BigDecimal(tv3.getText());
        s[3] = new BigDecimal(tv4.getText());
        s[4] = new BigDecimal(tv5.getText());
        s[5] = new BigDecimal(tv6.getText());
        s[6] = new BigDecimal(tv7.getText());
        predio = cp.vm2(s);
        th6.setText(predio.toString());
    }

    @FXML
    public void Enter(KeyEvent event) throws IOException,
Exception {
        Deprecia();
    }
}

```

class CCrearPredio

```

/*
 * To change this license header, choose License Headers in
Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package control;

import dao.DBano;
import dao.DaoConservacion;
import dao.DaoHabitacion;
import dao.DaoInstalaciones;
import dao.DaoMaterialMuro;
import dao.DaoMatpredominante;
import dao.DaoPiso;
import dao.DaoPredio;
import dao.DaoPuerta;
import dao.DaoRevestimiento;
import dao.DaoTecho;
import java.io.IOException;
import java.math.BigDecimal;
import java.net.URL;
import java.util.ResourceBundle;
import javafx.fxml.FXML;
import javafx.fxml.Initializable;
import javafx.scene.control.ComboBox;
import javafx.scene.control.TextField;
import javafx.scene.input.KeyEvent;
import javafx.stage.Stage;
import modelo.Bano;
import modelo.Instalaciones;
import modelo.Materialmuro;
import modelo.Piso;
import modelo.Predio;
import modelo.Propiedad;
import modelo.Puerta;
import modelo.Revestimiento;

```

```

import modelo.Techo;

/**
 *
 * @author YONNY
 */
public class CCrearPredio implements Initializable {

    @FXML
    ComboBox ch1, ch2, ch3, cv1, cv2, cv3, cv4, cv5, cv6, cv7;
    @FXML
    TextField th1, th2, th3, th4;
    @FXML
    TextField tv1, tv2, tv3, tv4, tv5, tv6, tv7;
    @FXML
    TextField th5, th6;

    public Stage stage;
    public Predio reg;
    public boolean isOk = false;
    Propiedad pro;

    public BigDecimal predio;
    public BigDecimal depreciacion;

    @Override
    public void initialize(URL location, ResourceBundle
resources) {

        DaoHabitacion dh = new DaoHabitacion();
        dh.all().stream().map((h) -> {
            ch1.getItems().addAll(h.getDescripcion());
            return h;
        }).forEachOrdered((__item) -> {
            ch1.getSelectionModel().select(0);
        });

        DaoMatpredominante dm = new DaoMatpredominante();
        dm.all().stream().map((h) -> {
            ch2.getItems().addAll(h.getDescripcion());
            return h;
        }).forEachOrdered((__item) -> {
            ch2.getSelectionModel().select(0);
        });

        DaoConservacion dc = new DaoConservacion();
        dc.all().stream().map((h) -> {
            ch3.getItems().addAll(h.getDescripcion());
            return h;
        }).forEachOrdered((__item) -> {
            ch3.getSelectionModel().select(0);
        });
        /*---*/
        DaoMaterialMuro dmm = new DaoMaterialMuro();
        dmm.all().stream().map((h) -> {
            cv1.getItems().addAll(h.getLetra() + " : " +
h.getDescripcion());
            return h;
        }).forEachOrdered((__item) -> {

```

```

        cv1.getSelectionModel().select(0);
    });

    DaoTecho dt = new DaoTecho();
    dt.all().stream().map(h -> {
        cv2.getItems().addAll(h.getLetra() + " : " +
h.getDescripcion());
        return h;
    }).forEachOrdered((_item) -> {
        cv2.getSelectionModel().select(0);
    });

    DaoPiso dp = new DaoPiso();
    dp.all().stream().map(h -> {
        cv3.getItems().addAll(h.getLetra() + " : " +
h.getDescripcion());
        return h;
    }).forEachOrdered((_item) -> {
        cv3.getSelectionModel().select(0);
    });

    DaoPuerta dpt = new DaoPuerta();
    dpt.all().stream().map(h -> {
        cv4.getItems().addAll(h.getLetra() + " : " +
h.getDescripcion());
        return h;
    }).forEachOrdered((_item) -> {
        cv4.getSelectionModel().select(0);
    });

    DaoRevestimiento dr = new DaoRevestimiento();
    dr.all().stream().map(h -> {
        cv5.getItems().addAll(h.getLetra() + " : " +
h.getDescripcion());
        return h;
    }).forEachOrdered((_item) -> {
        cv5.getSelectionModel().select(0);
    });

    DBano db = new DBano();
    db.all().stream().map(h -> {
        cv6.getItems().addAll(h.getLetra() + " : " +
h.getDescripcion());
        return h;
    }).forEachOrdered((_item) -> {
        cv6.getSelectionModel().select(0);
    });

    DaoInstalaciones di = new DaoInstalaciones();
    di.all().stream().map(h -> {
        cv7.getItems().addAll(h.getLetra() + " : " +
h.getDescripcion());
        return h;
    }).forEachOrdered((_item) -> {
        cv7.getSelectionModel().select(0);
    });

    Muro();
    Piso();

```

```

Puerta ();
Revestimiento ();
Techo ();
Bano ();
Electrica ();
Deprecia ();
}

public void setPerson(Propiedad pr, String a) {
    th3.setText(a);
    pro = pr;
}

@FXML
public void Enter() {

}

@FXML
public void Save() {
    DaoMaterialMuro d1 = new DaoMaterialMuro();
    String r1 = (String) cv1.getValue();
    r1 = r1.substring(0, 1);
    Materialmuro i1 = d1.data(r1);

    DaoTecho d2 = new DaoTecho();
    String r2 = (String) cv2.getValue();
    r2 = r2.substring(0, 1);
    Techo i2 = d2.data(r2);

    DaoPiso d3 = new DaoPiso();
    String r3 = (String) cv3.getValue();
    r3 = r3.substring(0, 1);
    Piso i3 = d3.data(r3);

    DaoPuerta d4 = new DaoPuerta();
    String r4 = (String) cv4.getValue();
    r4 = r4.substring(0, 1);
    Puerta i4 = d4.data(r4);

    DaoRevestimiento d5 = new DaoRevestimiento();
    String r5 = (String) cv5.getValue();
    r5 = r5.substring(0, 1);
    Revestimiento i5 = d5.data(r5);

    DBano d6 = new DBano();
    String r6 = (String) cv6.getValue();
    r6 = r6.substring(0, 1);
    Bano i6 = d6.data(r6);

    DaoInstalaciones d7 = new DaoInstalaciones();
    String r7 = (String) cv7.getValue();
    r7 = r7.substring(0, 1);
    Instalaciones i7 = d7.data(r7);

    DaoConservacion c1 = new DaoConservacion();
    DaoHabitacion c2 = new DaoHabitacion();
    DaoMatpredominante c3 = new DaoMatpredominante();
    Predio dp = new Predio(i6,

```

```

c1.data(ch3.getSelectionModel().getSelectedIndex() + 1),
c2.data(ch1.getSelectionModel().getSelectedIndex() + 1),
        i7, i1,

c3.data(ch2.getSelectionModel().getSelectedIndex() + 1),
        i3, pro, i4, i5, i2,
Integer.parseInt(th3.getText()), th1.getText() + " " +
th2.getText(), th4.getText(),
        predio, new BigDecimal(th5.getText()),
depreciacion);
        DaoPredio c4 = new DaoPredio();
        c4.save(dp);
        isOk = true;
        stage.close();
    }

@FXML
public void Salir() {
    stage.close();
}

void setDialogStage(Stage dialogStage) {
    stage = dialogStage;
}

boolean isOk() {
    return isOk;
}

@FXML
public void Muro() {
    DaoMaterialMuro di = new DaoMaterialMuro();
    String r = (String) cv1.getValue();
    r = r.substring(0, 1);
    Materialmuro i = di.data(r);
    tv1.setText(i.getMonto().toString());
    Sumar();
}

@FXML
public void Techo() {
    DaoTecho di = new DaoTecho();
    String r = (String) cv2.getValue();
    r = r.substring(0, 1);
    Techo i = di.data(r);
    tv2.setText(i.getMonto().toString());
    Sumar();
}

@FXML
public void Piso() {
    DaoPiso di = new DaoPiso();
    String r = (String) cv3.getValue();
    r = r.substring(0, 1);
    Piso i = di.data(r);
    tv3.setText(i.getMonto().toString());
    Sumar();
}

```

```

}

@FXML
public void Puerta() {
    DaoPuerta di = new DaoPuerta();
    String r = (String) cv4.getValue();
    r = r.substring(0, 1);
    Puerta i = di.data(r);
    tv4.setText(String.valueOf(i.getMonto()));
    Sumar();
}

@FXML
public void Revestimiento() {
    DaoRevestimiento di = new DaoRevestimiento();
    String r = (String) cv5.getValue();
    r = r.substring(0, 1);
    Revestimiento i = di.data(r);
    tv5.setText(i.getMonto().toString());
    Sumar();
}

@FXML
public void Bano() {
    DBano di = new DBano();
    String r = (String) cv6.getValue();
    r = r.substring(0, 1);
    Bano i = di.data(r);
    tv6.setText(i.getMonto().toString());
    Sumar();
}

@FXML
public void Electrica() {
    DaoInstalaciones di = new DaoInstalaciones();
    String r = (String) cv7.getValue();
    r = r.substring(0, 1);
    Instalaciones i = di.data(r);
    tv7.setText(i.getMonto().toString());
    Sumar();
}

@FXML
public void Deprecia() {
    CPredio cp = new CPredio();
    depreciacion =
cp.vdep(ch1.getSelectionModel().getSelectedIndex() + 1,
        antiguedad(th4.getText()),
        ch2.getSelectionModel().getSelectedIndex() + 1,
        ch3.getSelectionModel().getSelectedIndex() + 1);
    System.out.println("Deprecia : " +
depreciacion.toString());
}

public int antiguedad(String an) {
    int a = 5;
    if (Integer.parseInt(an) <= 5) {
        a = 5;
    } else if (Integer.parseInt(an) <= 10) {

```

```

        a = 10;
    } else if (Integer.parseInt(an) <= 15) {
        a = 15;
    } else if (Integer.parseInt(an) <= 20) {
        a = 20;
    } else if (Integer.parseInt(an) <= 25) {
        a = 25;
    } else if (Integer.parseInt(an) <= 30) {
        a = 30;
    } else if (Integer.parseInt(an) <= 35) {
        a = 35;
    } else if (Integer.parseInt(an) <= 40) {
        a = 40;
    } else if (Integer.parseInt(an) <= 45) {
        a = 45;
    } else if (Integer.parseInt(an) > 45) {
        a = 50;
    }
    return a;
}

public void Sumar() {
    CPredio cp = new CPredio();
    BigDecimal[] s = new BigDecimal[7];
    s[0] = new BigDecimal(tv1.getText());
    s[1] = new BigDecimal(tv2.getText());
    s[2] = new BigDecimal(tv3.getText());
    s[3] = new BigDecimal(tv4.getText());
    s[4] = new BigDecimal(tv5.getText());
    s[5] = new BigDecimal(tv6.getText());
    s[6] = new BigDecimal(tv7.getText());
    predio = cp.vm2(s);
    th6.setText(predio.toString());
}

@FXML
public void Enter(KeyEvent event) throws IOException,
Exception {
    Deprecia();
}
}

```

class CCrearPredioR

```

/*
 * To change this license header, choose License Headers in
 * Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package control;

import dao.DBano;
import dao.DaoConservacion;
import dao.DaoHabitacion;
import dao.DaoInstalaciones;
import dao.DaoMaterialMuro;
import dao.DaoMatpredominante;

```

```

import dao.DaoPiso;
import dao.DaoPredio;
import dao.DaoPuerta;
import dao.DaoRevestimiento;
import dao.DaoTecho;
import java.io.IOException;
import java.math.BigDecimal;
import java.net.URL;
import java.util.ResourceBundle;
import javafx.fxml.FXML;
import javafx.fxml.Initializable;
import javafx.scene.control.ComboBox;
import javafx.scene.control.TextField;
import javafx.scene.input.KeyEvent;
import javafx.stage.Stage;
import modelo.Bano;
import modelo.Instalaciones;
import modelo.Materialmuro;
import modelo.Piso;
import modelo.Predio;
import modelo.Propiedad;
import modelo.Puerta;
import modelo.Revestimiento;
import modelo.Techo;

/**
 *
 * @author YONNY
 */
public class CCrearPredioR implements Initializable {

    @FXML
    ComboBox ch1, ch2, ch3, cv1, cv2, cv3, cv4, cv5, cv6, cv7;
    @FXML
    TextField th1, th2, th3, th4;
    @FXML
    TextField tv1, tv2, tv3, tv4, tv5, tv6, tv7;
    @FXML
    TextField th5, th6;

    public Stage stage;
    public Predio reg;
    public boolean isOk = false;
    Propiedad pro;

    public BigDecimal predio;
    public BigDecimal depreciacion;

    @Override
    public void initialize(URL location, ResourceBundle
resources) {

        DaoHabitacion dh = new DaoHabitacion();
        dh.all().stream().map((h) -> {
            ch1.getItems().addAll(h.getDescripcion());
            return h;
        }).forEachOrdered((item) -> {
            ch1.getSelectionModel().select(0);
        });
    }
}

```



```

DaoMatpredominante dm = new DaoMatpredominante ();
dm.all ().stream ().map ((h) -> {
    ch2.getItems ().addAll (h.getDescripcion ());
    return h;
}).forEachOrdered ((_item) -> {
    ch2.getSelectionModel ().select (0);
});

DaoConservacion dc = new DaoConservacion ();
dc.all ().stream ().map ((h) -> {
    ch3.getItems ().addAll (h.getDescripcion ());
    return h;
}).forEachOrdered ((_item) -> {
    ch3.getSelectionModel ().select (0);
});
/*---*/
DaoMaterialMuro dmm = new DaoMaterialMuro ();
dmm.all ().stream ().map ((h) -> {
    cv1.getItems ().addAll (h.getLetra () + " : " +
h.getDescripcion ());
    return h;
}).forEachOrdered ((_item) -> {
    cv1.getSelectionModel ().select (0);
});

DaoTecho dt = new DaoTecho ();
dt.all ().stream ().map ((h) -> {
    cv2.getItems ().addAll (h.getLetra () + " : " +
h.getDescripcion ());
    return h;
}).forEachOrdered ((_item) -> {
    cv2.getSelectionModel ().select (0);
});

DaoPiso dp = new DaoPiso ();
dp.all ().stream ().map ((h) -> {
    cv3.getItems ().addAll (h.getLetra () + " : " +
h.getDescripcion ());
    return h;
}).forEachOrdered ((_item) -> {
    cv3.getSelectionModel ().select (0);
});

DaoPuerta dpt = new DaoPuerta ();
dpt.all ().stream ().map ((h) -> {
    cv4.getItems ().addAll (h.getLetra () + " : " +
h.getDescripcion ());
    return h;
}).forEachOrdered ((_item) -> {
    cv4.getSelectionModel ().select (0);
});

DaoRevestimiento dr = new DaoRevestimiento ();
dr.all ().stream ().map ((h) -> {
    cv5.getItems ().addAll (h.getLetra () + " : " +
h.getDescripcion ());
    return h;
}).forEachOrdered ((_item) -> {

```

```

        cv5.getSelectionModel().select(0);
    });

    DBano db = new DBano();
    db.all().stream().map(h -> {
        cv6.getItems().addAll(h.getLetra() + " : " +
h.getDescripcion());
        return h;
    }).forEachOrdered((_item) -> {
        cv6.getSelectionModel().select(0);
    });

    DaoInstalaciones di = new DaoInstalaciones();
    di.all().stream().map(h -> {
        cv7.getItems().addAll(h.getLetra() + " : " +
h.getDescripcion());
        return h;
    }).forEachOrdered((_item) -> {
        cv7.getSelectionModel().select(0);
    });

    Muro();
    Piso();
    Puerta();
    Revestimiento();
    Techo();
    Bano();
    Electrica();
    Deprecia();
}

public void setPerson(Propiedad pr, String a) {
    th3.setText(a);
    pro = pr;
}

@FXML
public void Enter() {

}

@FXML
public void Save() {
    DaoMaterialMuro d1 = new DaoMaterialMuro();
    String r1 = (String) cv1.getValue();
    r1 = r1.substring(0, 1);
    Materialmuro i1 = d1.data(r1);

    DaoTecho d2 = new DaoTecho();
    String r2 = (String) cv2.getValue();
    r2 = r2.substring(0, 1);
    Techo i2 = d2.data(r2);

    DaoPiso d3 = new DaoPiso();
    String r3 = (String) cv3.getValue();
    r3 = r3.substring(0, 1);
    Piso i3 = d3.data(r3);

    DaoPuerta d4 = new DaoPuerta();

```

```

String r4 = (String) cv4.getValue();
r4 = r4.substring(0, 1);
Puerta i4 = d4.data(r4);

DaoRevestimiento d5 = new DaoRevestimiento();
String r5 = (String) cv5.getValue();
r5 = r5.substring(0, 1);
Revestimiento i5 = d5.data(r5);

DBano d6 = new DBano();
String r6 = (String) cv6.getValue();
r6 = r6.substring(0, 1);
Bano i6 = d6.data(r6);

DaoInstalaciones d7 = new DaoInstalaciones();
String r7 = (String) cv7.getValue();
r7 = r7.substring(0, 1);
Instalaciones i7 = d7.data(r7);

DaoConservacion c1 = new DaoConservacion();
DaoHabitacion c2 = new DaoHabitacion();
DaoMatpredominante c3 = new DaoMatpredominante();
Predio dp = new Predio(i6,

c1.data(ch3.getSelectionModel().getSelectedIndex() + 1),
c2.data(ch1.getSelectionModel().getSelectedIndex() + 1),
    i7, i1,

c3.data(ch2.getSelectionModel().getSelectedIndex() + 1),
    i3, pro, i4, i5, i2,
Integer.parseInt(th3.getText()), th1.getText() + " " +
th2.getText(), th4.getText(),
    predio, new BigDecimal(th5.getText()),
depreciacion);
    DaoPredio c4 = new DaoPredio();
    c4.save(dp);
    isOk = true;
    stage.close();
}

@FXML
public void Salir() {
    stage.close();
}

void setDialogStage(Stage dialogStage) {
    stage = dialogStage;
}

boolean isOk() {
    return isOk;
}

@FXML
public void Muro() {
    DaoMaterialMuro di = new DaoMaterialMuro();
    String r = (String) cv1.getValue();
    r = r.substring(0, 1);

```

```

        Materialmuro i = di.data(r);
        tv1.setText(i.getMonto().toString());
        Sumar();
    }

    @FXML
    public void Techo() {
        DaoTecho di = new DaoTecho();
        String r = (String) cv2.getValue();
        r = r.substring(0, 1);
        Techo i = di.data(r);
        tv2.setText(i.getMonto().toString());
        Sumar();
    }

    @FXML
    public void Piso() {
        DaoPiso di = new DaoPiso();
        String r = (String) cv3.getValue();
        r = r.substring(0, 1);
        Piso i = di.data(r);
        tv3.setText(i.getMonto().toString());
        Sumar();
    }

    @FXML
    public void Puerta() {
        DaoPuerta di = new DaoPuerta();
        String r = (String) cv4.getValue();
        r = r.substring(0, 1);
        Puerta i = di.data(r);
        tv4.setText(String.valueOf(i.getMonto()));
        Sumar();
    }

    @FXML
    public void Revestimiento() {
        DaoRevestimiento di = new DaoRevestimiento();
        String r = (String) cv5.getValue();
        r = r.substring(0, 1);
        Revestimiento i = di.data(r);
        tv5.setText(i.getMonto().toString());
        Sumar();
    }

    @FXML
    public void Bano() {
        DBano di = new DBano();
        String r = (String) cv6.getValue();
        r = r.substring(0, 1);
        Bano i = di.data(r);
        tv6.setText(i.getMonto().toString());
        Sumar();
    }

    @FXML
    public void Electrica() {
        DaoInstalaciones di = new DaoInstalaciones();
        String r = (String) cv7.getValue();

```

```

        r = r.substring(0, 1);
        Instalaciones i = di.data(r);
        tv7.setText(i.getMonto().toString());
        Sumar();
    }

    @FXML
    public void Deprecia() {
        CPredio cp = new CPredio();
        depreciacion =
cp.vdep(ch1.getSelectionModel().getSelectedIndex() + 1,
        antiguedad(th4.getText()),
        ch2.getSelectionModel().getSelectedIndex() + 1,
        ch3.getSelectionModel().getSelectedIndex() + 1);
        System.out.println("Deprecia : " +
depreciacion.toString());
    }

    public int antiguedad(String an) {
        int a = 5;
        if (Integer.parseInt(an) <= 5) {
            a = 5;
        } else if (Integer.parseInt(an) <= 10) {
            a = 10;
        } else if (Integer.parseInt(an) <= 15) {
            a = 15;
        } else if (Integer.parseInt(an) <= 20) {
            a = 20;
        } else if (Integer.parseInt(an) <= 25) {
            a = 25;
        } else if (Integer.parseInt(an) <= 30) {
            a = 30;
        } else if (Integer.parseInt(an) <= 35) {
            a = 35;
        } else if (Integer.parseInt(an) <= 40) {
            a = 40;
        } else if (Integer.parseInt(an) <= 45) {
            a = 45;
        } else if (Integer.parseInt(an) > 45) {
            a = 50;
        }
        return a;
    }

    public void Sumar() {
        CPredio cp = new CPredio();
        BigDecimal[] s = new BigDecimal[7];
        s[0] = new BigDecimal(tv1.getText());
        s[1] = new BigDecimal(tv2.getText());
        s[2] = new BigDecimal(tv3.getText());
        s[3] = new BigDecimal(tv4.getText());
        s[4] = new BigDecimal(tv5.getText());
        s[5] = new BigDecimal(tv6.getText());
        s[6] = new BigDecimal(tv7.getText());
        predio = cp.vm2(s);
        th6.setText(predio.toString());
    }

    @FXML

```

```

        public void Enter(KeyEvent event) throws IOException,
        Exception {
            Deprecia ();
        }
    }
}

```

class CCrearPropiedad

```

/*
 * To change this license header, choose License Headers in
 * Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package control;

import dao.DaoPropiedad;
import java.io.IOException;
import java.net.URL;
import java.util.ResourceBundle;
import javafx.fxml.FXML;
import javafx.fxml.Initializable;
import javafx.scene.control.ComboBox;
import javafx.scene.control.Label;
import javafx.scene.control.TextField;
import javafx.scene.input.KeyCode;
import javafx.scene.input.KeyEvent;
import javafx.stage.Stage;
import modelo.Propiedad;
import modelo.Propietario;

/**
 *
 * @author YONNY
 */
public class CCrearPropiedad implements Initializable {

    @FXML
    public TextField t1;

    @FXML
    public TextField t2;

    @FXML
    public TextField t3;

    @FXML
    public TextField t4;

    @FXML
    public ComboBox de;

    @FXML
    public Label et1;
    @FXML
    public Label et2;
    @FXML
    public Label et3;
}

```

```

@FXML
public Label et4;

public Stage stage;
public Propiedad reg;
public boolean isOkk = false;
Propietario pro;

public void setDialogStage(Stage dialogStage) {
    stage = dialogStage;
}

public void setPerson(Propiedad re, Propietario pr) {
    pro = pr;
    reg = re;
    t1.setText(re.getDireccion());
    t2.setText(re.getDep());
    t3.setText(re.getPro());
    t4.setText(re.getDis());
}

public boolean isOk() {
    return isOkk;
}

@FXML
public void Cancel() {
    stage.close();
}

@FXML
private void Savep() {
    if (isInputV()) {
reg.setDireccion(de.getValue().toString().substring(0, 3) + " "
+ t1.getText());
        reg.setDep(t2.getText());
        reg.setPro(t3.getText());
        reg.setDis(t4.getText());
        reg.setPropietario(pro);
        DaoPropiedad dr = new DaoPropiedad();
        dr.save(reg);
        isOkk = true;
        stage.close();
    }
}

private boolean isInputV() {
    int i = 0;
    if (t1.getText() == null || t1.getText().length() == 0)
{
        et1.setText("La direccion es requerido");
        i++;
    }
    if (t2.getText() == null || t2.getText().length() == 0)
{
        et2.setText("El departamento es requerido");
        i++;
    }
}

```

```

        if (t3.getText() == null || t3.getText().length() == 0)
        {
            et3.setText("El provincia es requerido");
            i++;
        }
        if (t4.getText() == null || t4.getText().length() == 0)
        {
            et4.setText("El distrito es requerida");
            i++;
        }
        return i == 0;
    }

    public void Enter(KeyEvent event) throws IOException,
Exception {
        if (event.getCode() == KeyCode.ENTER) {
            Savep();
        }
    }

    @Override
    public void initialize(URL location, ResourceBundle
resources) {
        de.getItems().addAll("CP. Centro Poblado", "AV.
Avenida", "JR. Jiron", "PJ. Pasaje", "OT. Otro");
        de.getSelectionModel().select(0);
    }
}

```

class CCrearPropiedadR

```

/*
 * To change this license header, choose License Headers in
Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package control;

import dao.DaoPropiedad;
import java.io.IOException;
import java.net.URL;
import java.util.ResourceBundle;
import javafx.fxml.FXML;
import javafx.fxml.Initializable;
import javafx.scene.control.ComboBox;
import javafx.scene.control.Label;
import javafx.scene.control.TextField;
import javafx.scene.input.KeyCode;
import javafx.scene.input.KeyEvent;
import javafx.stage.Stage;
import modelo.Propiedad;
import modelo.Propietario;

/**
 *
 */

```



```

* @author YONNY
*/
public class CCrearPropiedadR implements Initializable {

    @FXML
    public TextField t1;

    @FXML
    public TextField t2;

    @FXML
    public TextField t3;

    @FXML
    public TextField t4;

    @FXML
    public TextField t5;

    @FXML
    public TextField t6;

    @FXML
    public TextField t7;

    @FXML
    public ComboBox de;

    @FXML
    public Label et1;
    @FXML
    public Label et2;
    @FXML
    public Label et3;
    @FXML
    public Label et4;

    public Stage stage;
    public Propiedad reg;
    public boolean isOkk = false;
    Propietario pro;

    public void setDialogStage(Stage dialogStage) {
        stage = dialogStage;
    }

    public void setPerson(Propiedad re, Propietario pr) {
        pro = pr;
        reg = re;
        t1.setText(re.getDireccion());
        t2.setText(re.getDep());
        t3.setText(re.getPro());
        t4.setText(re.getDis());
    }

    public boolean isOk() {
        return isOkk;
    }
}

```

```

@FXML
public void Cancel() {
    stage.close();
}

@FXML
private void Savep() {
    if (isInputV()) {
reg.setDireccion(de.getValue().toString().substring(0, 3) + " "
+ t1.getText());
        reg.setDep(t2.getText());
        reg.setPro(t3.getText());
        reg.setDis(t4.getText());
        reg.setPropietario(pro);
        DaoPropiedad dr = new DaoPropiedad();
        dr.save(reg);
        isOkk = true;
        stage.close();
    }
}

private boolean isInputV() {
    int i = 0;
    if (t1.getText() == null || t1.getText().length() == 0)
{
        et1.setText("La direccion es requerido");
        i++;
    }
    if (t2.getText() == null || t2.getText().length() == 0)
{
        et2.setText("El departamento es requerido");
        i++;
    }
    if (t3.getText() == null || t3.getText().length() == 0)
{
        et3.setText("El provincia es requerido");
        i++;
    }
    if (t4.getText() == null || t4.getText().length() == 0)
{
        et4.setText("El distrito es requerida");
        i++;
    }
    return i == 0;
}

public void Enter(KeyEvent event) throws IOException,
Exception {
    if (event.getCode() == KeyCode.ENTER) {
        Savep();
    }
}

@Override
public void initialize(URL location, ResourceBundle
resources) {
    de.getItems().addAll("SA. Santillana", "CP. Centro
Poblado", "OT. Otro");
}

```

```

        de.getSelectionModel().select(0);
    }
}

```

class CEditDialog

```

/*
 * To change this license header, choose License Headers in
 * Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package control;

import dao.DUs;
import java.io.IOException;
import javafx.fxml.FXML;
import javafx.scene.Node;
import javafx.scene.control.Label;
import javafx.scene.control.TextField;
import javafx.scene.input.KeyCode;
import javafx.scene.input.KeyEvent;
import javafx.stage.Stage;
import modelo.Usuario;

/**
 *
 * @author Manrrique
 */
public class CEditDialog {

    @FXML
    public TextField n;
    @FXML
    public TextField d;
    @FXML
    public TextField u;
    @FXML
    public TextField c;

    @FXML
    public Label en;
    @FXML
    public Label ed;
    @FXML
    public Label eu;
    @FXML
    public Label ec;

    public Stage stage;
    public Usuario reg;
    public boolean isOk = false;

    public void setDialogStage(Stage dialogStage) {
        stage = dialogStage;
    }

    public void setPerson(Usuario re) {

```

```

        reg = re;
        n.setText(re.getNombre());
        d.setText(re.getDni());
        u.setText(re.getUsu());
        c.setText(re.getCont());
    }

    public boolean isOk() {
        return isOk;
    }

    @FXML
    public void Cancel() {
        stage.close();
    }

    @FXML
    private void Save() {
        if (isInputValid()) {
            reg.setNombre(n.getText());
            reg.setDni(d.getText());
            reg.setUsu(u.getText());
            reg.setCont(c.getText());
            DUs dr = new DUs();
            dr.save(reg);
            isOk = true;
            stage.close();
        }
    }

    private boolean isInputValid() {
        int i = 0;
        if (n.getText() == null || n.getText().length() == 0) {
            en.setText("El nombre es requerido");
            i++;
        }
        if (d.getText() == null || d.getText().length() == 0) {
            ed.setText("El DNI es requerido");
            i++;
        }
        if (u.getText() == null || u.getText().length() == 0) {
            eu.setText("El usuario es requerido");
            i++;
        }
        if (c.getText() == null || c.getText().length() == 0) {
            ec.setText("La contraseña es requerida");
            i++;
        }
        return i == 0;
    }

    public void Enter(KeyEvent event) throws IOException,
    Exception {
        if (event.getCode() == KeyCode.ENTER) {
            Save();
        }
    }
}

```

class CLogin

```
/*
 * To change this license header, choose License Headers in
 Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package control;

import dao.DUs;
import java.io.IOException;
import java.net.URL;
import java.util.ResourceBundle;
import javafx.fxml.FXML;
import javafx.fxml.Initializable;
import javafx.scene.Node;
import javafx.scene.control.Label;
import javafx.scene.control.PasswordField;
import javafx.scene.control.ProgressBar;
import javafx.scene.control.TextField;
import javafx.scene.input.KeyCode;
import javafx.scene.input.KeyEvent;
import javafx.scene.input.MouseEvent;
import javafx.stage.Stage;
import modelo.Usuario;

/*
 * @author Manrrique
 */
public class CLogin implements Initializable {

    @FXML
    public TextField us;
    @FXML
    public PasswordField ps;
    @FXML
    public Label msg;

    public static Usuario User;

    @FXML
    public ProgressBar pb;

    @FXML
    public void Ingresar(MouseEvent event) throws IOException,
Exception {
        if (Acceso()) {
            Node node = (Node) event.getSource();
            Stage stage = (Stage) node.getScene().getWindow();
            Mains(stage);
        }
    }

    public boolean Acceso() {
        if (!us.getText().equals("") &&
!ps.getText().equals("")) {
```

```

        DUs rg = new DUs();
        User = rg.data(us.getText(), ps.getText());
        if (User != null) {
            return true;
        } else {
            msg.setText("El usuario o contraseña
incorrecto");
            return false;
        }
    } else {
        msg.setText("Llene loa campos vacios");
        return false;
    }
}

public void Enter(KeyEvent event) throws IOException,
Exception {
    if (event.getCode() == KeyCode.ENTER) {
        if (Acceso()) {
            Node node = (Node) event.getSource();
            Stage stage = (Stage)
node.getScene().getWindow();
            Mains(stage);
        }
    }
}

public void Mains(Stage stage) throws IOException, Exception
{
    stage.close();
    CMain cm = new CMain();
    cm.start(new Stage());
}

@FXML
public void Salir() {
    System.exit(0);
}

@Override
public void initialize(URL location, ResourceBundle
resources) {
}
}

```

class CMain

```

/*
 * To change this license header, choose License Headers in
Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package control;

```

```

import java.io.IOException;
import javafx.application.Application;
import javafx.fxml.FXML;
import javafx.fxml.FXMLLoader;
import javafx.scene.Scene;
import javafx.scene.layout.AnchorPane;
import javafx.scene.layout.BorderPane;
import javafx.stage.Stage;
import modelo.HibernateUtil;
import modelo.Usuario;
import org.hibernate.Session;

/**
 *
 * @author Manrrique
 */
public class CMain extends Application {

    private Stage s;

    @FXML
    private BorderPane m;

    public Usuario User;

    @Override
    public void start(Stage primaryStage) throws Exception {
        s = primaryStage;
        User = CLogin.User;
        Menu();
        principal();
    }

    @FXML
    public void Menu() {
        try {
            FXMLLoader loader = new
FXMLLoader(getClass().getResource("/view/VMenu.fxml"));
            m = (BorderPane) loader.load();
            Scene scene = new Scene(m);
            s.setScene(scene);
            s.show();
        } catch (Exception e) {
            System.out.println(e);
        }
    }

    @FXML
    public void Caja() throws IOException {
        View("/view/VCaja.fxml");
    }

    @FXML
    public void Registro() throws IOException {
        View("/view/VRegistra.fxml");
    }

    //rustico

```

```

@FXML
public void CajaR() throws IOException {
    View("/view/VCajaR.fxml");
}

@FXML
public void ContribuyenteR() throws IOException {
    View("/view/ContribuyenteR.fxml");
}
//rustico

@FXML
public void Salir() {
    System.exit(0);
}

@FXML
public void Main() throws IOException {
    View("/view/VLogin.fxml");
}

@FXML
public void Contribuyente() throws IOException {
    View("/view/Contribuyente.fxml");
}

@FXML
public void Cayuda() throws IOException {
    View("/view/Vayuda.fxml");
}

@FXML
public void principal() throws IOException {
    View("/view/VMenuPrinc.fxml");
}

@FXML
public void Cnlegales() throws IOException {
    View("/view/Vnlegales.fxml");
}

public void View(String vista) throws IOException {
    m.setCenter((AnchorPane)
FXMLLoader.load(getClass().getResource(vista)));
}
}

```

class CMCagregar

```

/*
 * To change this license header, choose License Headers in
 * Project Properties.

```



```

* To change this template file, choose Tools | Templates
* and open the template in the editor.
*/
package control;

import com.jfoenix.controls.JFXTextField;
import dao.DaoPropietario;
import java.io.IOException;
import javafx.fxml.FXML;
import javafx.scene.control.Label;
import javafx.scene.input.KeyCode;
import javafx.scene.input.KeyEvent;
import javafx.stage.Stage;
import modelo.Propietario;

/**
 *
 * @author YONNY
 */
public class CMCagregar {

    @FXML
    public JFXTextField ptip;
    @FXML
    public JFXTextField pnum;
    @FXML
    public JFXTextField pnom;
    @FXML
    public JFXTextField pcel;
    @FXML
    public JFXTextField pcor;
    @FXML
    public JFXTextField pobs;

    @FXML
    public Label eptip;
    @FXML
    public Label epnum;
    @FXML
    public Label epnom;
    @FXML
    public Label epcel;
    @FXML
    public Label epcor;
    @FXML
    public Label epobs;

    public Stage stage;
    public Propietario reg;
    public boolean isOk = false;

    public void setDialogStage(Stage dialogStage) {
        stage = dialogStage;
    }

    public void setPerson(Propietario re) {
        reg = re;

        pnum.setText(re.getNumdoc());
    }
}

```

```

        pnom.setText(re.getNombre());
        pcel.setText(re.getCelular());
        pcor.setText(re.getCorreo());
        pobs.setText(re.getObs());

        if (re.getIdpro() == null) {
            ptip.setText("DNI");
            pcel.setText("CASA - HABITACION");
            pcor.setText("TERMINADO");
            pobs.setText("PREDIO TOTAL");
        }
    }

    public boolean isOk() {
        return isOk;
    }

    @FXML
    public void Cancel() {
        stage.close();
    }

    @FXML
    private void Savee() {
        if (isInputValid()) {
            try {
                reg.setTipodoc(ptip.getText());
                reg.setNumdoc(pnum.getText());
                reg.setNombre(pnom.getText());
                reg.setCelular(pcel.getText());
                reg.setCorreo(pcor.getText());
                reg.setObs(pobs.getText());
                DaoPropietario dr = new DaoPropietario();
                dr.save(reg);
                isOk = true;
                stage.close();
            } catch (Exception e) {
                System.out.println(e);
            }
        }
    }

    private boolean isInputValid() {
        int i = 0;
        if (ptip.getText() == null || ptip.getText().length() ==
0) {
            eptip.setText("El tipo de documento es requerido");
            i++;
        }
        if (pnum.getText() == null || pnum.getText().length() ==
0) {
            epnum.setText("El numero de documento es
requerido");
            i++;
        }
        if (pnom.getText() == null || pnom.getText().length() ==
0) {
            epnom.setText("El nombre es requerido");
            i++;
        }
    }

```

```

    }
    if (pcel.getText() == null || pcel.getText().length() ==
0) {
        epcel.setText("el numero de celular es requerida");
        i++;
    }
    if (pcor.getText() == null || pcor.getText().length() ==
0) {
        epcor.setText("el correo es requerida");
        i++;
    }
    if (pobs.getText() == null || pobs.getText().length() ==
0) {
        epobs.setText("observacion es requerida");
        i++;
    }
    return i == 0;
}

public void Enter(KeyEvent event) throws IOException,
Exception {
    if (event.getCode() == KeyCode.ENTER) {
        Savee();
    }
}
}
}

```

class CPredio

```

/*
 * To change this license header, choose License Headers in
Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package control;

import dao.DaoDepreciacion;
import java.math.BigDecimal;
import java.math.RoundingMode;
import modelo.Depreciacion;

/**
 *
 * @author Manrrique
 */
public class CPredio {

    public BigDecimal vm2(BigDecimal[] b) {
        BigDecimal m2 = new BigDecimal(0);
        m2 = m2.add(b[0]);
        m2 = m2.add(b[1]);
        m2 = m2.add(b[2]);
    }
}

```

```

        m2 = m2.add(b[3]);
        m2 = m2.add(b[4]);
        m2 = m2.add(b[5]);
        m2 = m2.add(b[6]);
        return m2.setScale(2, RoundingMode.CEILING);
    }

    public BigDecimal vdep(int h, int a, int m, int e) {
        BigDecimal dp = new BigDecimal(0);
        DaoDepreciacion da = new DaoDepreciacion();
        Depreciacion de = da.data(h, a, m);
        switch (e) {
            case 1:
                dp = de.getMuybueno();
                break;
            case 2:
                dp = de.getBueno();
                break;
            case 3:
                dp = de.getRegular();
                break;
            case 4:
                dp = de.getMalo();
                break;
        }
        return dp;
    }

    public BigDecimal vter(BigDecimal s, BigDecimal a) {
        BigDecimal dp = new BigDecimal(0);
        dp = s.multiply(a);
        return dp.setScale(2, RoundingMode.CEILING);
    }
}

```

class CRegistra

```

/*
 * To change this license header, choose License Headers in
 * Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package control;

import dao.DUs;
import java.net.URL;
import java.util.Iterator;
import java.util.List;
import java.util.ResourceBundle;
import javafx.collections.FXCollections;
import javafx.collections.ObservableList;
import javafx.fxml.FXML;
import javafx.fxml.FXMLLoader;
import javafx.fxml.Initializable;
import javafx.scene.Scene;

```

```

import javafx.scene.control.Alert;
import javafx.scene.control.Label;
import javafx.scene.control.TableColumn;
import javafx.scene.control.TableView;
import javafx.scene.layout.AnchorPane;
import javafx.stage.Modality;
import javafx.stage.Stage;
import modelo.Usuario;

/**
 *
 * @author Manrique
 */
public class CRegistra implements Initializable {

    @FXML
    public Label msg;
    @FXML
    public Label n;
    @FXML
    public Label d;
    @FXML
    public Label u;
    @FXML
    public Label c;

    @FXML
    private TableView<Usuario> lista;
    @FXML
    private TableColumn<Usuario, String> c1;
    @FXML
    private TableColumn<Usuario, String> c2;

    public ObservableList<Usuario> ls =
FXCollections.observableArrayList();

    @Override
    public void initialize(URL location, ResourceBundle
resources) {
        msg.setText(CLogin.User.getNombre());
        data();
        perfil(null);

lista.getSelectionModel().selectedItemProperty().addListener(
        (observable, oldValue, ul) -> perfil(ul));

    }

    public void data() {
        msg.setText(CLogin.User.getNombre());
        perfil(null);

lista.getSelectionModel().selectedItemProperty().addListener(
        (observable, oldValue, ul) -> perfil(ul));
        ls.clear();

        DUs us = new DUs();
        List<Usuario> l = us.all();

```

```

        l.forEach((ul) -> {
            ls.add(ul);
        });

        c1.setCellValueFactory(da -> da.getValue().name());
        c2.setCellValueFactory(da -> da.getValue().dni());
        lista.setItems(ls);
    }

    public void perfil(Usuario re) {
        if (re != null) {
            n.setText(re.getNombre());
            d.setText(re.getDni());
            u.setText(re.getUsu());
            c.setText(re.getCont());
        } else {
            n.setText("");
            d.setText("");
            u.setText("");
            c.setText("");
        }
    }

    @FXML
    private void Delete() {
        try {
            int id =
lista.getSelectionModel().getSelectedIndex();
            DUs reg = new DUs();
            Usuario us =
lista.getSelectionModel().getSelectedItem();
            reg.eliminar(us);
            lista.getItems().remove(id);
        } catch (Exception e) {
            Alert al = new Alert(Alert.AlertType.ERROR);
            al.setHeaderText("Error al Eliminar");
            al.setContentText("Seleccione una fila");
            al.showAndWait();
            System.out.println(e);
        }
    }

    @FXML
    private void New() {
        Usuario re = new Usuario();
        boolean okClicked = EditDialog(re);
        if (okClicked) {
            data();
        }
    }

    public boolean EditDialog(Usuario re) {
        try {
            FXMLLoader loader = new FXMLLoader();

loader.setLocation(getClass().getResource("/view/VEditDialog.fxml
1"));

            AnchorPane page = (AnchorPane) loader.load();
            Stage stage = new Stage();

```

```

        stage.setTitle("Edit Person");
        stage.initModality(Modality.WINDOW_MODAL);
        Scene scene = new Scene(page);
        stage.setScene(scene);
        CEditDialog con = loader.getController();
        con.setDialogStage(stage);
        con.setPerson(re);
        stage.showAndWait();
        return con.isOk();
    } catch (Exception e) {
        System.out.println(e);
        return false;
    }
}

@FXML
private void Edit() {
    Usuario id =
lista.getSelectionModel().getSelectedItem();
    if (id != null) {
        boolean okClicked = EditDialog(id);
        if (okClicked) {
            data();
        }
    } else {
        Alert al = new Alert(Alert.AlertType.ERROR);
        al.setHeaderText("Error al Modificar");
        al.setContentText("Seleccione una fila");
        al.showAndWait();
    }
}
}
}

```

class IDJ

```

package control;

import com.itextpdf.text.Document;
import com.itextpdf.text.DocumentException;
import com.itextpdf.text.Image;
import com.itextpdf.text.pdf.BaseFont;
import com.itextpdf.text.pdf.PdfContentByte;
import com.itextpdf.text.pdf.PdfWriter;
import static control.Index.fo;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import java.math.BigDecimal;
import java.math.RoundingMode;
import java.time.LocalDate;
import java.util.List;
import java.util.logging.Level;
import java.util.logging.Logger;
import modelo.Caja;

```

```

import modelo.Predio;
import modelo.Propiedad;
import modelo.Propietario;
import modelo.Valorpropiedad;

public class IDJ {

    public int IP(List<Predio> l, Propietario pr, Propiedad pd,
Valorpropiedad vp, Caja c) {
        CCPredio cc = new CCPredio();
        int i = createPdf(l,
            new IDJurada(
                t8(String.valueOf(c.getIdCaja())) + "-
PU",
                String.valueOf(vp.getAno()),
                t8(String.valueOf(c.getIdCaja())),
                pr.getNumdoc(),
                pr.getNombre().toUpperCase(),
                "SANTILLANA",
                pd.getDireccion().substring(0, 3),
                pd.getDireccion().replaceFirst(pd.getDireccion().substring(0,
3), "").toUpperCase(),
                pd.getDep(),
                pd.getPro(),
                pd.getDis(),
                String.valueOf(cc.predio(pd,
vp.getAno()))),
                String.valueOf(vp.getAreat()),
                String.valueOf(vp.getArancel()),
                String.valueOf(vp.getValort()),
                String.valueOf(c.getBase()),
                "Normal",
                String.valueOf(LocalDate.now().getDayOfMonth()),
                String.valueOf(LocalDate.now().getMonth()),
                String.valueOf(LocalDate.now().getYear()),
                new IPUrbano(
                    String.valueOf(c.getAno()),
                    pr.getNombre().toUpperCase(),
                    pr.getNumdoc(),
                    fo.format(c.getFecha()),
                    pd.getDireccion().toUpperCase() + " N° "
+ pd.getDep() + " Mz. " + pd.getPro() + " Lt. " + pd.getDis(),
                    String.valueOf(c.getUit()),
                    t8(String.valueOf(c.getIdCaja())),
                    "1",
                    "0",
                    String.valueOf(c.getAno()),
                    String.valueOf(c.getBase()),
                    String.valueOf(c.getMonto())
                );
            return i;
        }

    public int createPdf(List<Predio> ls, IDJurada dec, IPUrbano
pu) {

```



```

String pdf = "D://Autovaluo/predio.pdf";
String img = "D://Autovaluo/Resources/Auto1.jpg";
String num1 = pu.getFnac(),
        anol = pu.getAno(),
        nom1 = pu.getNom(),
        dni = pu.getDni(),
        fecha = pu.getFecha(),
        dir = pu.getDir(),
        uit = pu.getUit(),
        anod = pu.getAnod(),
        der = pu.getDer(),
        anos = pu.getAnos(),
        basei = pu.getBasei(),
        ianual = pu.getIanual();
BigDecimal vtri = new BigDecimal(ianual).divide(new
BigDecimal(4)).setScale(2, RoundingMode.CEILING);

try {
    Document document = new Document();
    PdfWriter writer = PdfWriter.getInstance(document,
new FileOutputStream(pdf));
    document.open();
    Image imagen = Image.getInstance(img);
    imagen.scaleAbsoluteHeight(842);
    imagen.scaleAbsoluteWidth(595);
    imagen.setAbsolutePosition(0, 0);
    document.add(imagen);

    BaseFont bf = BaseFont.createFont();

    PdfContentByte n0 = writer.getDirectContent();
    n0.beginText();
    n0.setTextMatrix(460, a4(118));
    n0.setFontAndSize(bf, 10);
    n0.showText(num1);
    n0.endText();

    PdfContentByte n1 = writer.getDirectContent();
    n1.beginText();
    n1.setTextMatrix(277, a4(129));
    n1.setFontAndSize(bf, 10);
    n1.showText(anol);
    n1.endText();

    PdfContentByte n2 = writer.getDirectContent();
    n2.beginText();
    n2.setTextMatrix(54, a4(167));
    n2.setFontAndSize(bf, 10);
    n2.showText(nom1);
    n2.endText();

    PdfContentByte n3 = writer.getDirectContent();
    n3.beginText();
    n3.setTextMatrix(373, a4(167));
    n3.setFontAndSize(bf, 10);
    n3.showText(dni);
    n3.endText();

    PdfContentByte n4 = writer.getDirectContent();

```

```

n4.beginText ();
n4.setTextMatrix(480, a4(167));
n4.setFontAndSize(bf, 10);
n4.showText( fecha );
n4.endText ();

PdfContentByte n5 = writer.getDirectContent ();
n5.beginText ();
n5.setTextMatrix(192, a4(190));
n5.setFontAndSize(bf, 10);
n5.showText( dir );
n5.endText ();

PdfContentByte n6 = writer.getDirectContent ();
n6.beginText ();
n6.setTextMatrix(195, a4(213));
n6.setFontAndSize(bf, 10);
n6.showText( uit );
n6.endText ();

PdfContentByte n7 = writer.getDirectContent ();
n7.beginText ();
n7.setTextMatrix(388, a4(213));
n7.setFontAndSize(bf, 10);
n7.showText( anod );
n7.endText ();

PdfContentByte n8 = writer.getDirectContent ();
n8.beginText ();
n8.setTextMatrix(195, a4(236));
n8.setFontAndSize(bf, 10);
n8.showText( der );
n8.endText ();

PdfContentByte n9 = writer.getDirectContent ();
n9.beginText ();
n9.setTextMatrix(388, a4(237));
n9.setFontAndSize(bf, 10);
n9.showText( anos );
n9.endText ();

PdfContentByte n10 = writer.getDirectContent ();
n10.beginText ();
n10.setTextMatrix(144, a4(272));
n10.setFontAndSize(bf, 10);
n10.showText( basei );
n10.endText ();

PdfContentByte n11 = writer.getDirectContent ();
n11.beginText ();
n11.setTextMatrix(292, a4(272));
n11.setFontAndSize(bf, 10);
n11.showText( iannual );
n11.endText ();

PdfContentByte n12 = writer.getDirectContent ();
n12.beginText ();
n12.setTextMatrix(387, a4(272));
n12.setFontAndSize(bf, 10);

```

```

n12.showText (String.valueOf(vtri));
n12.endText ();

int y1 = a4(311);
for (int i = 0; i < 4; i++) {

    PdfContentByte n13 = writer.getDirectContent ();
    n13.beginText ();
    n13.setTextMatrix(155, y1);
    n13.setFontAndSize(bf, 10);
    n13.showText (String.valueOf(vtri));
    n13.endText ();

    PdfContentByte n14 = writer.getDirectContent ();
    n14.beginText ();
    n14.setTextMatrix(249, y1);
    n14.setFontAndSize(bf, 10);
    n14.showText ("0");
    n14.endText ();

    PdfContentByte n15 = writer.getDirectContent ();
    n15.beginText ();
    n15.setTextMatrix(328, y1);
    n15.setFontAndSize(bf, 10);
    n15.showText (String.valueOf(vtri));
    n15.endText ();

    PdfContentByte n16 = writer.getDirectContent ();
    n16.beginText ();
    n16.setTextMatrix(404, y1);
    n16.setFontAndSize(bf, 10);
    n16.showText ("X");
    n16.endText ();

    PdfContentByte n17 = writer.getDirectContent ();
    n17.beginText ();
    n17.setTextMatrix(430, y1);
    n17.setFontAndSize(bf, 10);
    n17.showText ("Pagar");
    n17.endText ();

    PdfContentByte n18 = writer.getDirectContent ();
    n18.beginText ();
    n18.setTextMatrix(490, y1);
    n18.setFontAndSize(bf, 10);
    n18.showText ("A");
    n18.endText ();

    y1 -= 22;
}

PdfContentByte n19 = writer.getDirectContent ();
n19.beginText ();
n19.setTextMatrix(328, a4(400));
n19.setFontAndSize(bf, 10);
n19.showText (ianual);
n19.endText ();

Numeros ns = new Numeros ();

```

```

PdfContentByte n20 = writer.getDirectContent ();
n20.beginText ();
n20.setTextMatrix(86, 421);
n20.setFontAndSize(bf, 10);
n20.showText (ns.Convertir(ianual, true));
n20.endText ();

document.newPage ();
//String pdf1 = "D://Autovaluo/djurada.pdf";
String img1 = "D://Autovaluo/Resources/Auto2.jpg";
String codigo = dec.getCodigo (),
      ano = dec.getAno (),
      n_ddjj = dec.getN_ddjj (),
      n_dni_ruc = dec.getN_dni_ruc (),
      ape_nom = dec.getApe_nom (),
      lugar = dec.getLugar (),
      denom = dec.getDenom (),
      direc = dec.getDirec (),
      num = dec.getNum (),
      manz = dec.getManz (),
      lote = dec.getLote (),
      area_terreno = dec.getArea_terreno (),
      arancel = dec.getArancel (),
      valortotal_const =
dec.getValortotal_const (),
      valortotal_terren =
dec.getValortotal_terren (),
      total_autovaluo = dec.getTotal_autovaluo (),
      regimen = dec.getRegimen (),
      dia = dec.getDia (),
      mes = dec.getMes (),
      añ = dec.getAñ ();

Image imagen1 = Image.getInstance (img1);
imagen1.scaleAbsoluteHeight (842);
imagen1.scaleAbsoluteWidth (595);
imagen1.setAbsolutePosition (0, 0);
document.add (imagen1);

PdfContentByte nli = writer.getDirectContent ();
nli.beginText ();
nli.setTextMatrix (50, 675);
nli.setFontAndSize (bf, 10);
nli.showText (codigo);
nli.endText ();

PdfContentByte nfo = writer.getDirectContent ();
nfo.beginText ();
nfo.setTextMatrix (320, 708);
nfo.setFontAndSize (bf, 10);
nfo.showText (ano);
nfo.endText ();

PdfContentByte no = writer.getDirectContent ();
no.beginText ();
no.setTextMatrix (320, 680);
no.setFontAndSize (bf, 10);
no.showText (n_ddjj);

```

```

no.endText ();

PdfContentByte bv = writer.getDirectContent ();
bv.beginText ();
bv.setTextMatrix (50, 605);
bv.setFontAndSize (bf, 10);
bv.showText (n_dni_ruc);
bv.endText ();

PdfContentByte c = writer.getDirectContent ();
c.beginText ();
c.setTextMatrix (135, 605);
c.setFontAndSize (bf, 10);
c.showText (ape_nom);
c.endText ();

//.....8
PdfContentByte ra = writer.getDirectContent ();
ra.beginText ();
ra.setTextMatrix (120, 550);
ra.setFontAndSize (bf, 10);
ra.showText (lugar);
ra.endText ();

//.....9
//.....10
PdfContentByte f = writer.getDirectContent ();
f.beginText ();
f.setTextMatrix (420, 550);
f.setFontAndSize (bf, 10);
f.showText (denom);
f.endText ();

//.....11
PdfContentByte g = writer.getDirectContent ();
g.beginText ();
g.setTextMatrix (120, 522);
g.setFontAndSize (bf, 10);
g.showText (direc);
g.endText ();

//.....12
PdfContentByte h = writer.getDirectContent ();
h.beginText ();
h.setTextMatrix (393, 522);
h.setFontAndSize (bf, 10);
h.showText (num);
h.endText ();

//.....13
PdfContentByte ia = writer.getDirectContent ();
ia.beginText ();
ia.setTextMatrix (464, 522);
ia.setFontAndSize (bf, 10);
ia.showText (manz);
ia.endText ();

//.....14
PdfContentByte j = writer.getDirectContent ();

```

```

j.beginText ();
j.setTextMatrix (540, 522);
j.setFontAndSize (bf, 10);
j.showText (lote);
j.endText ();

//.....15
PdfContentByte k = writer.getDirectContent ();
k.beginText ();
k.setTextMatrix (145, 460);
k.setFontAndSize (bf, 10);
k.showText ("SOCIEDAD CONYUGAL");
k.endText ();

//.....16
PdfContentByte l = writer.getDirectContent ();
l.beginText ();
l.setTextMatrix (90, 419);
l.setFontAndSize (bf, 10);
l.showText ("TERMINADO");
l.endText ();

PdfContentByte pa = writer.getDirectContent ();
pa.beginText ();
pa.setTextMatrix (265, 419);
pa.setFontAndSize (bf, 10);
pa.showText ("PREDIO TOTAL");
pa.endText ();

Predio dj = ls.get (0);

PdfContentByte ta = writer.getDirectContent ();
ta.beginText ();
ta.setTextMatrix (415, 419);
ta.setFontAndSize (bf, 10);
ta.showText (dj.getHabitacion ().getDescripcion ());
ta.endText ();

PdfContentByte w = writer.getDirectContent ();
w.beginText ();
w.setTextMatrix (40, 340);
w.setFontAndSize (bf, 10);
w.showText (dj.getHabitacion ().getDescripcion ());
w.endText ();

//.....17
PdfContentByte ab = writer.getDirectContent ();
ab.beginText ();
ab.setTextMatrix (230, 340);
ab.setFontAndSize (bf, 10);

ab.showText (dj.getMatpredominante ().getDescripcion ());
ab.endText ();

PdfContentByte ac = writer.getDirectContent ();
ac.beginText ();
ac.setTextMatrix (450, 340);
ac.setFontAndSize (bf, 10);
ac.showText (dj.getConservacion ().getDescripcion ());

```

```

ac.endText ();

int y = 255;
for (Predio l1 : ls) {

    PdfContentByte ax = writer.getDirectContent ();
    ax.beginText ();
    ax.setTextMatrix(24, y);
    ax.setFontAndSize(bf, 10);
    ax.showText(l1.getBloque ());
    ax.endText ();

    PdfContentByte af = writer.getDirectContent ();
    af.beginText ();
    af.setTextMatrix(43, y);
    af.setFontAndSize(bf, 10);
    af.showText(l1.getPiso_1 ());
    af.endText ();

    PdfContentByte ay = writer.getDirectContent ();
    ay.beginText ();
    ay.setTextMatrix(64, y);
    ay.setFontAndSize(bf, 10);
    ay.showText(l1.getMaterialmuro ().getLetra ());
    ay.endText ();

    PdfContentByte ayi = writer.getDirectContent ();
    ayi.beginText ();
    ayi.setTextMatrix(77, y);
    ayi.setFontAndSize(bf, 10);
    ayi.showText(l1.getTecho ().getLetra ());
    ayi.endText ();

    PdfContentByte ad = writer.getDirectContent ();
    ad.beginText ();
    ad.setTextMatrix(89, y);
    ad.setFontAndSize(bf, 10);
    ad.showText(l1.getPiso ().getLetra ());
    ad.endText ();

    PdfContentByte pe = writer.getDirectContent ();
    pe.beginText ();
    pe.setTextMatrix(103, y);
    pe.setFontAndSize(bf, 10);
    pe.showText(l1.getPuerta ().getLetra ());
    pe.endText ();

    PdfContentByte pi = writer.getDirectContent ();
    pi.beginText ();
    pi.setTextMatrix(116, y);
    pi.setFontAndSize(bf, 10);
    pi.showText(l1.getRevestimiento ().getLetra ());
    pi.endText ();

    PdfContentByte oe = writer.getDirectContent ();
    oe.beginText ();
    oe.setTextMatrix(129, y);
    oe.setFontAndSize(bf, 10);
    oe.showText(l1.getBano ().getLetra ());

```

```

oe.endText ();

PdfContentByte ji = writer.getDirectContent ();
ji.beginText ();
ji.setTextMatrix(142, y);
ji.setFontAndSize(bf, 10);
ji.showText(l1.getInstalaciones().getLetra());
ji.endText ();

PdfContentByte ah = writer.getDirectContent ();
ah.beginText ();
ah.setTextMatrix(172, y);
ah.setFontAndSize(bf, 10);
ah.showText(String.valueOf(l1.getValorconst()));
ah.endText ();

PdfContentByte ae = writer.getDirectContent ();
ae.beginText ();
ae.setTextMatrix(215, y);
ae.setFontAndSize(bf, 10);
ae.showText("0.00");
ae.endText ();

BigDecimal vd =
l1.getDeprec().multiply(l1.getValorconst()).divide(new
BigDecimal(100)).setScale(2, RoundingMode.CEILING);

PdfContentByte bi = writer.getDirectContent ();
bi.beginText ();
bi.setTextMatrix(255, y);
bi.setFontAndSize(bf, 10);
bi.showText(String.valueOf(vd));
bi.endText ();

PdfContentByte ap = writer.getDirectContent ();
ap.beginText ();
ap.setTextMatrix(310, y);
ap.setFontAndSize(bf, 10);
ap.showText(String.valueOf(l1.getDeprec()));
ap.endText ();

PdfContentByte an = writer.getDirectContent ();
an.beginText ();
an.setTextMatrix(360, y);
an.setFontAndSize(bf, 10);
an.showText(String.valueOf(l1.getArea()));
an.endText ();

BigDecimal vt =
l1.getArea().multiply(l1.getValorconst()).setScale(2,
RoundingMode.CEILING);

PdfContentByte av = writer.getDirectContent ();
av.beginText ();
av.setTextMatrix(420, y);
av.setFontAndSize(bf, 10);
av.showText(String.valueOf(vt));
av.endText ();

```



```

        PdfContentByte aj = writer.getDirectContent ();
        aj.beginText ();
        aj.setTextMatrix(470, y);
        aj.setFontAndSize(bf, 10);
        aj.showText("0.00");
        aj.endText ();

        BigDecimal vc =
vt.subtract(vd.multiply(l1.getArea()).setScale(2,
RoundingMode.CEILING));

        PdfContentByte al = writer.getDirectContent ();
        al.beginText ();
        al.setTextMatrix(530, y);
        al.setFontAndSize(bf, 10);
        al.showText(String.valueOf(vc));
        al.endText ();

        y -= 16;
    }

    PdfContentByte ao = writer.getDirectContent ();
    ao.beginText ();
    ao.setTextMatrix(155, 115);
    ao.setFontAndSize(bf, 10);
    ao.showText(area_terreno);
    ao.endText ();

    PdfContentByte ai = writer.getDirectContent ();
    ai.beginText ();
    ai.setTextMatrix(290, 115);
    ai.setFontAndSize(bf, 10);
    ai.showText(arancel);
    ai.endText ();

    BigDecimal nb = new
BigDecimal(valortotal_const).subtract(new
BigDecimal(valortotal_terren));

    PdfContentByte tu = writer.getDirectContent ();
    tu.beginText ();
    tu.setTextMatrix(520, 115);
    tu.setFontAndSize(bf, 10);
    tu.showText(String.valueOf(nb));
    tu.endText ();

    PdfContentByte be = writer.getDirectContent ();
    be.beginText ();
    be.setTextMatrix(520, 85);
    be.setFontAndSize(bf, 10);
    be.showText("0.00");
    be.endText ();

    PdfContentByte bo = writer.getDirectContent ();
    bo.beginText ();
    bo.setTextMatrix(520, 58);
    bo.setFontAndSize(bf, 10);
    bo.showText(valortotal_terren);
    bo.endText ();

```

```

        PdfContentByte ci = writer.getDirectContent ();
        ci.beginText ();
        ci.setTextMatrix(520, 32);
        ci.setFontAndSize(bf, 10);
        ci.showText(total_autovalor);
        ci.endText ();

        PdfContentByte di = writer.getDirectContent ();
        di.beginText ();
        di.setTextMatrix(70, 35);
        di.setFontAndSize(bf, 10);
        di.showText(regimen);
        di.endText ();

        PdfContentByte da = writer.getDirectContent ();
        da.beginText ();
        da.setTextMatrix(400, 5);
        da.setFontAndSize(bf, 10);
        da.showText(dia);
        da.endText ();

        PdfContentByte du = writer.getDirectContent ();
        du.beginText ();
        du.setTextMatrix(420, 5);
        du.setFontAndSize(bf, 10);
        du.showText(mes);
        du.endText ();

        PdfContentByte fi = writer.getDirectContent ();
        fi.beginText ();
        fi.setTextMatrix(470, 5);
        fi.setFontAndSize(bf, 10);
        fi.showText(añ);
        fi.endText ();
        document.close ();

    } catch (FileNotFoundException ex) {
        Logger.getLogger(IDJ.class.getName()).log(Level.SEVERE, null,
ex);
    } catch (DocumentException ex) {
        Logger.getLogger(IDJ.class.getName()).log(Level.SEVERE, null,
ex);
    } catch (IOException ex) {
        Logger.getLogger(IDJ.class.getName()).log(Level.SEVERE, null,
ex);
    }
    return 1;
}

public int a4(int a) {
    return 842 - a;
}

public String t8(String n) {
    String a = n;

```

```

        while (a.length() < 8) {
            a = "0" + a;
        }
        return a;
    }
}

```

IDJurada

```

package control;

public class IDJurada {

    String codigo, ano, n_ddjj, n_dni_ruc, ape_nom,
        lugar, denom, direc, num, manz,
        lote, area_terreno, arancel, valortotal_const,
        valortotal_terren, total_autovaluo, regimen, dia,
mes, añ;

    public IDJurada(String codigo, String ano, String n_ddjj,
String n_dni_ruc, String ape_nom, String lugar, String denom,
String direc, String num, String manz, String lote, String
valortotal_const, String area_terreno, String arancel, String
valortotal_terren, String total_autovaluo, String regimen,
String dia, String mes, String añ) {
        this.codigo = codigo;
        this.ano = ano;
        this.n_ddjj = n_ddjj;
        this.n_dni_ruc = n_dni_ruc;
        this.ape_nom = ape_nom;
        this.lugar = lugar;
        this.denom = denom;
        this.direc = direc;
        this.num = num;
        this.manz = manz;
        this.lote = lote;

        this.valortotal_const = valortotal_const;
        this.area_terreno = area_terreno;
        this.arancel = arancel;
        this.valortotal_terren = valortotal_terren;
        this.total_autovaluo = total_autovaluo;

        this.regimen = regimen;
        this.dia = dia;
        this.mes = mes;
        this.añ = añ;
    }

    public String getCodigo() {
        return codigo;
    }

    public void setCodigo(String codigo) {
        this.codigo = codigo;
    }
}

```

```

public String getAno() {
    return ano;
}

public void setAno(String ano) {
    this.ano = ano;
}

public String getN_ddjj() {
    return n_ddjj;
}

public void setN_ddjj(String n_ddjj) {
    this.n_ddjj = n_ddjj;
}

public String getN_dni_ruc() {
    return n_dni_ruc;
}

public void setN_dni_ruc(String n_dni_ruc) {
    this.n_dni_ruc = n_dni_ruc;
}

public String getApe_nom() {
    return ape_nom;
}

public void setApe_nom(String ape_nom) {
    this.ape_nom = ape_nom;
}

public String getLugar() {
    return lugar;
}

public void setLugar(String lugar) {
    this.lugar = lugar;
}

public String getDenom() {
    return denom;
}

public void setDenom(String denom) {
    this.denom = denom;
}

public String getDirec() {
    return direc;
}

public void setDirec(String direc) {
    this.direc = direc;
}

public String getNum() {
    return num;
}

```

```

}

public void setNum(String num) {
    this.num = num;
}

public String getManz() {
    return manz;
}

public void setManz(String manz) {
    this.manz = manz;
}

public String getLote() {
    return lote;
}

public void setLote(String lote) {
    this.lote = lote;
}

public String getArea_terreno() {
    return area_terreno;
}

public void setArea_terreno(String area_terreno) {
    this.area_terreno = area_terreno;
}

public String getArancel() {
    return arancel;
}

public void setArancel(String arancel) {
    this.arancel = arancel;
}

public String getValortotal_const() {
    return valortotal_const;
}

public void setValortotal_const(String valortotal_const) {
    this.valortotal_const = valortotal_const;
}

public String getValortotal_terren() {
    return valortotal_terren;
}

public void setValortotal_terren(String valortotal_terren) {
    this.valortotal_terren = valortotal_terren;
}

public String getTotal_autovaluo() {
    return total_autovaluo;
}

public void setTotal_autovaluo(String total_autovaluo) {

```

```

        this.total_autovaluo = total_autovaluo;
    }

    public String getRegimen() {
        return regimen;
    }

    public void setRegimen(String regimen) {
        this.regimen = regimen;
    }

    public String getDia() {
        return dia;
    }

    public void setDia(String dia) {
        this.dia = dia;
    }

    public String getMes() {
        return mes;
    }

    public void setMes(String mes) {
        this.mes = mes;
    }

    public String getAñ() {
        return añ;
    }

    public void setAñ(String añ) {
        this.añ = añ;
    }
}

```

class Index

```

/*
 * To change this license header, choose License Headers in
 * Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package control;

import java.text.SimpleDateFormat;
import java.util.Timer;
import java.util.TimerTask;
import javafx.application.Application;
import static javafx.application.Application.launch;
import javafx.fxml.FXMLLoader;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.stage.Stage;

```

```

import javafx.stage.StageStyle;
import modelo.HibernateUtil;
import org.hibernate.Session;

/**
 *
 * @author Manrrique
 */
public class Index extends Application {

    static SimpleDateFormat fo = new
SimpleDateFormat("dd/MM/yyyy");
    static SimpleDateFormat fi = new SimpleDateFormat("yyyy-MM-
dd");

    @Override
    public void start(Stage stage) throws Exception {

        Parent root =
FXMLLoader.load(getClass().getResource("/view/VLogin.fxml"));
        Scene scene = new Scene(root);
        stage.setScene(scene);
        stage.initStyle(StageStyle.UNDECORATED);
        stage.show();
    }
//content.getChildren().setAll(FXMLLoader.load("vista2.fxml"));

    public static void main(String[] args) {
        launch(args);
    }
}

```

class IPU

```

/*
 * To change this license header, choose License Headers in
Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package control;

import com.itextpdf.text.Document;
import com.itextpdf.text.DocumentException;
import com.itextpdf.text.Image;
import com.itextpdf.text.pdf.BaseFont;
import com.itextpdf.text.pdf.PdfContentByte;
import com.itextpdf.text.pdf.PdfWriter;
import static control.Index.fo;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import java.math.BigDecimal;
import java.math.RoundingMode;
import java.util.logging.Level;
import java.util.logging.Logger;
import modelo.Caja;

```

```

import modelo.Propiedad;
import modelo.Propietario;

/**
 *
 * @author Manrrique
 */
public class IPU {

    public int IP(Caja c, Propietario pr, Propiedad pd) {
        int i = createPdf(new IPUrbano(
            String.valueOf(c.getAno()),
            pr.getNombre(),
            pr.getNumdoc(),
            fo.format(c.getFecha()),
            pd.getDireccion(),
            String.valueOf(c.getUit()),
            String.valueOf(c.getIdCaja()),
            "1",
            "0",
            String.valueOf(c.getAno()),
            String.valueOf(c.getBase()),
            String.valueOf(c.getMonto())
        ));
        return i;
    }

    public int createPdf(IPUrbano pu) {
        String pdf = "D://Autovaluo/predio.pdf";
        String img = "D://Autovaluo/Resources/Auto1.jpg";
        String num = pu.getFnac(),
            ano = pu.getAno(),
            nom = pu.getNom(),
            dni = pu.getDni(),
            fecha = pu.getFecha(),
            dir = pu.getDir(),
            uit = pu.getUit(),
            anod = pu.getAnod(),
            der = pu.getDer(),
            anos = pu.getAnos(),
            basei = pu.getBasei(),
            iannual = pu.getIannual();
        BigDecimal vtri = new BigDecimal(iannual).divide(new
        BigDecimal(4)).setScale(2, RoundingMode.CEILING);

        try {
            Document document = new Document();
            PdfWriter writer = PdfWriter.getInstance(document,
            new FileOutputStream(pdf));
            document.open();
            Image imagen = Image.getInstance(img);
            imagen.scaleAbsoluteHeight(842);
            imagen.scaleAbsoluteWidth(595);
            imagen.setAbsolutePosition(0, 0);
            document.add(imagen);

            BaseFont bf = BaseFont.createFont();

            PdfContentByte n0 = writer.getDirectContent();

```



```

n0.beginText ();
n0.setTextMatrix(460, a4(118));
n0.setFontAndSize(bf, 10);
n0.showText(num);
n0.endText ();

PdfContentByte n1 = writer.getDirectContent ();
n1.beginText ();
n1.setTextMatrix(277, a4(129));
n1.setFontAndSize(bf, 10);
n1.showText(ano);
n1.endText ();

PdfContentByte n2 = writer.getDirectContent ();
n2.beginText ();
n2.setTextMatrix(54, a4(167));
n2.setFontAndSize(bf, 10);
n2.showText(nom);
n2.endText ();

PdfContentByte n3 = writer.getDirectContent ();
n3.beginText ();
n3.setTextMatrix(373, a4(167));
n3.setFontAndSize(bf, 10);
n3.showText(dni);
n3.endText ();

PdfContentByte n4 = writer.getDirectContent ();
n4.beginText ();
n4.setTextMatrix(480, a4(167));
n4.setFontAndSize(bf, 10);
n4.showText(fecha);
n4.endText ();

PdfContentByte n5 = writer.getDirectContent ();
n5.beginText ();
n5.setTextMatrix(192, a4(190));
n5.setFontAndSize(bf, 10);
n5.showText(dir);
n5.endText ();

PdfContentByte n6 = writer.getDirectContent ();
n6.beginText ();
n6.setTextMatrix(195, a4(213));
n6.setFontAndSize(bf, 10);
n6.showText(uit);
n6.endText ();

PdfContentByte n7 = writer.getDirectContent ();
n7.beginText ();
n7.setTextMatrix(388, a4(213));
n7.setFontAndSize(bf, 10);
n7.showText(anod);
n7.endText ();

PdfContentByte n8 = writer.getDirectContent ();
n8.beginText ();
n8.setTextMatrix(195, a4(236));
n8.setFontAndSize(bf, 10);

```

```

n8.showText (der);
n8.endText ();

PdfContentByte n9 = writer.getDirectContent ();
n9.beginText ();
n9.setTextMatrix (388, a4 (237));
n9.setFontAndSize (bf, 10);
n9.showText (anos);
n9.endText ();

PdfContentByte n10 = writer.getDirectContent ();
n10.beginText ();
n10.setTextMatrix (144, a4 (272));
n10.setFontAndSize (bf, 10);
n10.showText (basei);
n10.endText ();

PdfContentByte n11 = writer.getDirectContent ();
n11.beginText ();
n11.setTextMatrix (292, a4 (272));
n11.setFontAndSize (bf, 10);
n11.showText (ianual);
n11.endText ();

PdfContentByte n12 = writer.getDirectContent ();
n12.beginText ();
n12.setTextMatrix (387, a4 (272));
n12.setFontAndSize (bf, 10);
n12.showText (String.valueOf (vtri));
n12.endText ();

int y = a4 (311);
for (int i = 0; i < 4; i++) {

    PdfContentByte n13 = writer.getDirectContent ();
    n13.beginText ();
    n13.setTextMatrix (155, y);
    n13.setFontAndSize (bf, 10);
    n13.showText (String.valueOf (vtri));
    n13.endText ();

    PdfContentByte n14 = writer.getDirectContent ();
    n14.beginText ();
    n14.setTextMatrix (249, y);
    n14.setFontAndSize (bf, 10);
    n14.showText ("0");
    n14.endText ();

    PdfContentByte n15 = writer.getDirectContent ();
    n15.beginText ();
    n15.setTextMatrix (328, y);
    n15.setFontAndSize (bf, 10);
    n15.showText (String.valueOf (vtri));
    n15.endText ();

    PdfContentByte n16 = writer.getDirectContent ();
    n16.beginText ();
    n16.setTextMatrix (404, y);
    n16.setFontAndSize (bf, 10);

```

```

        n16.showText("X");
        n16.endText();

        PdfContentByte n17 = writer.getDirectContent();
        n17.beginText();
        n17.setTextMatrix(430, y);
        n17.setFontAndSize(bf, 10);
        n17.showText("Pagar");
        n17.endText();

        PdfContentByte n18 = writer.getDirectContent();
        n18.beginText();
        n18.setTextMatrix(490, y);
        n18.setFontAndSize(bf, 10);
        n18.showText("A");
        n18.endText();

        y -= 22;
    }
    PdfContentByte n19 = writer.getDirectContent();
    n19.beginText();
    n19.setTextMatrix(328, a4(400));
    n19.setFontAndSize(bf, 10);
    n19.showText(ianual);
    n19.endText();
    Numeros ns = new Numeros();
    PdfContentByte n20 = writer.getDirectContent();
    n20.beginText();
    n20.setTextMatrix(86, 421);
    n20.setFontAndSize(bf, 10);
    n20.showText(ns.Convertir(ianual, true));
    n20.endText();
    document.close();
} catch (FileNotFoundException ex) {

Logger.getLogger(IPU.class.getName()).log(Level.SEVERE, null,
ex);
    } catch (DocumentException ex) {

Logger.getLogger(IPU.class.getName()).log(Level.SEVERE, null,
ex);
    } catch (IOException ex) {

Logger.getLogger(IPU.class.getName()).log(Level.SEVERE, null,
ex);
    }
    return 1;
}
public int a4(int a) {
    return 842 - a;
}
}

```

class IPUrbano

```

/*
 * To change this license header, choose License Headers in
 * Project Properties.

```

```

* To change this template file, choose Tools | Templates
* and open the template in the editor.
*/
package control;
/**
 *
 * @author Manrrique
 */
public class IPUrbano {

    String ano,
           nom,
           dni,
           fecha,
           dir,
           uit,
           fnac,
           anod,
           der,
           anos,
           basei,
           ianual;

    public IPUrbano(String ano, String nom, String dni, String
fecha, String dir, String uit, String fnac, String anod, String
der, String anos, String basei, String ianual) {
        this.ano = ano;
        this.nom = nom;
        this.dni = dni;
        this.fecha = fecha;
        this.dir = dir;
        this.uit = uit;
        this.fnac = fnac;
        this.anod = anod;
        this.der = der;
        this.anos = anos;
        this.basei = basei;
        this.ianual = ianual;
    }
    public String getAno() {
        return ano;
    }
    public void setAno(String ano) {
        this.ano = ano;
    }
    public String getNom() {
        return nom;
    }
    public void setNom(String nom) {
        this.nom = nom;
    }
    public String getDni() {
        return dni;
    }
    public void setDni(String dni) {
        this.dni = dni;
    }
    public String getFecha() {
        return fecha;
    }
}

```

```

public void setFecha(String fecha) {
    this.fecha = fecha;
}
public String getDir() {
    return dir;
}
public void setDir(String dir) {
    this.dir = dir;
}
public String getUit() {
    return uit;
}
public void setUit(String uit) {
    this.uit = uit;
}
public String getFnac() {
    return fnac;
}
public void setFnac(String fnac) {
    this.fnac = fnac;
}
public String getAnod() {
    return anod;
}
public void setAnod(String anod) {
    this.anod = anod;
}
public String getDer() {
    return der;
}
public void setDer(String der) {
    this.der = der;
}
public String getAnos() {
    return anos;
}
public void setAnos(String anos) {
    this.anos = anos;
}
public String getBasei() {
    return basei;
}
public void setBasei(String basei) {
    this.basei = basei;
}

public String getIannual() {
    return iannual;
}

public void setIannual(String iannual) {
    this.ianual = iannual;
}
}

```

class Numeros

```
/*
 * To change this license header, choose License Headers in
 * Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package control;
/**
 *
 * @author Manrrique
 */
import java.util.regex.Pattern;
public class Numeros {

    private final String[] UNIDADES = {"", "un ", "dos ", "tres
", "cuatro ", "cinco ", "seis ", "siete ", "ocho ", "nueve "};
    private final String[] DECENAS = {"diez ", "once ", "doce ",
"trece ", "catorce ", "quince ", "dieciseis ",
    "diecisiete ", "dieciocho ", "diecinueve", "veinte ",
"treinta ", "cuarenta ",
    "cincuenta ", "sesenta ", "setenta ", "ochenta ",
"noventa "};
    private final String[] CENTENAS = {"", "ciento ",
"doscientos ", "trecientos ", "cuatrocientos ", "quinientos ",
"seiscientos ",
    "setecientos ", "ochocientos ", "novecientos "};

    public Numeros() {
    }
    public String Convertir(String numero, boolean mayusculas) {
        String literal = "";
        String parte_decimal;
        numero = numero.replace(".", ",");
        if (numero.indexOf(",") == -1) {
            numero = numero + ",00";
        }
        if (Pattern.matches("\\d{1,9},\\d{1,2}", numero)) {
            String Num[] = numero.split(",");
            parte_decimal = Num[1] + "/100 Soles.";
            if (Integer.parseInt(Num[0]) == 0) {
                literal = "cero ";
            } else if (Integer.parseInt(Num[0]) > 999999) {
                literal = getMillones(Num[0]);
            } else if (Integer.parseInt(Num[0]) > 999) {
                literal = getMiles(Num[0]);
            } else if (Integer.parseInt(Num[0]) > 99) {
                literal = getCentenas(Num[0]);
            } else if (Integer.parseInt(Num[0]) > 9) {
                literal = getDecenas(Num[0]);
            } else {
                literal = getUnidades(Num[0]);
            }
        }
        if (mayusculas) {
            return (literal + parte_decimal).toUpperCase();
        } else {
            return (literal + parte_decimal);
        }
    }
}
```

```

    } else {
        return literal = null;
    }
}
private String getUnidades(String numero) {
    String num = numero.substring(numero.length() - 1);
    return UNIDADES[Integer.parseInt(num)];
}

private String getDecenas(String num) {
    int n = Integer.parseInt(num);
    if (n < 10) {
        return getUnidades(num);
    } else if (n > 19) {
        String u = getUnidades(num);
        if (u.equals("")) {
            return DECENAS[Integer.parseInt(num.substring(0,
1)) + 8];
        } else {
            return DECENAS[Integer.parseInt(num.substring(0,
1)) + 8] + "y " + u;
        }
    } else {
        return DECENAS[n - 10];
    }
}
private String getCentenas(String num) {
    if (Integer.parseInt(num) > 99) {
        if (Integer.parseInt(num) == 100) {
            return " cien ";
        } else {
            return
CENTENAS[Integer.parseInt(num.substring(0, 1))] +
getDecenas(num.substring(1));
        }
    } else {
        return getDecenas(Integer.parseInt(num) + "");
    }
}
private String getMiles(String numero) {
    String c = numero.substring(numero.length() - 3);
    String m = numero.substring(0, numero.length() - 3);
    String n = "";
    if (Integer.parseInt(m) > 0) {
        n = getCentenas(m);
        return n + "mil " + getCentenas(c);
    } else {
        return "" + getCentenas(c);
    }
}
private String getMillones(String numero) {
    String miles = numero.substring(numero.length() - 6);
    String millon = numero.substring(0, numero.length() -
6);
    String n = "";
    if (millon.length() > 1) {
        n = getCentenas(millon) + "millones ";
    } else {

```

```

        n = getUnidades(millon) + "millon ";
    }
    return n + getMiles(miles);
}
}

```

b. DAO

class DaoConservacion

```

package dao;
import java.util.ArrayList;
import java.util.List;
import modelo.Conservacion;
import modelo.HibernateUtil;
import org.hibernate.Session;
import org.hibernate.criterion.Restrictions;
public class DaoConservacion {
    public List<Conservacion> all() {
        List<Conservacion> l = new ArrayList();
        l = (List)
HibernateUtil.getSession().createCriteria(Conservacion.class).list();
        return l;
    }
    public Conservacion data(int idconservacion) {
        Conservacion rg = (Conservacion)
HibernateUtil.getSession().createCriteria(Conservacion.class)
            .add(Restrictions.and(
                Restrictions.eq("idconservacion",
idconservacion)))
            .setMaxResults(1)
            .uniqueResult();
        return rg;
    }
    public void save(Conservacion reg) {
        Session s =
HibernateUtil.getSessionFactory().getCurrentSession();
        s.beginTransaction();
        s.saveOrUpdate(reg);
        s.getTransaction().commit();
    }
    public void eliminar(Conservacion reg) {
        Session s =
HibernateUtil.getSessionFactory().getCurrentSession();
        s.beginTransaction();
        s.delete(reg);
        s.getTransaction().commit();
    }
}

```

class DaoDepreciacion

```

package dao;

```



```

import java.util.ArrayList;
import java.util.List;
import modelo.Depreciacion;
import modelo.HibernateUtil;
import org.hibernate.Session;
import org.hibernate.criterion.Restrictions;
public class DaoDepreciacion {
    public List<Depreciacion> all() {
        List<Depreciacion> l = new ArrayList();
        l = (List)
HibernateUtil.getSession().createCriteria(Depreciacion.class).list();
        return l;
    }
    public Depreciacion data(int idhabitacion, int antiguedad,
int material) {
        Depreciacion rg = (Depreciacion)
HibernateUtil.getSession().createCriteria(Depreciacion.class)
        .add(Restrictions.and(
            Restrictions.eq("idhabitacion",
idhabitacion),
            Restrictions.eq("antiguedad",
antiguedad),
            Restrictions.eq("material", material)))
        .setMaxResults(1)
        .uniqueResult();
        return rg;
    }
    public void save(Depreciacion reg) {
        Session s =
HibernateUtil.getSessionFactory().getCurrentSession();
        s.beginTransaction();
        s.saveOrUpdate(reg);
        s.getTransaction().commit();
    }
    public void eliminar(Depreciacion reg) {
        Session s =
HibernateUtil.getSessionFactory().getCurrentSession();
        s.beginTransaction();
        s.delete(reg);
        s.getTransaction().commit();
    }
}

```

DaoFiltrar

```

package dao;
import java.util.ArrayList;
import java.util.List;
import modelo.HibernateUtil;
import modelo.Predio;
import modelo.Propiedad;
import modelo.Propietario;
import org.hibernate.Session;
import org.hibernate.criterion.Restrictions;
public class DaoFiltrar {
    public List all() {
        List l = new ArrayList();

```

```

        l = (List)
HibernateUtil.getSession().createCriteria(Propiedad.class).list(
);
        return l;
    }
    public Propiedad data(int idprop) {
        Propiedad rg = (Propiedad)
HibernateUtil.getSession().createCriteria(Propiedad.class)
        .add(Restrictions.eq("idprop", idprop))
        .setMaxResults(1)
        .uniqueResult();
        return rg;
    }
    public Predio año(String ano) {
        Predio rg = (Predio)
HibernateUtil.getSession().createCriteria(Predio.class)
        .add(Restrictions.like("ano", "%"+ano+"%"))
        .setMaxResults(1)
        .uniqueResult();
        return rg;
    }
    public void save(Propietario reg) {
        Session s =
HibernateUtil.getSessionFactory().getCurrentSession();
        s.beginTransaction();
        s.saveOrUpdate(reg);
        s.getTransaction().commit();
    }
    public void eliminar(Propietario reg) {
        Session s =
HibernateUtil.getSessionFactory().getCurrentSession();
        s.beginTransaction();
        s.delete(reg);
        s.getTransaction().commit();
    }
}

```

class DaoHabitacion

```

package dao;
import java.util.ArrayList;
import java.util.List;
import modelo.Habitacion;
import modelo.HibernateUtil;
import org.hibernate.Session;
import org.hibernate.criterion.Restrictions;
public class DaoHabitacion {
    public List<Habitacion> all() {
        List<Habitacion> l = new ArrayList();
        l = (List)
HibernateUtil.getSession().createCriteria(Habitacion.class).list
();
        return l;
    }
    public Habitacion data(int idhabitacion) {
        Habitacion rg = (Habitacion)
HibernateUtil.getSession().createCriteria(Habitacion.class)
        .add(Restrictions.and(

```

```

Restrictions.eq("idhabitacion",
idhabitacion)))
        .setMaxResults(1)
        .uniqueResult();
    return rg;
}
public void save(Habitacion reg) {
    Session s =
HibernateUtil.getSessionFactory().getCurrentSession();
    s.beginTransaction();
    s.saveOrUpdate(reg);
    s.getTransaction().commit();
}
public void eliminar(Habitacion reg) {
    Session s =
HibernateUtil.getSessionFactory().getCurrentSession();
    s.beginTransaction();
    s.delete(reg);
    s.getTransaction().commit();
}
}
}

```

class DaoInstalaciones

```

package dao;
import java.util.ArrayList;
import java.util.List;
import modelo.HibernateUtil;
import modelo.Instalaciones;
import org.hibernate.Session;
import org.hibernate.criterion.Restrictions;
public class DaoInstalaciones {

    public List<Instalaciones> all() {
        List<Instalaciones> l = new ArrayList();
        l = (List)
HibernateUtil.getSession().createCriteria(Instalaciones.class).list();
        return l;
    }
    public Instalaciones data(String letra) {
        Instalaciones rg = (Instalaciones)
HibernateUtil.getSession().createCriteria(Instalaciones.class)
            .add(Restrictions.and(
                Restrictions.eq("letra", letra)))
            .setMaxResults(1)
            .uniqueResult());
        return rg;
    }
    public void save(Instalaciones reg) {
        Session s =
HibernateUtil.getSessionFactory().getCurrentSession();
        s.beginTransaction();
        s.saveOrUpdate(reg);
        s.getTransaction().commit();
    }
    public void eliminar(Instalaciones reg) {

```

```

        Session s =
HibernateUtil.getSessionFactory().getCurrentSession();
        s.beginTransaction();
        s.delete(reg);
        s.getTransaction().commit();
    }
}

```

class DaoMaterialMuro

```

package dao;
import java.util.ArrayList;
import java.util.List;
import modelo.HibernateUtil;
import modelo.Materialmuro;
import org.hibernate.Session;
import org.hibernate.criterion.Restrictions;
public class DaoMaterialMuro {
    public List<Materialmuro> all() {
        List<Materialmuro> l = new ArrayList();
        l = (List)
HibernateUtil.getSession().createCriteria(Materialmuro.class).list();
        return l;
    }
    public Materialmuro data(String letra) {
        Materialmuro rg = (Materialmuro)
HibernateUtil.getSession().createCriteria(Materialmuro.class)
            .add(Restrictions.and(
                Restrictions.eq("letra", letra)))
            .setMaxResults(1)
            .uniqueResult();
        return rg;
    }
    public void save(Materialmuro reg) {
        Session s =
HibernateUtil.getSessionFactory().getCurrentSession();
        s.beginTransaction();
        s.saveOrUpdate(reg);
        s.getTransaction().commit();
    }
    public void eliminar(Materialmuro reg) {
        Session s =
HibernateUtil.getSessionFactory().getCurrentSession();
        s.beginTransaction();
        s.delete(reg);
        s.getTransaction().commit();
    }
}

```

class DaoMatpredominante

```

package dao;
import java.util.ArrayList;
import java.util.List;
import modelo.HibernateUtil;

```

```

import modelo.Matpredominante;
import org.hibernate.Session;
import org.hibernate.criterion.Restrictions;
public class DaoMatpredominante {

    public List<Matpredominante> all() {
        List<Matpredominante> l = new ArrayList();
        l = (List)
HibernateUtil.getSession().createCriteria(Matpredominante.class)
.list();
        return l;
    }
    public Matpredominante data(int idmatpredominante) {
        Matpredominante rg = (Matpredominante)
HibernateUtil.getSession().createCriteria(Matpredominante.class)
.add(Restrictions.and(
        Restrictions.eq("idmatpredominante",
idmatpredominante)))
        .setMaxResults(1)
        .uniqueResult();
        return rg;
    }
    public void save(Matpredominante reg) {
        Session s =
HibernateUtil.getSessionFactory().getCurrentSession();
        s.beginTransaction();
        s.saveOrUpdate(reg);
        s.getTransaction().commit();
    }
    public void eliminar(Matpredominante reg) {
        Session s =
HibernateUtil.getSessionFactory().getCurrentSession();
        s.beginTransaction();
        s.delete(reg);
        s.getTransaction().commit();
    }
}

```

class DaoPiso

```

/*
 * To change this license header, choose License Headers in
Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package dao;
import java.util.ArrayList;
import java.util.List;
import modelo.HibernateUtil;
import modelo.Piso;
import org.hibernate.Session;
import org.hibernate.criterion.Restrictions;
/**
 *
 * @author PABEL
 */
public class DaoPiso {

```

```

    public List<Piso> all() {
        List<Piso> l = new ArrayList();
        l = (List)
HibernateUtil.getSession().createCriteria(Piso.class).list();
        return l;
    }
    public Piso data(String letra) {
        Piso rg = (Piso)
HibernateUtil.getSession().createCriteria(Piso.class)
        .add(Restrictions.and(
            Restrictions.eq("letra", letra)))
        .setMaxResults(1)
        .uniqueResult();
        return rg;
    }
    public void save(Piso reg) {
        Session s =
HibernateUtil.getSessionFactory().getCurrentSession();
        s.beginTransaction();
        s.saveOrUpdate(reg);
        s.getTransaction().commit();
    }
    public void eliminar(Piso reg) {
        Session s =
HibernateUtil.getSessionFactory().getCurrentSession();
        s.beginTransaction();
        s.delete(reg);
        s.getTransaction().commit();
    }
}

```

DaoPredio

```

/*
 * To change this license header, choose License Headers in
Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package dao;
import java.util.ArrayList;
import java.util.List;
import modelo.HibernateUtil;
import modelo.Predio;
import org.hibernate.Session;
import org.hibernate.criterion.Restrictions;
/**
 *
 * @author PABEL
 */
public class DaoPredio {

    HibernateUtil hu = new HibernateUtil();

    public void save(Predio reg) {
        Session s = hu.Sesion().openSession();
        s.beginTransaction();

```

```

        s.saveOrUpdate(reg);
        s.getTransaction().commit();
        s.close();
    }
}

```

class DaoPropiedad

```

package dao;

import java.util.ArrayList;
import java.util.List;
import modelo.HibernateUtil;
import modelo.Propiedad;
import modelo.Propietario;
import org.hibernate.Session;
import org.hibernate.criterion.Restrictions;

public class DaoPropiedad {

    HibernateUtil hp = new HibernateUtil();

    public List<Propiedad> all() {
        Session s = hp.Sesion().openSession();
        List<Propiedad> l = new ArrayList();
        l = (List) s.createCriteria(Propiedad.class).list();
        s.close();
        return l;
    }

    public Propiedad data(int idprop, Propietario propietario) {
        Session s = hp.Sesion().openSession();
        Propiedad rg = (Propiedad)
s.createCriteria(Propiedad.class)
        .add(Restrictions.and(
            Restrictions.eq("idprop", idprop),
            Restrictions.eq("propietario",
propietario)))
        .setMaxResults(1)
        .uniqueResult();
        s.close();
        return rg;
    }

    public void save(Propiedad reg) {
        Session s = hp.Sesion().openSession();
        s.beginTransaction();
        s.saveOrUpdate(reg);
        s.getTransaction().commit();
        s.close();
    }

    public void eliminar(Propiedad reg) {
        Session s = hp.Sesion().openSession();
        s.beginTransaction();
        s.delete(reg);
        s.getTransaction().commit();
        s.close();
    }
}

```

```
}  
}
```

class DaoPropietario

```
/*  
 * To change this license header, choose License Headers in  
Project Properties.  
 * To change this template file, choose Tools | Templates  
 * and open the template in the editor.  
 */  
package dao;  
  
import java.util.ArrayList;  
import java.util.List;  
import modelo.HibernateUtil;  
import modelo.Propietario;  
import org.hibernate.Session;  
import org.hibernate.criterion.Restrictions;  
  
/**  
 *  
 * @author PABEL  
 */  
public class DaoPropietario {  
  
    HibernateUtil hu = new HibernateUtil();  
  
    public List<Propietario> all() {  
        Session s = hu.Sesion().openSession();  
        List<Propietario> l = new ArrayList();  
        l = (List) s.createCriteria(Propietario.class).list();  
        s.close();  
        return l;  
    }  
  
    public List<Propietario> data(String numdoc) {  
        Session s = hu.Sesion().openSession();  
        List<Propietario> rg =  
s.createCriteria(Propietario.class)  
            .add(Restrictions.eq("numdoc", numdoc))  
            .list();  
        s.close();  
        return rg;  
    }  
  
    public List<Propietario> nombre(String nom) {  
        Session s = hu.Sesion().openSession();  
        List<Propietario> rg =  
s.createCriteria(Propietario.class)  
            .add(Restrictions.like("nombre", "%" + nom +  
"%"))  
            .list();  
        s.close();  
        return rg;  
    }  
}
```



```

    public void save(Propietario reg) {
        Session s = hu.Sesion().openSession();
        s.beginTransaction();
        s.saveOrUpdate(reg);
        s.getTransaction().commit();
        s.close();
    }

    public void eliminar(Propietario reg) {
        Session s = hu.Sesion().openSession();
        s.beginTransaction();
        s.delete(reg);
        s.getTransaction().commit();
        s.close();
    }
}

```

DaoPuerta

```

/*
 * To change this license header, choose License Headers in
 * Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package dao;

import java.util.ArrayList;
import java.util.List;
import modelo.HibernateUtil;
import modelo.Puerta;
import org.hibernate.Session;
import org.hibernate.criterion.Restrictions;

/**
 *
 * @author PABEL
 */
public class DaoPuerta {

    public List<Puerta> all() {
        List<Puerta> l = new ArrayList();
        l = (List)
HibernateUtil.getSession().createCriteria(Puerta.class).list();
        return l;
    }

    public Puerta data(String letra) {
        Puerta rg = (Puerta)
HibernateUtil.getSession().createCriteria(Puerta.class)
                .add(Restrictions.and(
                    Restrictions.eq("letra", letra)))
                .setMaxResults(1)
                .uniqueResult();
        return rg;
    }
}

```

```

    public void save(Puerta reg) {
        Session s =
HibernateUtil.getSessionFactory().getCurrentSession();
        s.beginTransaction();
        s.saveOrUpdate(reg);
        s.getTransaction().commit();
    }

    public void eliminar(Puerta reg) {
        Session s =
HibernateUtil.getSessionFactory().getCurrentSession();
        s.beginTransaction();
        s.delete(reg);
        s.getTransaction().commit();
    }
}

```

class DaoRevestimiento

```

/*
 * To change this license header, choose License Headers in
Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package dao;

import java.util.ArrayList;
import java.util.List;
import modelo.HibernateUtil;
import modelo.Revestimiento;
import org.hibernate.Session;
import org.hibernate.criterion.Restrictions;

/**
 *
 * @author PABEL
 */
public class DaoRevestimiento {

    public List<Revestimiento> all() {
        List<Revestimiento> l = new ArrayList();
        l = (List)
HibernateUtil.getSession().createCriteria(Revestimiento.class).list();
        return l;
    }

    public Revestimiento data(String letra) {
        Revestimiento rg = (Revestimiento)
HibernateUtil.getSession().createCriteria(Revestimiento.class)
                .add(Restrictions.and(
                    Restrictions.eq("letra", letra)))
                .setMaxResults(1)
                .uniqueResult();
        return rg;
    }
}

```

```

    public void save(Revestimiento reg) {
        Session s =
HibernateUtil.getSessionFactory().getCurrentSession();
        s.beginTransaction();
        s.saveOrUpdate(reg);
        s.getTransaction().commit();
    }

    public void eliminar(Revestimiento reg) {
        Session s =
HibernateUtil.getSessionFactory().getCurrentSession();
        s.beginTransaction();
        s.delete(reg);
        s.getTransaction().commit();
    }
}

```

class DaoTecho

```

/*
 * To change this license header, choose License Headers in
Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package dao;
import java.util.ArrayList;
import java.util.List;
import modelo.HibernateUtil;
import modelo.Techo;
import org.hibernate.Session;
import org.hibernate.criterion.Restrictions;
/**
 *
 * @author PABEL
 */
public class DaoTecho {

    public List<Techo> all() {
        List<Techo> l = new ArrayList();
        l = (List)
HibernateUtil.getSession().createCriteria(Techo.class).list();
        return l;
    }

    public Techo data(String letra) {
        Techo rg = (Techo)
HibernateUtil.getSession().createCriteria(Techo.class)
                .add(Restrictions.and(
                        Restrictions.eq("letra", letra)))
                .setMaxResults(1)
                .uniqueResult();
        return rg;
    }

    public void save(Techo reg) {
        Session s =
HibernateUtil.getSessionFactory().getCurrentSession();
        s.beginTransaction();
        s.saveOrUpdate(reg);
    }
}

```

```

        s.getTransaction().commit();
    }
    public void eliminar(Techo reg) {
        Session s =
HibernateUtil.getSessionFactory().getCurrentSession();
        s.beginTransaction();
        s.delete(reg);
        s.getTransaction().commit();
    }
}

```

DaoValorpropiedad

```

/*
 * To change this license header, choose License Headers in
Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package dao;
import java.math.BigDecimal;
import java.util.ArrayList;
import java.util.List;
import modelo.HibernateUtil;
import modelo.Valorpropiedad;
import org.hibernate.Session;
import org.hibernate.criterion.Restrictions;
/**
 *
 * @author PABEL
 */
public class DaoValorpropiedad {

    public List<Valorpropiedad> all() {
        List<Valorpropiedad> l = new ArrayList();
        l = (List)
HibernateUtil.getSession().createCriteria(Valorpropiedad.class).
list();
        return l;
    }
    public Valorpropiedad data(int idvalorpro, int ano) {
        Valorpropiedad rg = (Valorpropiedad)
HibernateUtil.getSession().createCriteria(Valorpropiedad.class)
        .add(Restrictions.and(
            Restrictions.eq("idvalorpro",
idvalorpro),
            Restrictions.eq("ano", ano)))
        .setMaxResults(1)
        .uniqueResult();
        return rg;
    }
    public void save(Valorpropiedad reg) {
        Session s =
HibernateUtil.getSessionFactory().getCurrentSession();
        s.beginTransaction();
        s.saveOrUpdate(reg);
        s.getTransaction().commit();
    }
}

```

```

    public void eliminar(Valorpropiedad reg) {
        Session s =
HibernateUtil.getSessionFactory().getCurrentSession();
        s.beginTransaction();
        s.delete(reg);
        s.getTransaction().commit();
    }
}

```

DBano

```

/*
 * To change this license header, choose License Headers in
Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package dao;
import java.util.ArrayList;
import java.util.List;
import modelo.Bano;
import modelo.HibernateUtil;
import org.hibernate.criterion.Restrictions;
/**
 *
 * @author Manrrique
 */
public class DBano {
    public List<Bano> all() {
        List<Bano> l = new ArrayList();
        l = (List)
HibernateUtil.getSession().createCriteria(Bano.class).list();
        return l;
    }
    public Bano data(String letra) {
        Bano rg = (Bano)
HibernateUtil.getSession().createCriteria(Bano.class)
                .add(Restrictions.eq("letra", letra))
                .setMaxResults(1)
                .uniqueResult();
        return rg;
    }
}

```

class DCaja

```

/*
 * To change this license header, choose License Headers in
Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package dao;
import java.util.List;
import modelo.Caja;
import modelo.Depreciacion;

```

```

import modelo.HibernateUtil;
import org.hibernate.Session;
import org.hibernate.criterion.Restrictions;

/**
 *
 * @author Manrrique
 */
public class DCaja {
    public List<Caja> all(Integer idprop) {
        List rg =
HibernateUtil.getSession().createCriteria(Caja.class)
                .add(Restrictions.eq("idPropiedad", idprop))
                .list();
        return rg;
    }
    public void save(Caja cc) {
        Session s =
HibernateUtil.getSessionFactory().getCurrentSession();
        s.beginTransaction();
        s.saveOrUpdate(cc);
        s.getTransaction().commit();
    }
    public void eliminar(Caja reg) {
        Session s =
HibernateUtil.getSessionFactory().getCurrentSession();
        s.beginTransaction();
        s.delete(reg);
        s.getTransaction().commit();
    }
}

```

class DCcont

```

/*
 * To change this license header, choose License Headers in
Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package dao;
import java.util.ArrayList;
import java.util.List;
import modelo.HibernateUtil;
import modelo.Propietario;
import org.hibernate.Session;
import org.hibernate.criterion.Restrictions;
/**
 *
 * @author YONNY
 */
public class DCcont {
    public List all(String nom, String tdoc) {
        List l = new ArrayList();
        l = (List)
HibernateUtil.getSession().createCriteria(Propietario.class).lis
t();
        return l;
    }
}

```

```

    }
    public void save(Propietario reg) {
        Session s =
HibernateUtil.getSessionFactory().getCurrentSession();
        s.beginTransaction();
        s.saveOrUpdate(reg);
        s.getTransaction().commit();
    }
    public void eliminar(Propietario reg) {
        Session s =
HibernateUtil.getSessionFactory().getCurrentSession();
        s.beginTransaction();
        s.delete(reg);
        s.getTransaction().commit();
    }
}
}

```

class DPropiedad

```

/*
 * To change this license header, choose License Headers in
Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package dao;
import java.util.List;
import modelo.Caja;
import modelo.HibernateUtil;
import modelo.Predio;
import modelo.Propiedad;
import modelo.Propietario;
import modelo.Valorpropiedad;
import org.hibernate.Session;
import org.hibernate.criterion.Restrictions;
/**
 *
 * @author Manrrique
 */
public class DPropiedad {

    public List all(Propietario propietario) {
        List rg =
HibernateUtil.getSession().createCriteria(Propiedad.class)
                .add(Restrictions.eq("propietario",
propietario))
                .list();
        return rg;
    }
    public Propiedad data(Propiedad pro) {
        Propiedad rg = (Propiedad)
HibernateUtil.getSession().createCriteria(Propiedad.class)
                .add(Restrictions.eq("idprop", pro.getIdprop()))
                .setMaxResults(1)
                .uniqueResult();
        return rg;
    }
    public List<Predio> predios(int a, Propiedad p) {

```

```

        List rg = HibernateUtil.getSession()
            .createCriteria(Predio.class, "predio")
            .createAlias("predio.propiedad", "propiedad")
            .add(Restrictions.and(
                Restrictions.eq("propiedad", p),
                Restrictions.eq("ano", a)
            )).list();
        return rg;
    }
    public List<Valorpropiedad> valor(int a, Propiedad pr) {
        List rg = HibernateUtil.getSession()
            .createCriteria(Valorpropiedad.class, "valor")
            .createAlias("valor.propiedad", "propiedad")
            .add(Restrictions.and(
                Restrictions.eq("propiedad", pr),
                Restrictions.eq("ano", a)
            )).list();
        return rg;
    }
    public List<Valorpropiedad> anos(Propiedad pr) {
        List<Valorpropiedad> rg = (List)
HibernateUtil.getSession()
            .createCriteria(Valorpropiedad.class, "valor")
            .createAlias("valor.propiedad", "propiedad")
            .add(Restrictions.eq("propiedad", pr)).list();
        return rg;
    }
    public boolean existe(Propiedad pr, int a) {
        List rg = HibernateUtil.getSession()
            .createCriteria(Valorpropiedad.class, "valor")
            .createAlias("valor.propiedad", "propiedad")
            .add(Restrictions.and(
                Restrictions.eq("propiedad", pr),
                Restrictions.eq("ano", a)
            )).list();
        return rg.isEmpty();
    }
    public boolean existeC(Propiedad pr, int a) {
        List rg = HibernateUtil.getSession()
            .createCriteria(Caja.class)
            .add(Restrictions.and(
                Restrictions.eq("idPropiedad",
pr.getIdprop()),
                Restrictions.eq("ano", a)
            )).list();
        return rg.isEmpty();
    }
    public void EAno(Propiedad pr, int a) {
        Valorpropiedad rg = (Valorpropiedad)
HibernateUtil.getSession()
            .createCriteria(Valorpropiedad.class, "valor")
            .createAlias("valor.propiedad", "propiedad")
            .add(Restrictions.and(
                Restrictions.eq("propiedad", pr),
                Restrictions.eq("ano", a)
            )).setMaxResults(1).uniqueResult();

        Session s =
HibernateUtil.getSessionFactory().getCurrentSession();

```



```

        s.beginTransaction();
        s.delete(rg);
        s.getTransaction().commit();
    }
}

```

class DUs

```

/**
 * To change this license header, choose License Headers in
 * Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package dao;
import java.util.ArrayList;
import java.util.List;
import modelo.HibernateUtil;
import modelo.Usuario;
import org.hibernate.Session;
import org.hibernate.criterion.Restrictions;
/**
 *
 * @author YONNY
 */
public class DUs {

    public List all() {
        Session s =
HibernateUtil.getSessionFactory().openSession();
        List l = new ArrayList();
        l = (List) s.createCriteria(Usuario.class).list();
        s.close();
        return l;
    }
    public Usuario data(String us, String ps) {
        Session s =
HibernateUtil.getSessionFactory().openSession();
        Usuario rg = (Usuario) s.createCriteria(Usuario.class)
            .add(Restrictions.and(Restrictions.eq("usu",
us), Restrictions.eq("cont", ps)))
            .setMaxResults(1)
            .uniqueResult();
        s.close();
        return rg;
    }
    public void save(Usuario reg) {
        Session s =
HibernateUtil.getSessionFactory().openSession();
        s.beginTransaction();
        s.saveOrUpdate(reg);
        s.getTransaction().commit();
        s.close();
    }
    public void eliminar(Usuario reg) {
        Session s =
HibernateUtil.getSessionFactory().openSession();

```

```

        s.beginTransaction();
        s.delete(reg);
        s.getTransaction().commit();
        s.close();
    }
}

```

c. MODELO

class Bano

```

package modelo;
// Generated 25/06/2019 08:56:13 PM by Hibernate Tools 4.3.1
import java.math.BigDecimal;
import java.util.HashSet;
import java.util.Set;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.FetchType;
import javax.persistence.GeneratedValue;
import static javax.persistence.GenerationType.IDENTITY;
import javax.persistence.Id;
import javax.persistence.OneToMany;
import javax.persistence.Table;
/**
 * Bano generated by hbm2java
 */
@Entity
@Table(name="bano"
, catalog="autovaluo"
)
public class Bano implements java.io.Serializable {
    private Integer idbano;
    private String letra;
    private String descripcion;
    private BigDecimal monto;
    private Set<Predio> predios = new HashSet<Predio>(0);
    public Bano() {
    }
    public Bano(String letra) {
        this.letra = letra;
    }
    public Bano(String letra, String descripcion, BigDecimal
monto, Set<Predio> predios) {
        this.letra = letra;
        this.descripcion = descripcion;
        this.monto = monto;
        this.predios = predios;
    }
    @Id @GeneratedValue(strategy=IDENTITY)
    @Column(name="idbano", unique=true, nullable=false)
    public Integer getIdbano() {
        return this.idbano;
    }
    public void setIdbano(Integer idbano) {
        this.idbano = idbano;
    }
    @Column(name="letra", nullable=false, length=15)

```

```

public String getLetra() {
    return this.letra;
}
public void setLetra(String letra) {
    this.letra = letra;
}
@Column(name="descripcion", length=250)
public String getDescripcion() {
    return this.descripcion;
}
public void setDescripcion(String descripcion) {
    this.descripcion = descripcion;
}
@Column(name="monto", precision=10)
public BigDecimal getMonto() {
    return this.monto;
}
public void setMonto(BigDecimal monto) {
    this.monto = monto;
}
@OneToMany(fetch=FetchType.LAZY, mappedBy="bano")
public Set<Predio> getPredios() {
    return this.predios;
}
public void setPredios(Set<Predio> predios) {
    this.predios = predios;
}
}

```

class Caja

```

package modelo;
// Generated 25/06/2019 08:56:13 PM by Hibernate Tools 4.3.1
import java.math.BigDecimal;
import java.util.Date;
import javafx.beans.property.SimpleStringProperty;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import static javax.persistence.GenerationType.IDENTITY;
import javax.persistence.Id;
import javax.persistence.Table;
import javax.persistence.Temporal;
import javax.persistence.TemporalType;
/**
 * Caja generated by hbm2java
 */
@Entity
@Table(name = "caja",
        catalog = "autovaluo"
)
public class Caja implements java.io.Serializable {
    private Integer idCaja;
    private int idPropiedad;
    private int ano;
    private BigDecimal base;
    private BigDecimal monto;
    private boolean estado;
}

```

```

private BigDecimal uit;
private Date fecha;
public Caja() {
}
public Caja(int idPropiedad, int ano, BigDecimal base,
BigDecimal monto, boolean estado, BigDecimal uit, Date fecha) {
    this.idPropiedad = idPropiedad;
    this.ano = ano;
    this.base = base;
    this.monto = monto;
    this.estado = estado;
    this.uit = uit;
    this.fecha = fecha;
}
@Id
@GeneratedValue(strategy = IDENTITY)

@Column(name = "idCaja", unique = true, nullable = false)
public Integer getIdCaja() {
    return this.idCaja;
}
public void setIdCaja(Integer idCaja) {
    this.idCaja = idCaja;
}
@Column(name = "idPropiedad", nullable = false)
public int getIdPropiedad() {
    return this.idPropiedad;
}
public void setIdPropiedad(int idPropiedad) {
    this.idPropiedad = idPropiedad;
}
@Column(name = "ano", nullable = false)
public int getAno() {
    return this.ano;
}
public void setAno(int ano) {
    this.ano = ano;
}
@Column(name = "base", nullable = false, precision = 10)
public BigDecimal getBase() {
    return this.base;
}
public void setBase(BigDecimal base) {
    this.base = base;
}
@Column(name = "monto", nullable = false, precision = 10)
public BigDecimal getMonto() {
    return this.monto;
}
public void setMonto(BigDecimal monto) {
    this.monto = monto;
}
@Column(name = "estado", nullable = false)
public boolean isEstado() {
    return this.estado;
}
public void setEstado(boolean estado) {
    this.estado = estado;
}
}

```

```

@Column(name = "uit", nullable = false, precision = 10)
public BigDecimal getUit() {
    return this.uit;
}
public void setUit(BigDecimal uit) {
    this.uit = uit;
}
@Temporal(TemporalType.DATE)
@Column(name = "fecha", nullable = false, length = 10)
public Date getFecha() {
    return this.fecha;
}
public void setFecha(Date fecha) {
    this.fecha = fecha;
}
public SimpleStringProperty ano() {
    return new
SimpleStringProperty(String.valueOf(this.getAno()));
}
public SimpleStringProperty base() {
    return new
SimpleStringProperty(String.valueOf(this.getBase()));
}
public SimpleStringProperty monto() {
    return new
SimpleStringProperty(String.valueOf(this.getMonto()));
}
public SimpleStringProperty estado() {
    if (this.isEstado()) {
        return new SimpleStringProperty("Pagado");
    } else {
        return new SimpleStringProperty("Activo");
    }
}
}
}

```

class Conservacion

```

package modelo;
// Generated 25/06/2019 08:56:13 PM by Hibernate Tools 4.3.1
import java.util.HashSet;
import java.util.Set;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.FetchType;
import javax.persistence.GeneratedValue;
import static javax.persistence.GenerationType.IDENTITY;
import javax.persistence.Id;
import javax.persistence.OneToMany;
import javax.persistence.Table;
/**
 * Conservacion generated by hbm2java
 */
@Entity
@Table(name="conservacion"
, catalog="autovaluo"
)
public class Conservacion implements java.io.Serializable {

```

```

private Integer idconservacion;
private String descripcion;
private Set<Predio> predios = new HashSet<Predio>(0);

public Conservacion() {
}
public Conservacion(String descripcion, Set<Predio> predios)
{
    this.descripcion = descripcion;
    this.predios = predios;
}
@Id @GeneratedValue(strategy=IDENTITY)
@Column(name="idconservacion", unique=true, nullable=false)
public Integer getIdconservacion() {
    return this.idconservacion;
}
public void setIdconservacion(Integer idconservacion) {
    this.idconservacion = idconservacion;
}
@Column(name="descripcion", length=60)
public String getDescripcion() {
    return this.descripcion;
}
public void setDescripcion(String descripcion) {
    this.descripcion = descripcion;
}
@OneToMany(fetch=FetchType.LAZY, mappedBy="conservacion")
public Set<Predio> getPredios() {
    return this.predios;
}
public void setPredios(Set<Predio> predios) {
    this.predios = predios;
}
}

```

class Depreciacion

```

package modelo;
// Generated 25/06/2019 08:56:13 PM by Hibernate Tools 4.3.1
import java.math.BigDecimal;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import static javax.persistence.GenerationType.IDENTITY;
import javax.persistence.Id;
import javax.persistence.Table;
/**
 * Depreciacion generated by hbm2java
 */
@Entity
@Table(name="depreciacion"
, catalog="autovaluo"
)
public class Depreciacion implements java.io.Serializable {
    private Integer iddepreciacion;
    private Integer idhabitacion;
    private Integer antiguedad;
    private Integer material;
}

```

```

private BigDecimal muybueno;
private BigDecimal bueno;
private BigDecimal regular;
private BigDecimal malo;

public Depreciacion() {
}
public Depreciacion(Integer idhabitacion, Integer
antiguedad, Integer material, BigDecimal muybueno, BigDecimal
bueno, BigDecimal regular, BigDecimal malo) {
    this.idhabitacion = idhabitacion;
    this.antiguedad = antiguedad;
    this.material = material;
    this.muybueno = muybueno;
    this.bueno = bueno;
    this.regular = regular;
    this.malo = malo;
}
@Id @GeneratedValue(strategy=IDENTITY)

@Column(name="iddepreciacion", unique=true, nullable=false)
public Integer getIddepreciacion() {
    return this.iddepreciacion;
}
public void setIddepreciacion(Integer iddepreciacion) {
    this.iddepreciacion = iddepreciacion;
}
@Column(name="idhabitacion")
public Integer getIdhabitacion() {
    return this.idhabitacion;
}
public void setIdhabitacion(Integer idhabitacion) {
    this.idhabitacion = idhabitacion;
}
@Column(name="antiguedad")
public Integer getAntiguedad() {
    return this.antiguedad;
}
public void setAntiguedad(Integer antiguedad) {
    this.antiguedad = antiguedad;
}
@Column(name="material")
public Integer getMaterial() {
    return this.material;
}
public void setMaterial(Integer material) {
    this.material = material;
}
@Column(name="muybueno", precision=10)
public BigDecimal getMuybueno() {
    return this.muybueno;
}
public void setMuybueno(BigDecimal muybueno) {
    this.muybueno = muybueno;
}

@Column(name="bueno", precision=10)
public BigDecimal getBueno() {
    return this.bueno;
}

```

```

    }
    public void setBueno(BigDecimal bueno) {
        this.bueno = bueno;
    }
    @Column(name="regular", precision=10)
    public BigDecimal getRegular() {
        return this.regular;
    }
    public void setRegular(BigDecimal regular) {
        this.regular = regular;
    }
    @Column(name="malo", precision=10)
    public BigDecimal getMalo() {
        return this.malo;
    }
    public void setMalo(BigDecimal malo) {
        this.malo = malo;
    }
}

```

class Habitation

```

package modelo;
// Generated 25/06/2019 08:56:13 PM by Hibernate Tools 4.3.1
import java.util.HashSet;
import java.util.Set;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.FetchType;
import javax.persistence.GeneratedValue;
import static javax.persistence.GenerationType.IDENTITY;
import javax.persistence.Id;
import javax.persistence.OneToMany;
import javax.persistence.Table;
/**
 * Habitation generated by hbm2java
 */
@Entity
@Table(name="habitation"
, catalog="autovaluo"
)
public class Habitation implements java.io.Serializable {

    private Integer idhabitation;
    private String descripcion;
    private Set<Predio> predios = new HashSet<Predio>(0);
    public Habitation() {
    }
    public Habitation(String descripcion, Set<Predio> predios) {
        this.descripcion = descripcion;
        this.predios = predios;
    }
    @Id @GeneratedValue(strategy=IDENTITY)

    @Column(name="idhabitation", unique=true, nullable=false)
    public Integer getIdhabitation() {
        return this.idhabitation;
    }
}

```



```

    public void setIdhabitacion(Integer idhabitacion) {
        this.idhabitacion = idhabitacion;
    }
    @Column(name="descripcion", length=60)
    public String getDescripcion() {
        return this.descripcion;
    }
    public void setDescripcion(String descripcion) {
        this.descripcion = descripcion;
    }
    @OneToMany(fetch=FetchType.LAZY, mappedBy="habitacion")
    public Set<Predio> getPredios() {
        return this.predios;
    }
    public void setPredios(Set<Predio> predios) {
        this.predios = predios;
    }
}

```

class HibernateUtil

```

/*
 * To change this license header, choose License Headers in
 * Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package modelo;
import org.hibernate.Session;
import org.hibernate.cfg.AnnotationConfiguration;
import org.hibernate.SessionFactory;
/**
 * Hibernate Utility class with a convenient method to get
 * Session Factory
 * object.
 *
 * @author Manrrique
 */
public class HibernateUtil {
    private static final SessionFactory sessionFactory;
    static {
        try {
            // Create the SessionFactory from standard
            (hibernate.cfg.xml)
            // config file.
            sessionFactory = new
            AnnotationConfiguration().configure().buildSessionFactory();
        } catch (Throwable ex) {
            // Log the exception.
            System.err.println("Initial SessionFactory creation
            failed." + ex);
            throw new ExceptionInInitializerError(ex);
        }
    }
    public static SessionFactory getSessionFactory() {
        return sessionFactory;
    }
    public static Session getSession() {

```

```

        return sessionFactory.openSession();
    }
    public SessionFactory Sesion() {
        return new
AnnotationConfiguration().configure().buildSessionFactory();
    }
}

```

class Instalaciones

```

package modelo;
// Generated 25/06/2019 08:56:13 PM by Hibernate Tools 4.3.1
import java.math.BigDecimal;
import java.util.HashSet;
import java.util.Set;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.FetchType;
import javax.persistence.GeneratedValue;
import static javax.persistence.GenerationType.IDENTITY;
import javax.persistence.Id;
import javax.persistence.OneToMany;
import javax.persistence.Table;
/**
 * Instalaciones generated by hbm2java
 */
@Entity
@Table(name="instalaciones"
, catalog="autovaluo"
)
public class Instalaciones implements java.io.Serializable {
    private Integer idinstalaciones;
    private String letra;
    private String descripcion;
    private BigDecimal monto;
    private Set<Predio> predios = new HashSet<Predio>(0);
    public Instalaciones() {
    }
    public Instalaciones(String letra) {
        this.letra = letra;
    }
    public Instalaciones(String letra, String descripcion,
BigDecimal monto, Set<Predio> predios) {
        this.letra = letra;
        this.descripcion = descripcion;
        this.monto = monto;
        this.predios = predios;
    }
    @Id @GeneratedValue(strategy=IDENTITY)

    @Column(name="idinstalaciones", unique=true, nullable=false)
    public Integer getIdinstalaciones() {
        return this.idinstalaciones;
    }
    public void setIdinstalaciones(Integer idinstalaciones) {
        this.idinstalaciones = idinstalaciones;
    }
    @Column(name="letra", nullable=false, length=15)

```

```

public String getLetra() {
    return this.letra;
}
public void setLetra(String letra) {
    this.letra = letra;
}
@Column(name="descripcion", length=250)
public String getDescripcion() {
    return this.descripcion;
}
public void setDescripcion(String descripcion) {
    this.descripcion = descripcion;
}
@Column(name="monto", precision=10)
public BigDecimal getMonto() {
    return this.monto;
}
public void setMonto(BigDecimal monto) {
    this.monto = monto;
}
}
@OneToMany(fetch=FetchType.LAZY, mappedBy="instalaciones")
public Set<Predio> getPredios() {
    return this.predios;
}
public void setPredios(Set<Predio> predios) {
    this.predios = predios;
}
}
}

```

class Materialmuro

```

package modelo;
// Generated 25/06/2019 08:56:13 PM by Hibernate Tools 4.3.1
import java.math.BigDecimal;
import java.util.HashSet;
import java.util.Set;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.FetchType;
import javax.persistence.GeneratedValue;
import static javax.persistence.GenerationType.IDENTITY;
import javax.persistence.Id;
import javax.persistence.OneToMany;
import javax.persistence.Table;
/**
 * Materialmuro generated by hbm2java
 */
@Entity
@Table(name="materialmuro"
, catalog="autovaluo"
)
public class Materialmuro implements java.io.Serializable {

    private Integer idmaterialmuro;
    private String letra;
    private String descripcion;
    private BigDecimal monto;
}

```

```

    private Set<Predio> predios = new HashSet<Predio>(0);
    public Materialmuro() {
    }
    public Materialmuro(String letra) {
        this.letra = letra;
    }
    public Materialmuro(String letra, String descripcion,
BigDecimal monto, Set<Predio> predios) {
        this.letra = letra;
        this.descripcion = descripcion;
        this.monto = monto;
        this.predios = predios;
    }
    @Id @GeneratedValue(strategy=IDENTITY)

    @Column(name="idmaterialmuro", unique=true, nullable=false)
    public Integer getIdmaterialmuro() {
        return this.idmaterialmuro;
    }

    public void setIdmaterialmuro(Integer idmaterialmuro) {
        this.idmaterialmuro = idmaterialmuro;
    }
    @Column(name="letra", nullable=false, length=15)
    public String getLetra() {
        return this.letra;
    }
    public void setLetra(String letra) {
        this.letra = letra;
    }
    @Column(name="descripcion", length=250)
    public String getDescripcion() {
        return this.descripcion;
    }
    public void setDescripcion(String descripcion) {
        this.descripcion = descripcion;
    }
    @Column(name="monto", precision=10)
    public BigDecimal getMonto() {
        return this.monto;
    }
    public void setMonto(BigDecimal monto) {
        this.monto = monto;
    }
    @OneToMany(fetch=FetchType.LAZY, mappedBy="materialmuro")
    public Set<Predio> getPredios() {
        return this.predios;
    }
    public void setPredios(Set<Predio> predios) {
        this.predios = predios;
    }
}

```

class Matpredominante

```

package modelo;
// Generated 25/06/2019 08:56:13 PM by Hibernate Tools 4.3.1
import java.util.HashSet;

```

```

import java.util.Set;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.FetchType;
import javax.persistence.GeneratedValue;
import static javax.persistence.GenerationType.IDENTITY;
import javax.persistence.Id;
import javax.persistence.OneToMany;
import javax.persistence.Table;
/**
 * Matpredominante generated by hbm2java
 */
@Entity
@Table(name="matpredominante"
, catalog="autovaluo"
)
public class Matpredominante implements java.io.Serializable {

    private Integer idmatpredominante;
    private String descripcion;
    private Set<Predio> predios = new HashSet<Predio>(0);

    public Matpredominante() {
    }
    public Matpredominante(String descripcion, Set<Predio>
predios) {
        this.descripcion = descripcion;
        this.predios = predios;
    }
    @Id @GeneratedValue(strategy=IDENTITY)

    @Column(name="idmatpredominante", unique=true,
nullable=false)
    public Integer getIdmatpredominante() {
        return this.idmatpredominante;
    }
    public void setIdmatpredominante(Integer idmatpredominante)
{
        this.idmatpredominante = idmatpredominante;
    }
    @Column(name="descripcion", length=60)
    public String getDescripcion() {
        return this.descripcion;
    }
    public void setDescripcion(String descripcion) {
        this.descripcion = descripcion;
    }
    @OneToMany(fetch=FetchType.LAZY, mappedBy="matpredominante")
    public Set<Predio> getPredios() {
        return this.predios;
    }
    public void setPredios(Set<Predio> predios) {
        this.predios = predios;
    }
}

```

class Piso

```

package modelo;
// Generated 25/06/2019 08:56:13 PM by Hibernate Tools 4.3.1
import java.math.BigDecimal;
import java.util.HashSet;
import java.util.Set;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.FetchType;
import javax.persistence.GeneratedValue;
import static javax.persistence.GenerationType.IDENTITY;
import javax.persistence.Id;
import javax.persistence.OneToOne;
import javax.persistence.Table;
/**
 * Piso generated by hbm2java
 */
@Entity
@Table(name="piso"
, catalog="autovaluo"
)
public class Piso implements java.io.Serializable {
    private Integer idpiso;
    private String letra;
    private String descripcion;
    private BigDecimal monto;
    private Set<Predio> predios = new HashSet<Predio>(0);
    public Piso() {
    }

    public Piso(String letra) {
        this.letra = letra;
    }

    public Piso(String letra, String descripcion, BigDecimal
monto, Set<Predio> predios) {
        this.letra = letra;
        this.descripcion = descripcion;
        this.monto = monto;
        this.predios = predios;
    }

    @Id @GeneratedValue(strategy=IDENTITY)

    @Column(name="idpiso", unique=true, nullable=false)
    public Integer getIdpiso() {
        return this.idpiso;
    }

    public void setIdpiso(Integer idpiso) {
        this.idpiso = idpiso;
    }

    @Column(name="letra", nullable=false, length=15)
    public String getLetra() {
        return this.letra;
    }

    public void setLetra(String letra) {
        this.letra = letra;
    }

    @Column(name="descripcion", length=250)
    public String getDescripcion() {
        return this.descripcion;
    }
}

```

```

    public void setDescription(String descripcion) {
        this.descripcion = descripcion;
    }
    @Column(name="monto", precision=10)
    public BigDecimal getMonto() {
        return this.monto;
    }
    public void setMonto(BigDecimal monto) {
        this.monto = monto;
    }
    @OneToMany(fetch=FetchType.LAZY, mappedBy="piso")
    public Set<Predio> getPredios() {
        return this.predios;
    }
    public void setPredios(Set<Predio> predios) {
        this.predios = predios;
    }
}

```

class Predio

```

package modelo;
// Generated 25/06/2019 08:56:13 PM by Hibernate Tools 4.3.1
import java.math.BigDecimal;
import javafx.beans.property.SimpleStringProperty;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.FetchType;
import javax.persistence.GeneratedValue;
import static javax.persistence.GenerationType.IDENTITY;
import javax.persistence.Id;
import javax.persistence.JoinColumn;
import javax.persistence.ManyToOne;
import javax.persistence.Table;
/**
 * Predio generated by hbm2java
 */
@Entity
@Table(name = "predio",
        catalog = "autovaluo"
)
public class Predio implements java.io.Serializable {
    private Integer idpre;
    private Bano bano;
    private Conservacion conservacion;
    private Habitacion habitacion;
    private Instalaciones instalaciones;
    private Materialmuro materialmuro;
    private Matpredominante matpredominante;
    private Piso piso;
    private Propiedad propiedad;
    private Puerta puerta;
    private Revestimiento revestimiento;
    private Techo techo;
    private Integer ano;
    private String bloque;
    private String piso_1;
    private BigDecimal valorconst;
    private BigDecimal area;
    private BigDecimal deprec;
}

```

```

public Predio() {
}
public Predio(Bano bano, Conservacion conservacion,
Habitacion habitacion, Instalaciones instalaciones, Materialmuro
materialmuro, Matpredominante matpredominante, Piso piso,
Propiedad propiedad, Puerta puerta, Revestimiento revestimiento,
Techo techo, Integer ano, String bloque, String piso_1,
BigDecimal valorconst, BigDecimal area, BigDecimal deprec) {
    this.bano = bano;
    this.conservacion = conservacion;
    this.habitacion = habitacion;
    this.instalaciones = instalaciones;
    this.materialmuro = materialmuro;
    this.matpredominante = matpredominante;
    this.piso = piso;
    this.propiedad = propiedad;
    this.puerta = puerta;
    this.revestimiento = revestimiento;
    this.techo = techo;
    this.ano = ano;
    this.bloque = bloque;
    this.piso_1 = piso_1;
    this.valorconst = valorconst;
    this.area = area;
    this.deprec = deprec;
}
@Id
@GeneratedValue(strategy = IDENTITY)
@Column(name = "idpre", unique = true, nullable = false)
public Integer getIdpre() {
    return this.idpre;
}
public void setIdpre(Integer idpre) {
    this.idpre = idpre;
}
@ManyToOne(fetch = FetchType.LAZY)
@JoinColumn(name = "idbano")
public Bano getBano() {
    return this.bano;
}
public void setBano(Bano bano) {
    this.bano = bano;
}
@ManyToOne(fetch = FetchType.LAZY)
@JoinColumn(name = "idconservacion")
public Conservacion getConservacion() {
    return this.conservacion;
}
public void setConservacion(Conservacion conservacion) {
    this.conservacion = conservacion;
}
@ManyToOne(fetch = FetchType.LAZY)
@JoinColumn(name = "idhabitacion")
public Habitacion getHabitacion() {
    return this.habitacion;
}
public void setHabitacion(Habitacion habitacion) {
    this.habitacion = habitacion;
}
}

```



```

@ManyToOne(fetch = FetchType.LAZY)
@JoinColumn(name = "idinstalaciones")
public Instalaciones getInstalaciones() {
    return this.instalaciones;
}
public void setInstalaciones(Instalaciones instalaciones) {
    this.instalaciones = instalaciones;
}
@ManyToOne(fetch = FetchType.LAZY)
@JoinColumn(name = "idmaterialmuro")
public Materialmuro getMaterialmuro() {
    return this.materialmuro;
}
public void setMaterialmuro(Materialmuro materialmuro) {
    this.materialmuro = materialmuro;
}
@ManyToOne(fetch = FetchType.LAZY)
@JoinColumn(name = "idmatpredominante")
public Matpredominante getMatpredominante() {
    return this.matpredominante;
}
public void setMatpredominante(Matpredominante
matpredominante) {
    this.matpredominante = matpredominante;
}
@ManyToOne(fetch = FetchType.LAZY)
@JoinColumn(name = "idpiso")
public Piso getPiso() {
    return this.piso;
}
public void setPiso(Piso piso) {
    this.piso = piso;
}
@ManyToOne(fetch = FetchType.LAZY)
@JoinColumn(name = "idprop")
public Propiedad getPropiedad() {
    return this.propiedad;
}
public void setPropiedad(Propiedad propiedad) {
    this.propiedad = propiedad;
}
@ManyToOne(fetch = FetchType.LAZY)
@JoinColumn(name = "idpuerta")
public Puerta getPuerta() {
    return this.puerta;
}
public void setPuerta(Puerta puerta) {
    this.puerta = puerta;
}
@ManyToOne(fetch = FetchType.LAZY)
@JoinColumn(name = "idrevestimiento")
public Revestimiento getRevestimiento() {
    return this.revestimiento;
}
public void setRevestimiento(Revestimiento revestimiento) {
    this.revestimiento = revestimiento;
}

@ManyToOne(fetch = FetchType.LAZY)

```

```

@JoinColumn(name = "idtecho")
public Techo getTecho() {
    return this.techo;
}
public void setTecho(Techo techo) {
    this.techo = techo;
}
@Column(name = "ano")
public Integer getAno() {
    return this.ano;
}
public void setAno(Integer ano) {
    this.ano = ano;
}
@Column(name = "bloque", length = 60)
public String getBloque() {
    return this.bloque;
}
public void setBloque(String bloque) {
    this.bloque = bloque;
}
@Column(name = "piso", length = 45)
public String getPiso_1() {
    return this.piso_1;
}
public void setPiso_1(String piso_1) {
    this.piso_1 = piso_1;
}
@Column(name = "valorconst", precision = 10)
public BigDecimal getValorconst() {
    return this.valorconst;
}
public void setValorconst(BigDecimal valorconst) {
    this.valorconst = valorconst;
}
@Column(name = "area", precision = 10)
public BigDecimal getArea() {
    return this.area;
}
public void setArea(BigDecimal area) {
    this.area = area;
}
@Column(name = "deprec", precision = 10)
public BigDecimal getDeprec() {
    return this.deprec;
}
public void setDeprec(BigDecimal deprec) {
    this.deprec = deprec;
}
public SimpleStringProperty ano() {
    return new
SimpleStringProperty(String.valueOf(this.getAno()));
}
public SimpleStringProperty Clasificacion() {
    return new
SimpleStringProperty(String.valueOf(this.getHabitacion().getIdha
bitacion()));
}
public SimpleStringProperty Material() {

```

```

        return new
SimpleStringProperty(String.valueOf(this.getMatpredominante().get
tIdmatpredominante()));
    }
    public SimpleStringProperty Estado() {
        return new
SimpleStringProperty(String.valueOf(this.getConservacion().getId
conservacion()));
    }
    public SimpleStringProperty muro() {
        return new
SimpleStringProperty(String.valueOf(this.getMaterialmuro().getLe
tra()));
    }
    public SimpleStringProperty techo() {
        return new
SimpleStringProperty(String.valueOf(this.getTecho().getLetra()));
    }
    public SimpleStringProperty pisos() {
        return new
SimpleStringProperty(String.valueOf(this.getPiso().getLetra()));
    }
    public SimpleStringProperty puerta() {
        return new
SimpleStringProperty(String.valueOf(this.getPuerta().getLetra()
));
    }
    public SimpleStringProperty revestimiento() {
        return new
SimpleStringProperty(String.valueOf(this.getRevestimiento().getL
etra()));
    }
    public SimpleStringProperty bano() {
        return new
SimpleStringProperty(String.valueOf(this.getBano().getLetra()));
    }
    public SimpleStringProperty instalacion() {
        return new
SimpleStringProperty(String.valueOf(this.getInstalaciones().getL
etra()));
    }
    public SimpleStringProperty aconstruccion() {
        return new
SimpleStringProperty(String.valueOf(this.getArea()));
    }
    public SimpleStringProperty vconstruccion() {
        return new
SimpleStringProperty(String.valueOf(this.getValorconst()));
    }
    public SimpleStringProperty dconstruccion() {
        return new
SimpleStringProperty(String.valueOf(this.getDeprec()));
    }
}

```

class Propiedad

```

package modelo;
// Generated 25/06/2019 08:56:13 PM by Hibernate Tools 4.3.1
import java.util.HashSet;
import java.util.Set;
import javax.faces.property.SimpleStringProperty;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.FetchType;
import javax.persistence.GeneratedValue;
import static javax.persistence.GenerationType.IDENTITY;
import javax.persistence.Id;
import javax.persistence.JoinColumn;
import javax.persistence.ManyToOne;
import javax.persistence.OneToMany;
import javax.persistence.Table;
/**
 * Propiedad generated by hbm2java
 */
@Entity
@Table(name = "propiedad",
        catalog = "autovaluo"
)
public class Propiedad implements Serializable {
    private Integer idprop;
    private Propietario propietario;
    private String dep;
    private String pro;
    private String dis;
    private String direccion;
    private Set<Valorpropiedad> valorpropiedades = new
HashSet<Valorpropiedad>(0);
    private Set<Predio> predios = new HashSet<Predio>(0);
    public Propiedad() {
    }
    public Propiedad(Propietario propietario, String dep, String
pro, String dis, String direccion) {
        this.propietario = propietario;
        this.dep = dep;
        this.pro = pro;
        this.dis = dis;
        this.direccion = direccion;
    }
    public Propiedad(Propietario propietario, String dep, String
pro, String dis, String direccion, Set<Valorpropiedad>
valorpropiedades, Set<Predio> predios) {
        this.propietario = propietario;
        this.dep = dep;
        this.pro = pro;
        this.dis = dis;
        this.direccion = direccion;
        this.valorpropiedades = valorpropiedades;
        this.predios = predios;
    }
    @Id
    @GeneratedValue(strategy = IDENTITY)
    @Column(name = "idprop", unique = true, nullable = false)
    public Integer getIdprop() {
        return this.idprop;
    }
}

```

```

    }
    public void setIdprop(Integer idprop) {
        this.idprop = idprop;
    }
    @ManyToOne(fetch = FetchType.LAZY)
    @JoinColumn(name = "idpro", nullable = false)
    public Propietario getPropietario() {
        return this.propietario;
    }
    public void setPropietario(Propietario propietario) {
        this.propietario = propietario;
    }
    @Column(name = "dep", nullable = false, length = 45)
    public String getDep() {
        return this.dep;
    }
    public void setDep(String dep) {
        this.dep = dep;
    }
    @Column(name = "pro", nullable = false, length = 45)
    public String getPro() {
        return this.pro;
    }
    public void setPro(String pro) {
        this.pro = pro;
    }
    @Column(name = "dis", nullable = false, length = 45)
    public String getDis() {
        return this.dis;
    }
    public void setDis(String dis) {
        this.dis = dis;
    }
    @Column(name = "direccion", nullable = false, length = 60)
    public String getDireccion() {
        return this.direccion;
    }
    public void setDireccion(String direccion) {
        this.direccion = direccion;
    }
    @OneToMany(fetch = FetchType.LAZY, mappedBy = "propiedad")
    public Set<Valorpropiedad> getValorpropiedades() {
        return this.valorpropiedades;
    }
    public void setValorpropiedades(Set<Valorpropiedad>
valorpropiedades) {
        this.valorpropiedades = valorpropiedades;
    }
    @OneToMany(fetch = FetchType.LAZY, mappedBy = "propiedad")
    public Set<Predio> getPredios() {
        return this.predios;
    }
    public void setPredios(Set<Predio> predios) {
        this.predios = predios;
    }
    public SimpleStringProperty direccion() {
        return new
SimpleStringProperty(String.valueOf(this.getDireccion()));
    }
}

```

```

    public SimpleStringProperty dis() {
        return new
SimpleStringProperty(String.valueOf("SANTILLANA"));
    }
    public SimpleStringProperty distrito() {
        return new
SimpleStringProperty(String.valueOf(this.getDis()));
    }
    public SimpleStringProperty provincia() {
        return new
SimpleStringProperty(String.valueOf(this.getPro()));
    }
}

```

class Propietario

```

package modelo;
// Generated 25/06/2019 08:56:13 PM by Hibernate Tools 4.3.1
import java.util.HashSet;
import java.util.Set;
import javafx.beans.property.SimpleStringProperty;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.FetchType;
import javax.persistence.GeneratedValue;
import static javax.persistence.GenerationType.IDENTITY;
import javax.persistence.Id;
import javax.persistence.OneToOne;
import javax.persistence.Table;
/**
 * Propietario generated by hbm2java
 */
@Entity
@Table(name = "propietario",
        catalog = "autovaluo"
)
public class Propietario implements java.io.Serializable {
    private Integer idpro;
    private String nombre;
    private String numdoc;
    private String tipodoc;
    private String celular;
    private String correo;
    private String obs;
    private Set<Propiedad> propiedades = new
HashSet<Propiedad>(0);
    public Propietario() {
    }
    public Propietario(String celular, String correo, String
obs) {
        this.celular = celular;
        this.correo = correo;
        this.obs = obs;
    }
    public Propietario(String nombre, String numdoc, String
tipodoc, String celular, String correo, String obs,
Set<Propiedad> propiedades) {
        this.nombre = nombre;

```

```

        this.numdoc = numdoc;
        this.tipodoc = tipodoc;
        this.celular = celular;
        this.correo = correo;
        this.obs = obs;
        this.propiedades = propiedades;
    }
    @Id
    @GeneratedValue(strategy = IDENTITY)

    @Column(name = "idpro", unique = true, nullable = false)
    public Integer getIdpro() {
        return this.idpro;
    }
    public void setIdpro(Integer idpro) {
        this.idpro = idpro;
    }
    @Column(name = "nombre", length = 60)
    public String getNombre() {
        return this.nombre;
    }
    public void setNombre(String nombre) {
        this.nombre = nombre;
    }
    @Column(name = "numdoc", length = 45)
    public String getNumdoc() {
        return this.numdoc;
    }
    public void setNumdoc(String numdoc) {
        this.numdoc = numdoc;
    }
    @Column(name = "tipodoc", length = 45)
    public String getTipodoc() {
        return this.tipodoc;
    }
    public void setTipodoc(String tipodoc) {
        this.tipodoc = tipodoc;
    }
    @Column(name = "celular", nullable = false, length = 40)
    public String getCelular() {
        return this.celular;
    }
    public void setCelular(String celular) {
        this.celular = celular;
    }
    @Column(name = "correo", nullable = false, length = 40)
    public String getCorreo() {
        return this.correo;
    }
    public void setCorreo(String correo) {
        this.correo = correo;
    }
    @Column(name = "obs", nullable = false, length = 120)
    public String getObs() {
        return this.obs;
    }
    public void setObs(String obs) {
        this.obs = obs;
    }
}

```

```

@OneToMany(fetch = FetchType.LAZY, mappedBy = "propietario")
public Set<Propiedad> getPropiedades() {
    return this.propiedades;
}
public void setPropiedades(Set<Propiedad> propiedades) {
    this.propiedades = propiedades;
}
public SimpleStringProperty nom() {
    return new
SimpleStringProperty(String.valueOf(this.getNombre()));
}
public SimpleStringProperty numdoc() {
    return new
SimpleStringProperty(String.valueOf(this.getNumdoc()));
}
public SimpleStringProperty tipdoc() {
    return new
SimpleStringProperty(String.valueOf(this.getTipodoc()));
}
}

```

class Puerta

```

package modelo;
// Generated 25/06/2019 08:56:13 PM by Hibernate Tools 4.3.1
import java.math.BigDecimal;
import java.util.HashSet;
import java.util.Set;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.FetchType;
import javax.persistence.GeneratedValue;
import static javax.persistence.GenerationType.IDENTITY;
import javax.persistence.Id;
import javax.persistence.OneToMany;
import javax.persistence.Table;
/**
 * Puerta generated by hbm2java
 */
@Entity
@Table(name="puerta"
, catalog="autovaluo"
)
public class Puerta implements java.io.Serializable {
    private Integer idpuerta;
    private String letra;
    private String descripcion;
    private BigDecimal monto;
    private Set<Predio> predios = new HashSet<Predio>(0);
    public Puerta() {
    }
    public Puerta(String letra, String descripcion, BigDecimal
monto) {
        this.letra = letra;
        this.descripcion = descripcion;
        this.monto = monto;
    }
}

```



```

    public Puerta(String letra, String descripcion, BigDecimal
monto, Set<Predio> predios) {
        this.letra = letra;
        this.descripcion = descripcion;
        this.monto = monto;
        this.predios = predios;
    }
    @Id @GeneratedValue(strategy=IDENTITY)
    @Column(name="idpuerta", unique=true, nullable=false)
    public Integer getIdpuerta() {
        return this.idpuerta;
    }
    public void setIdpuerta(Integer idpuerta) {
        this.idpuerta = idpuerta;
    }

    @Column(name="letra", nullable=false, length=15)
    public String getLetra() {
        return this.letra;
    }
    public void setLetra(String letra) {
        this.letra = letra;
    }
    @Column(name="descripcion", nullable=false, length=250)
    public String getDescripcion() {
        return this.descripcion;
    }
    public void setDescripcion(String descripcion) {
        this.descripcion = descripcion;
    }
    @Column(name="monto", nullable=false, precision=10)
    public BigDecimal getMonto() {
        return this.monto;
    }
    public void setMonto(BigDecimal monto) {
        this.monto = monto;
    }
    @OneToMany(fetch=FetchType.LAZY, mappedBy="puerta")
    public Set<Predio> getPredios() {
        return this.predios;
    }
    public void setPredios(Set<Predio> predios) {
        this.predios = predios;
    }
}

```

class Revestimiento

```

package modelo;
// Generated 25/06/2019 08:56:13 PM by Hibernate Tools 4.3.1
import java.math.BigDecimal;
import java.util.HashSet;
import java.util.Set;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.FetchType;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType.IDENTITY;

```

```

import javax.persistence.Id;
import javax.persistence.OneToMany;
import javax.persistence.Table;
/**
 * Revestimiento generated by hbm2java
 */
@Entity
@Table(name="revestimiento"
, catalog="autovaluo"
)
public class Revestimiento implements java.io.Serializable {
    private Integer idrevestimiento;
    private String letra;
    private String descripcion;
    private BigDecimal monto;
    private Set<Predio> predios = new HashSet<Predio>(0);
    public Revestimiento() {
    }
    public Revestimiento(String letra) {
        this.letra = letra;
    }
    public Revestimiento(String letra, String descripcion,
BigDecimal monto, Set<Predio> predios) {
        this.letra = letra;
        this.descripcion = descripcion;
        this.monto = monto;
        this.predios = predios;
    }
    @Id @GeneratedValue(strategy=IDENTITY)
    @Column(name="idrevestimiento", unique=true, nullable=false)
    public Integer getIdrevestimiento() {
        return this.idrevestimiento;
    }
    public void setIdrevestimiento(Integer idrevestimiento) {
        this.idrevestimiento = idrevestimiento;
    }

    @Column(name="letra", nullable=false, length=15)
    public String getLetra() {
        return this.letra;
    }
    public void setLetra(String letra) {
        this.letra = letra;
    }

    @Column(name="descripcion", length=250)
    public String getDescripcion() {
        return this.descripcion;
    }
    public void setDescripcion(String descripcion) {
        this.descripcion = descripcion;
    }
    @Column(name="monto", precision=10)
    public BigDecimal getMonto() {
        return this.monto;
    }
    public void setMonto(BigDecimal monto) {
        this.monto = monto;
    }
}

```

```

@OneToMany(fetch=FetchType.LAZY, mappedBy="revestimiento")
    public Set<Predio> getPredios() {
        return this.predios;
    }
    public void setPredios(Set<Predio> predios) {
        this.predios = predios;
    }
}

```

class Techo

```

package modelo;
// Generated 25/06/2019 08:56:13 PM by Hibernate Tools 4.3.1
import java.math.BigDecimal;
import java.util.HashSet;
import java.util.Set;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.FetchType;
import javax.persistence.GeneratedValue;
import static javax.persistence.GenerationType.IDENTITY;
import javax.persistence.Id;
import javax.persistence.OneToMany;
import javax.persistence.Table;
/**
 * Techo generated by hbm2java
 */
@Entity
@Table(name="techo"
, catalog="autovalor"
)
public class Techo implements java.io.Serializable {
    private Integer idtecho;
    private String letra;
    private String descripcion;
    private BigDecimal monto;
    private Set<Predio> predios = new HashSet<Predio>(0);
    public Techo() {
    }
    public Techo(String letra) {
        this.letra = letra;
    }
    public Techo(String letra, String descripcion, BigDecimal
monto, Set<Predio> predios) {
        this.letra = letra;
        this.descripcion = descripcion;
        this.monto = monto;
        this.predios = predios;
    }
    @Id @GeneratedValue(strategy=IDENTITY)

    @Column(name="idtecho", unique=true, nullable=false)
    public Integer getIdtecho() {
        return this.idtecho;
    }
    public void setIdtecho(Integer idtecho) {
        this.idtecho = idtecho;
    }
    @Column(name="letra", nullable=false, length=15)
    public String getLetra() {

```

```

        return this.letra;
    }
    public void setLetra(String letra) {
        this.letra = letra;
    }
    @Column(name="descripcion", length=250)
    public String getDescripcion() {
        return this.descripcion;
    }
    public void setDescripcion(String descripcion) {
        this.descripcion = descripcion;
    }
    @Column(name="monto", precision=10)
    public BigDecimal getMonto() {
        return this.monto;
    }
    public void setMonto(BigDecimal monto) {
        this.monto = monto;
    }
    @OneToMany(fetch=FetchType.LAZY, mappedBy="techo")
    public Set<Predio> getPredios() {
        return this.predios;
    }
    public void setPredios(Set<Predio> predios) {
        this.predios = predios;
    }
}

```

class Usuario

```

package modelo;
// Generated 25/06/2019 08:56:13 PM by Hibernate Tools 4.3.1
import javafx.beans.property.SimpleStringProperty;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import static javax.persistence.GenerationType.IDENTITY;
import javax.persistence.Id;
import javax.persistence.Table;
/**
 * Usuario generated by hbm2java
 */
@Entity
@Table(name = "usuario",
        catalog = "autovaluo"
)
public class Usuario implements java.io.Serializable {
    private Integer idusu;
    private String nombre;
    private String dni;
    private String usu;
    private String cont;
    public Usuario() {
    }
    public Usuario(String nombre, String dni, String usu, String
cont) {
        this.nombre = nombre;
        this.dni = dni;
    }
}

```

```

        this.usu = usu;
        this.cont = cont;
    }
    @Id
    @GeneratedValue(strategy = IDENTITY)
    @Column(name = "idusu", unique = true, nullable = false)
    public Integer getIdusu() {
        return this.idusu;
    }
    public void setIdusu(Integer idusu) {
        this.idusu = idusu;
    }
    @Column(name = "nombre", length = 60)
    public String getNombre() {
        return this.nombre;
    }
    public void setNombre(String nombre) {
        this.nombre = nombre;
    }
    @Column(name = "dni", length = 45)
    public String getDni() {
        return this.dni;
    }
    public void setDni(String dni) {
        this.dni = dni;
    }
    @Column(name = "usu", length = 60)
    public String getUsu() {
        return this.usu;
    }
    public void setUsu(String usu) {
        this.usu = usu;
    }
    @Column(name = "cont", length = 60)
    public String getCont() {
        return this.cont;
    }
    public void setCont(String cont) {
        this.cont = cont;
    }
    public SimpleStringProperty name() {
        return new
SimpleStringProperty(String.valueOf(this.getNombre()));
    }
    public SimpleStringProperty dni() {
        return new
SimpleStringProperty(String.valueOf(this.getDni()));
    }
}

```

class Valorpropiedad

```

package modelo;
// Generated 25/06/2019 08:56:13 PM by Hibernate Tools 4.3.1
import java.math.BigDecimal;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.FetchType;

```

```

import javax.persistence.GeneratedValue;
import static javax.persistence.GenerationType.IDENTITY;
import javax.persistence.Id;
import javax.persistence.JoinColumn;
import javax.persistence.ManyToOne;
import javax.persistence.Table;
/**
 * Valorpropiedad generated by hbm2java
 */
@Entity
@Table(name = "valorpropiedad",
        catalog = "autovaluo"
)
public class Valorpropiedad implements java.io.Serializable {
    private Integer idvalorpro;
    private Propiedad propiedad;
    private Integer ano;
    private BigDecimal areat;
    private BigDecimal arancel;
    private BigDecimal valort;
    public Valorpropiedad() {
    }
    public Valorpropiedad(Propiedad propiedad) {
        this.propiedad = propiedad;
    }
    public Valorpropiedad(Propiedad propiedad, Integer ano,
BigDecimal areat, BigDecimal arancel, BigDecimal valort) {
        this.propiedad = propiedad;
        this.ano = ano;
        this.areat = areat;
        this.arancel = arancel;
        this.valort = valort;
    }
    @Id
    @GeneratedValue(strategy = IDENTITY)
    @Column(name = "idvalorpro", unique = true, nullable =
false)
    public Integer getIdvalorpro() {
        return this.idvalorpro;
    }
    public void setIdvalorpro(Integer idvalorpro) {
        this.idvalorpro = idvalorpro;
    }
    @ManyToOne(fetch = FetchType.LAZY)
    @JoinColumn(name = "idprop", nullable = false)
    public Propiedad getPropiedad() {
        return this.propiedad;
    }
    public void setPropiedad(Propiedad propiedad) {
        this.propiedad = propiedad;
    }
    @Column(name = "ano")
    public Integer getAno() {
        return this.ano;
    }
    public void setAno(Integer ano) {
        this.ano = ano;
    }
    @Column(name = "areat", precision = 10)

```

```
public BigDecimal getAreat() {
    return this.areat;
}
public void setAreat(BigDecimal areat) {
    this.areat = areat;
}
@Column(name = "arancel", precision = 10)
public BigDecimal getArancel() {
    return this.arancel;
}
public void setArancel(BigDecimal arancel) {
    this.arancel = arancel;
}
@Column(name = "valort", precision = 10)
public BigDecimal getValort() {
    return this.valort;
}
public void setValort(BigDecimal valort) {
    this.valort = valort;
}
}
```

ANEXO 9. OPERACIONALIZACIÓN DE VARIABLES

OPERACIONALIZACIÓN DE VARIABLES		
VARIABLE	INDICADOR	ITEM
X: Sistema de Información de Recaudación Tributaria	X1: Impuesto Predio Urbano	¿Qué grado de beneficio alcanzaría la recaudación de impuesto predio urbano?
		¿Qué beneficio daría a la población al saber el valor del autovaluo de su vivienda?
		¿Cómo se define si conviene pagar impuesto predial urbano en forma mensual, trimestral o anualmente?
		¿Cómo el documento de predio urbano que se emite a título de propietario lo ayudaría para que pueda postular a techo propio o bien a otro programa?
	X2: Impuesto Predio Rural	¿Qué grado de beneficio alcanzaría la recaudación de impuesto predio rural o rustico?
		¿Cómo se define si conviene pagar impuesto predial rural o rustica en forma mensual, trimestral o anualmente?
		¿Cómo el documento de predio rural o rustico que se emite a título de propietario lo ayudaría para que pueda postular a techo propio o bien a otro programa?
		¿Cuánto velocidad y accesibilidad debe ser el sistema de recaudación tributaria de la Municipalidad?
Y: Jira Scrum Agile Software	Y1: Gestión regular de las expectativas del cliente	¿El cliente como se desenvuelve en todas las reuniones de la planificación de sprint?
		¿Cuánto expectativa tiene el cliente con el Scrum master y Scrum Team?
		¿Cuánto conformidad se sientes al finalizar de hacer las metas de cada reunión el cliente con Scrum master y el equipo de trabajo?
	Y2: Equipos auto dirigidos y auto organizados	¿El Scrum master cuanto liderazgo tiene?
		¿Cómo interactúan el equipo de trabajo con el proyecto de sistema de información de recaudación tributaria?
		¿El Scrum Team cuanto organización tiene?
		¿Producto Owner cuanto comunicación tiene con el Scrum master y el equipo de trabajo?

ANEXO 10. MATRIZ DE CONSISTENCIA

Título: “SISTEMA DE INFORMACIÓN DE RECAUDACIÓN TRIBUTARIA DE LA MUNICIPALIDAD DISTRITAL DE SANTILLANA BAJO EL MARCO JIRA SCRUM AGILE SOFTWARE, HUANTA, 2019”

PROBLEMA	OBJETIVO	MARCO TEÓRICO	VARIABLES	METODOLOGÍA DE INVESTIGACION
<p>PROBLEMA GENERAL</p> <p>¿Cómo el marco Jira Scrum Agile Software permite el desarrollo del Sistema Información de Recaudación Tributaria del impuesto predial de la Municipalidad distrital de Santillana, Huanta, 2019?</p> <p>PROBLEMAS ESPECIFICO</p> <p>a. ¿Cómo el marco Jira Scrum Agile Software realiza la gestión regular de las expectativas del</p>	<p>OBJETIVO GENERAL</p> <p>Aplicar el marco Jira Scrum Agile Software que nos permite el desarrollo del Sistema Información de Recaudación Tributaria del impuesto predial de la Municipalidad distrital de Santillana, Huanta, 2019.</p> <p>OBJETIVOS ESPECÍFICOS</p> <p>C. Aplicar el marco Jira Scrum Agile Software para la realización de la gestión regular de las expectativas del cliente que nos permita el desarrollo del Sistema Información de Recaudación</p>	<p>ANTECEDENTES</p> <p>Según Javier E. (2017), en su proyecto de grado denominado “Kanban y Scrumbar Orientados a Proyectos de Tecnología de la Información”, del Instituto Universitario Aeronáutico, define que Jira es un producto cuyo uso está extendido en todo tipo de compañías a lo largo del mundo, tanto tecnológicas como cualquier otra que requiera de una gestión de proyectos, procesos, incidencias, defectos y/o actividades. En este sentido, Jira también es un facilitador de comunicaciones y de seguimiento de trabajo fundamental para agilizar los procesos en un flujo de trabajo. La herramienta permite definir y hacer seguimiento a proyectos basados en cualquier de los marcos de trabajo Scrum, Kanban y Scrumban, proporcionando funcionalidades como: Creación de historias de usuario, estimación de las historias en la planificación de la iteración, visualización de la actividad del equipo, reporte de progreso, entre otros.</p> <p>Pilar R. y Juan G González (2008), en su Tesis de Máster "Estudio de la Aplicación de Metodologías Ágiles para la Evolución de Productos Software", en la Universidad Politécnica de Madrid,</p>	<p>VARIABLE DE INTERÉS</p> <p>X: Sistema de Información de Recaudación Tributaria.</p> <p>INDICADOR</p> <p>X1: Impuesto Predio Urbano. X2: Impuesto Predio Rural.</p> <p>VARIABLE DE ESTUDIO</p> <p>Y: Jira Scrum Agile Software</p> <p>INDICADOR</p> <p>Y1: Gestión regular de las expectativas del cliente.</p>	<p>TIPO DE INVESTIGACIÓN</p> <p>la investigación es de tipo observacional, retrospectivo y transversal.</p> <p>NIVEL DE INVESTIGACIÓN</p> <p>nivel de investigación es Descriptiva</p> <p>POBLACIÓN</p> <p>La población de estudio estará compuesta por todas las metodologías ágiles de desarrollo y gestión de software ágil.</p>

<p>cliente en el desarrollo del Sistema Información de Recaudación Tributaria del impuesto predial de la Municipalidad distrital de Santillana, Huanta, 2019?</p> <p>b. ¿Cómo el marco Jira Scrum Agile Software gestiona los equipos auto dirigidos y auto organizados en el desarrollo del Sistema Información de Recaudación Tributaria del impuesto predial de la Municipalidad distrital de Santillana, Huanta, 2019?</p>	<p>Tributaria del impuesto predial de la Municipalidad distrital de Santillana, Huanta, 2019.</p> <p>D. Aplicar el marco Jira Scrum Agile Software para la gestión de los equipos auto dirigidos y auto organizados que nos permitan el desarrollo del Sistema Información de Recaudación Tributaria del impuesto predial de la Municipalidad distrital de Santillana, Huanta, 2019.</p>	<p>concluye que la metodología SRCUM, es un proceso para la gestión y control del producto que trata de eliminar la complejidad en estas áreas para centrarse en la construcción de software que satisfaga las necesidades del negocio. Es simple y escalable, ya que no establece prácticas de ingeniería del software, sino que se aplica o combina, fácilmente, con otras prácticas ingenieriles, metodologías de desarrollo o estándares ya existentes en la organización.</p> <p>MARCO TEÓRICO</p> <p>SISTEMA DE INFORMACIÓN DE RECAUDACIÓN TRIBUTARIA</p> <p>Verónica, C. (2015), En su tesis “Implementación de un sistema de información para la recaudación del impuesto de alcabala ‘sipria’ de la municipalidad provincial de puno - 2015”, define:</p> <p>Es un conjunto de elementos que interactúan entre sí con el fin de apoyar las actividades de una empresa o negocio. Teniendo muy en cuenta el equipo computacional necesario para que el sistema de información pueda operar y el recurso humano que interactúa con el Sistema de Información, el cual está formado por las personas que utilizan el sistema. (p.9)</p> <p>TRIBUTO MUNICIPAL</p> <p>Marco normativa. (2015). Define que “tributario municipal es el conjunto de tributos (impuestos, tasas y contribuciones) cuya administración se encuentra a cargo de las municipalidades, sean éstas de ámbito provincial o distrital.</p>	<p>Y2: Equipos auto dirigidos y auto organizados.</p>	<p>MUESTRA</p> <p>Se tomó como muestra el Jira Scrum Agile Software para la gestión de desarrollo de software ágil.</p> <p>TÉCNICA</p> <p>Según (Bernal, C., 2006), la investigación descriptiva se soporta principalmente en técnicas como la encuesta, la entrevista, la observación y la revisión documental. En definición en nuestra investigación se utiliza la técnica de entrevista tomando como cuenta las reuniones que se da con el representante (secretaría) de la Municipalidad Distrital de Santillana y también se utilizó el análisis documental de parte legal y jira Scrum existentes en el mercado de desarrollo de software para obtener un logro comparativo acerca del estudio.</p> <p>INSTRUMENTOS</p>
--	--	---	--	---

		<p>Asimismo, forman parte de este sistema las normas tributarias municipales y las entidades municipales encargadas de administrar tales tributos” (P.17).</p> <p>AUTOVALUO</p> <p>Según Mendoza (2019). Un autoavalúo, también conocido como autoavalúo, es un proceso formal mediante el cual se estima el valor de una propiedad. Ello se hace con la única finalidad de establecer la base de cálculo de algunos impuestos municipales. A dicha base impositiva se le denomina Valor Total de Autoavalúo, o también base imponible.</p> <p>IMPUESTO PREDIAL</p> <p>Ley tributo municipal (1993), en su artículo 8, establece que “el Impuesto Predial es de periodicidad anual y grava el valor de los predios urbanos y rústicos. Para efectos del Impuesto se considera predios a los terrenos, incluyendo los terrenos ganados al mar, a los ríos y a otros espejos de agua, así como las edificaciones e instalaciones fijas y permanentes que constituyan partes integrantes de dichos predios, que no pudieran ser separadas sin alterar, deteriorar o destruir la edificación”. Entonces una propiedad predial es un tipo de bien que se consideran como tales como a los terrenos, casas, edificios, etc. y grava a esto el valor periódicamente el impuesto predial.</p> <p>IMPUESTO PREDIO URBANO</p> <p>Reglamento nacional de tasación del Perú (2007), artículo IL.a.02, considera al predio urbano al que “está situado en</p>		<p>Según el apartado anterior, el instrumento guía es la entrevista de las reuniones que se desarrolla con el personal representante de la Municipalidad Distrital de Santillana, con el fin de generar información adecuada para el marco de trabajo del desarrollo del sistema de información de recaudación tributaria durante el tiempo que dura, la generación de información se encuentra en el apartado del anexo 1, anexo 2 y anexo 3. La revisión de los documentos legales se encuentra en el anexo 4 y anexo 5.</p>
--	--	---	--	--

		<p>centro poblado y se destine a vivienda, comercio, industria o cualquier otro fin urbano; así como los terrenos sin edificar, siempre que cuenten con los servicios generales propios del centro poblado y los que tengan terminadas y recibidas sus obras de habilitación urbana, estés o no habilitadas legamente”.</p> <p>IMPUESTO PREDIO RURAL</p> <p>Reglamento nacional de tasación del Perú (2007), artículo III.a.01, considera al predio rural o rustico a los “terrenos ubicados en zona rural dedicados a uso agrícola, pecuaria, forestal y de protección y a los eriazos susceptibles de destinarse a dichos usos que no hayan habilitados como urbanos ni estén comprendidos dentro de los límites de expansión urbana. Forman parte del predio rustico, la tierra, el agua y su ecosistema, así como los construcciones e instalaciones fijas y permanentes que existan en él”.</p> <p>JIRA SCRUM AGILE SOFTWARE</p> <p>David, H. (2018), define que:</p> <p>Ha estado presente por un tiempo; originalmente, era un sistema de emisión de tickets (un tipo de software de seguimiento de errores). La gestión de proyectos ha evolucionado a lo largo de los años, y los procesos ágiles se han vuelto cada vez más populares, por lo que JIRA se ha convertido en una herramienta de gestión de proyectos ágil muy efectiva tanto para Scrum como para Kanban. Ahora es muy poderoso en ese sentido. (p.22)</p>		
--	--	--	--	--

		<p>GESTIÓN REGULAR DE LAS EXPECTATIVAS DEL CLIENTE</p> <p>Gestión regular de las expectativas del cliente, basada en resultados tangibles, es una de las ventajas o beneficios de Scrum, según Karenny, B. (2009), define:</p> <p>El cliente establece sus expectativas indicando el valor que le aporta cada requisito del proyecto y cuando espera que esté completado; y comprueba de manera regular si se van cumpliendo sus expectativas, da feedback, ya desde el inicio del proyecto puede tomar decisiones informadas a partir de resultados objetivos y dirige estos resultados del proyecto, iteración a iteración, hacia su meta. (p.76)</p> <p>EQUIPOS AUTO DIRIGIDOS Y AUTO ORGANIZADOS</p> <p>También llamado equipo Scrum (Scrum Team), según Ken S. y Jeff S. (2013), define:</p> <p>Los Equipos Scrum son auto organizados y multifuncionales. Los equipos auto organizados eligen la mejor forma de llevar a cabo su trabajo y no son dirigidos por personas externas al equipo. Los equipos multifuncionales tienen todas las competencias necesarias para llevar a cabo el trabajo sin depender de otras personas que no son parte del equipo. El modelo de equipo en Scrum está diseñado para optimizar la flexibilidad, la creatividad y la productividad. Los Equipos Scrum entregan productos de forma iterativa e incremental, maximizando las oportunidades de obtener retroalimentación. Las entregas incrementales de producto “Terminado” aseguran que siempre</p>		
--	--	---	--	--

		<p>estará disponible una versión potencialmente útil y funcional del producto. (p. 6)</p> <p>ACTIVIDADES</p> <p>Las actividades y artefactos de SCRUM según Kenneth S. (2012) describe que, el propietario del producto tiene una visión de lo que quiere crear (el gran cubo). Debido a que el cubo puede ser grande, a través de una actividad llamada preparación , se divide en un conjunto de características que se recopilan en una lista priorizada llamada la acumulación de productos</p> <p>ARTEFACTOS</p> <p>Es una lista ordenada de todo lo que podría ser necesario en el producto, y es la única fuente de requisitos para cualquier cambio a realizarse en el producto. La lista de producto enumera todas las características, funcionalidades, requisitos, mejoras y correcciones que constituyen cambios a ser hechos sobre el producto para entregas futuras. Los elementos de la lista de producto tienen como atributos la descripción, la ordenación, la estimación y el valor (Schwaber y Sutherland, 2013).</p>		
--	--	---	--	--

**UNSCH**FACULTAD DE
INGENIERÍA
DE MINAS, GEOLOGÍA Y CIVIL**“Año del Bicentenario del Perú: 200 años de Independencia”****ACTA DE SUSTENTACIÓN DE TESIS N° 006-2021-FIMGC**

En la ciudad de Ayacucho, en cumplimiento a la **RESOLUCIÓN DECANAL N° 257-2021-FIMGC-D**, siendo los veinticinco días del mes de junio del 2021, a horas 4:00 p.m.; se reunieron los jurados del acto de sustentación, en el Auditorium virtual google meet del Campus Universitario de la Universidad Nacional de San Cristóbal de Huamanga.

Siendo el Jurado de la sustentación de tesis compuesto por el presidente el **Dr. Ing. Efraín Elías PORRAS FLORES** Jurado-Asesor el **Mg. Ing. Hubner JANAMPA PATILLA**. Jurado **Ing. Karel PERALTA SOTOMAYOR**. Jurado **Mg. Ing. Christian LEZAMA CUÉLLAR** y secretario del proceso el **Ing. Richar Rubén JORGE BERROCAL**, con el objetivo de recepcionar la sustentación de la tesis denominada **“SISTEMA DE INFORMACIÓN DE RECAUDACIÓN TRIBUTARIA DE LA MUNICIPALIDAD DISTRITAL DE SANTILLANA BAJO EL MARCO JIRA SCRUM AGILE SOFTWARE, HUANTA, 2019”**; sustentado por el Sr: **Yonny HUARANCCA ÑAUPARI** bachiller en Ingeniería de Sistemas.

El Jurado luego de haber recepcionado la sustentación de la tesis y realizado las preguntas, el sustentante al haber dado respuesta a las preguntas, y el Jurado haber deliberado; califica con la nota aprobatoria de **15 (QUINCE)**.

En fe de lo cual, se firma la presente acta, por los miembros integrantes del proceso de sustentación.

Firmado digitalmente por Dr.
Ing. Efraín Elías Porras Flores
Fecha: 2021.07.06 09:58:48
-05'00'**Dr. Ing. Efraín Elías PORRAS FLORES**
Presidente**Ing. Karel PERALTA SOTOMAYOR**
Jurado**Ing. Hubner Janampa Patilla****Mg. Ing. Hubner JANAMPA PATILLA**
Jurado-Asesor**Mg. Ing. Christian LEZAMA CUÉLLAR**
Jurado**Ing. Richar Rubén JORGE BERROCAL**

Secretario del Proceso

C.C.:
Bach. Yonny HUARANCCA ÑAUPARI
Jurados (4)
ArchivoFACULTAD DE INGENIERÍA
DE MINAS, GEOLOGÍA Y CIVIL
Av. Independencia S/N
Ciudad Universitaria
Central Tel 066 312510



UNSCH

FACULTAD DE
INGENIERÍA
DE MINAS, GEOLOGÍA Y CIVIL

CONSTANCIA DE ORIGINALIDAD DE TRABAJO DE INVESTIGACIÓN

El que suscribe; responsable verificador de originalidad de trabajos de tesis de pregrado en segunda instancia para las Escuelas Profesionales de la Facultad de Ingeniería de Minas, Geología y Civil; en cumplimiento a la Resolución de Consejo Universitario N° 039-2021-UNSCH-CU, Reglamento de Originalidad de Trabajos de Investigación de la UNSCH y Resolución Decanal N° 158-2021-FIMGC-UNSCH-D, deja constancia que:

- Apellidos y Nombres del Bach. : HUARANCCA ÑAUPARI YONNY
- Escuela Profesional : Ingeniería De Sistemas
- Título de la Tesis : Sistema de información de recaudación tributaria de la municipalidad distrital de Santillana bajo el marco jira SCRUM AGILE software, Huanta, 2019.
- Evaluación de la originalidad : 15 % de similitud

Por tanto, según los artículos 12, 13 y 17 del Reglamento de Originalidad de Trabajos de Investigación, **es procedente otorgar la constancia de originalidad** para los fines que crea conveniente.

Ayacucho, 15 de junio del 2021



Firmado digitalmente
por Mg. Ing. Johnny
Henry Ccatamayo
Barrios
Fecha: 2021.06.15
07:54:55 -05'00'

Mg. Ing. Ccatamayo Barrios Johnny Henry
Verificador de originalidad de trabajos de tesis de pregrado de la FIMGC

Numero de constancia: 058-2021-FIMGC.

SISTEMA DE INFORMACIÓN DE RECAUDACIÓN TRIBUTARIA DE LA MUNICIPALIDAD DISTRITAL DE SANTILLANA BAJO EL MARCO JIRA SCRUM AGILE SOFTWARE, HUANTA, 2019

por Yonny Huarancca Ñaupari

Fecha de entrega: 15-jun-2021 06:36a.m. (UTC-0500)

Identificador de la entrega: 1606879758

Nombre del archivo: Tesis_Yonny_Huarancca.docx (18.48M)

Total de palabras: 51999

Total de caracteres: 343146

SISTEMA DE INFORMACIÓN DE RECAUDACIÓN TRIBUTARIA DE LA MUNICIPALIDAD DISTRITAL DE SANTILLANA BAJO EL MARCO JIRA SCRUM AGILE SOFTWARE, HUANTA, 2019

INFORME DE ORIGINALIDAD

15%

INDICE DE SIMILITUD

5%

FUENTES DE INTERNET

1%

PUBLICACIONES

11%

TRABAJOS DEL ESTUDIANTE

FUENTES PRIMARIAS

1	Submitted to Universidad Nacional de San Cristóbal de Huamanga Trabajo del estudiante	10%
2	es.scribd.com Fuente de Internet	1%
3	Renata Brasil Silva. "Análise empírica da influência da experiência do desenvolvedor na degradação da arquitetura de software em desenvolvimento open source.", Universidade de Sao Paulo, Agencia USP de Gestao da Informacao Academica (AGUIA), 2020 Publicación	1%
4	caminolibre.com Fuente de Internet	1%
5	www.scribd.com Fuente de Internet	1%
6	Submitted to tec Trabajo del estudiante	<1%

7	cristiancastrop.files.wordpress.com Fuente de Internet	<1 %
8	jesusp.net Fuente de Internet	<1 %
9	lossistemasdeinfoempresas.blogspot.com Fuente de Internet	<1 %
10	portal.gasnatural.com Fuente de Internet	<1 %
11	www.mysql-hispano.org Fuente de Internet	<1 %
12	galenoscopio.org Fuente de Internet	<1 %
13	Submitted to Escuela Superior Politécnica Agropecuaria de Manabí Manuel Félix López Trabajo del estudiante	<1 %
14	www.huanuco.hco.mp.gob.pe Fuente de Internet	<1 %
15	moam.info Fuente de Internet	<1 %
16	www.inicam.org.pe Fuente de Internet	<1 %

Excluir bibliografía Activo