

**UNIVERSIDAD NACIONAL DE SAN CRISTOBAL DE
HUAMANGA**

**FACULTAD DE INGENIERIA DE MINAS, GEOLOGIA Y
CIVIL**

**ESCUELA DE FORMACION PROFESIONAL DE INGENIERIA DE
SISTEMAS**



**APLICACION WEB PARA TELEFONOS INTELIGENTES,
EN EL CATASTRO DE USUARIO DE AGUA POTABLE DE
LA ENTIDAD PRESTADORA DE SERVICIOS DE
SANEAMIENTO AYACUCHO EPSASA, 2012**

TESIS PRESENTADO POR:

BACH. JERONIMO MEDINA HUAMAN

**TESIS PARA OPTAR EL TITULO PROFESIONAL DE
INGENIERO DE SISTEMAS**

ASESORA: ING. EDITH F. GUEVARA MOROTE

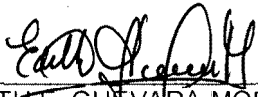
AYACUCHO-PERU

2012

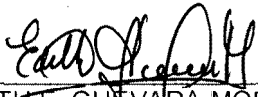
“APLICACIÓN WEB PARA TELÉFONOS INTELIGENTES, EN EL CATASTRO DE USUARIOS DE AGUA POTABLE DE LA ENTIDAD PRESTADORA DE SERVICIOS DE SANEAMIENTO AYACUCHO EPSASA, 2012”

RECOMENDADO: 01 DE AGOSTO DEL 2012


APROBADO : 09 DE AGOSTO DEL 2012



Ing° Dr. JAIME A. HUAMÁN MONTES
(Presidente)



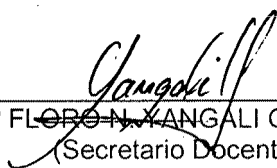
Ing° EDITH F. GUEVARA MOROTE
(Miembro)



Ing° ELINAR CARRILLO RIVEROS
(Miembro)

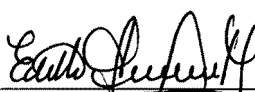


Ing° JUAN C. CARREÑO GAMARRA
(Miembro)

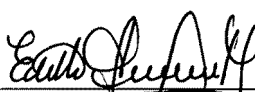


Ing° FLOREN YANGALI GUERRA
(Secretario Docente)

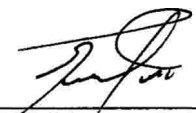
Según el acuerdo constatado en el Acta, levantada el 09 de agosto del 2012, en la sustentación de Tesis presentado por el Bachiller en Ingeniería de Sistemas Sr. Jerónimo MEDINA HUAMÁN, con el trabajo titulado: "APLICACIÓN WEB PARA TELÉFONOS INTELIGENTES, EN EL CATASTRO DE USUARIOS DE AGUA POTABLE DE LA ENTIDAD PRESTADORA DE SERVICIOS DE SANEAMIENTO AYACUCHO EPSASA, 2012", fue calificado con la nota de QUINCE (15) por lo que se da la respectiva APROBACIÓN.



Ing° Dr. JAIME A. HUAMÁN MONTES
(Presidente)



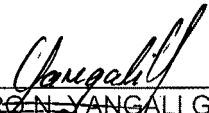
Ing° EDITH F. GUEVARA MOROTE
(Miembro)



Ing° ELINAR CARRILLO RIVEROS
(Miembro)



Ing° JUAN C. CARREÑO GAMARRA
(Miembro)



Ing° FLORÓN YANGALI GUERRA
(Secretario Docente)

DEDICATORIA

A mi padre Benjamín Medina Candia por haber sido un gran amigo. Una vida es poco para todas las cosas que me hubiera gustado vivir contigo.

AGRADECIMIENTO

A mis maestros por haber compartido sus conocimientos, también haberme guiado como profesional y sobre todo como persona.

CONTENIDO

	Página
DEDICATORIA	i
AGRADECIMIENTO.....	ii
CONTENIDO	iii
RESUMEN.....	vi
INTRODUCCIÓN	vii

CAPÍTULO I

PROBLEMA DE INVESTIGACIÓN

1.1 DIAGNOSTICO Y ENUNCIADO DEL PROBLEMA.....	1
1.2 DEFINICIÓN DEL PROBLEMA	2
1.3 OBJETIVOS DE LA INVESTIGACIÓN	3
1.3.1 OBJETIVO GENERAL.....	3
1.3.2 OBJETIVOS ESPECÍFICOS	3
1.4 HIPÓTESIS DE LA INVESTIGACIÓN	4
1.5 JUSTIFICACIÓN Y DELIMITACIÓN DE LA INVESTIGACIÓN	4
1.5.1 IMPORTANCIA Y JUSTIFICACIÓN.....	4
1.5.2 DELIMITACIÓN.....	5

CAPÍTULO II

MARCO DE REFERENCIA DE LA INVESTIGACIÓN

2.1 ANTECEDENTES.....	6
-----------------------	---

2.2	MARCO TEÓRICO.....	8
2.2.1	EPSASA.....	8
2.2.2	CATASTRO DE USUARIO.....	9
2.2.3	MANIFIESTO ÁGIL.....	16
2.2.4	METODOLOGIA SCRUM.....	18
2.2.5	ESTIMACIÓN Y PLANIFICACIÓN.....	26
2.2.6	DESARROLLO ÁGIL DE SOFTWARE.....	32
2.2.7	TECNOLOGÍA.....	34

CAPITULO III

METODOLOGÍA DE LA INVESTIGACIÓN

3.1	TIPO DE INVESTIGACIÓN.....	40
3.2	DISEÑO DE LA INVESTIGACIÓN.....	40
3.3	POBLACIÓN Y MUESTRA.....	41
3.4	VARIABLES E INDICADORES.....	42
3.5	TÉCNICAS E INSTRUMENTOS.....	43
3.5.1	INSTRUMENTO PARA RECOLECTAR INFORMACIÓN.....	43
3.5.2	HERRAMIENTAS PARA TRATAMIENTO DE DATOS E INFORMACIÓN.....	44
3.5.3	TÉCNICAS PARA APLICAR SCRUM.....	45

CAPITULO IV

RESULTADOS DE LA INVESTIGACIÓN

4.1 PROCESOS DEFINIDOS POR METODOLOGIA SCRUM	49
4.1.1 PLANINNG MEETING	49
4.1.2 RELASE PLAN	50
4.1.3 SPRINT 1	67
4.1.4 SPRINT 2	79
4.1.5 SPRINT 3	102

CAPITULO V

CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES	118
5.2 RECOMENDACIONES	119
BIBLIOGRAFÍA.....	120
ANEXO A.....	122
ANEXO B.....	124
ANEXO C.....	126
ANEXO D.....	127
ANEXO E.....	129
ANEXO F.....	131
ANEXO G.....	133
ANEXO H.....	134

RESUMEN

El objetivo principal ha sido desarrollar una aplicación web para teléfonos inteligentes (smartphones), que permite actualizar permanentemente el catastro de usuario de agua potable de la Entidad Prestadora de Servicio de Saneamiento EPSASA, 2012. Se empleó la metodología ágil Scrum que nos permite gestionar el proyecto, también el framework de desarrollo Rhodes y tecnologías de Internet, con el propósito de mejorar el nivel de recaudación por consumo de agua potable y con la finalidad de actualizar en tiempo real el catastro de usuario de agua potable.

Scrum con sus procesos como la planificación meeting (modelado de roles, identificación de requerimientos y escritura de historia de usuario), release plan y el sprint (daily meeting, desarrollo ágil, review meeting, retrospectiva) nos permitió generar artefactos.

El resultado de la investigación es una aplicación web para teléfonos inteligentes que se llegó a implementar en el Sprint 1 la construcción de la plataforma de desarrollo y las funcionalidades de localización del predio, en el Sprint 2 se implementó las funcionalidades de características de conexión y en el Sprint 3 se implementó las funcionalidades de uso de predio y administración de usuarios.

Palabras claves

Aplicación Web, Telefonos Inteligentes, Catastro, Usuario de Agua Potable, EPSASA, Scrum, Rhodes.

INTRODUCCION

La investigación que se realizó es para mantener la actualización permanente del catastro de usuarios de la Entidad Prestadora de Servicio de Saneamiento EPSASA, busca simplificar y hacer más rápida la tarea de localización de la conexión, de la obtención de las características de conexión y la identificación del predio, creando funcionalidades específicas en la aplicación web para teléfonos inteligentes.

El catastro de usuarios comprende el conjunto de registros y procedimientos que permiten la exacta identificación y localización de los usuarios de los servicios de agua potable y desagüe, posee toda la información necesaria de los usuarios activos, factibles, potenciales y clandestinos. La metodología Scrum es un proceso ágil que nos permite centrarnos en ofrecer el más alto valor de negocio en el menor tiempo y a la vez nos permite rápidamente y en repetidas ocasiones inspeccionar software real de trabajo (cada dos semanas o un mes). Scrum se considera “una manera simple de manejar problemas complejos”, proporcionando un marco de trabajo para soportar la innovación y permitir que equipos auto-organizados entreguen resultados de alta calidad en tiempos cortos. Rhodes es un framework open source de Rhomobile, se puede desarrollar aplicaciones nativas para casi todos los smartphones. Las aplicaciones construidas a través de Rhodes son puras aplicaciones nativas y se puede utilizar las capacidades del dispositivo, tales como GPS, contactos PIM y el calendario, cámara, native mapping, push, barcode, signature capture, and Bluetooth.

Los problemas planteados son; ¿De qué manera localizar la conexión de un

usuario de agua potable?, ¿Qué características de conexión se debe de registrar para un usuario de agua potable?, ¿Qué tipo de servicio de agua potable se da en el predio?

Los objetivos específicos son; implementar la localización de la conexión de un usuario de agua potable, con la finalidad de geolocalizar automáticamente y en tiempo real. Registrar las características de conexión para un usuario de agua potable, con la finalidad de dar mantenimiento a las características de conexiones. Identificar e implementar los usos de los predios que realiza el usuario de agua potable a fin de brindar información para la facturación del consumo de agua potable.

El trabajo consta de cinco capítulos: el primer capítulo se refiere al planteamiento del problema, sus objetivos, la hipótesis; el segundo capítulo se refiere al marco de referencia en la que se tratan las bases teóricas se analiza la literatura sobre las principales variables que son catastro y usuario de agua potable, así como el marco conceptual; en el tercer capítulo definimos el tipo de investigación, diseño de la investigación, población y muestra, variables e indicadores, instrumentos técnicos para aplicar Scrum, en el cuarto capítulo se implemento obteniendo los artefactos de los procesos y reuniones que nos da Scrum y finalizando en el quinto capítulo se determino las conclusiones y recomendaciones.

CAPITULO I

PROBLEMA DE INVESTIGACION

1.1 DIAGNOSTICO Y ENUNCIADO DEL PROBLEMA

El ámbito de jurisdicción de la Entidad Prestadora de Servicio de Saneamiento EPSASA se encuentra dentro de la región Ayacucho, específicamente atiende a dos provincias Huamanga con su capital Ayacucho y a Huanta, que cuentan con una población de 221,469 y 93,360 respectivamente. En el año 2009 se llegó a tener hasta 84% de cobertura de abastecimiento de agua potable. El número de conexiones de agua potable incrementó del 2008 al 2009 en 10000 y se hace esfuerzos en aumentar reapertura de medidores, implementación de lecturas por red privada movistar, al cambio de medidores inoperativos, EPSASA (2010).

El área de Catastro de la entidad prestadora de servicios de saneamiento EPSASA, el año 2010 invirtió la suma de S/. 270 000,00 en la actualización del catastro de usuario en el cual se realizó los siguientes procedimientos: levantamiento, procesamiento y utilización adecuada de información catastral de usuario. Lo que esto nos indica que es muy elevado el precio para poder realizar la actualización permanentemente.

Cuando se tiene actualizado el catastro de usuario de agua potable, se tiene el control de clandestinidad, servicios inactivos y uso de los predios. El beneficio es que se va a mejorar el nivel de recaudación y así invertir en proyectos que sirvan para aumentar el nivel de cobertura de servicio de agua potable.

Por lo expuesto, es muy necesario utilizar equipos tecnológicos que tenga la característica de localización y manejo de información para poder actualizar en tiempo real la información requerida por catastro de usuario, y a la vez utilizar formas ágiles de desarrollar software de manera que brinde valor al negocio.

1.2 DEFINICION DEL PROBLEMA

PROBLEMA PRINCIPAL

¿Cómo mantener actualizado el catastro de usuarios de agua potable de la entidad prestadora de saneamiento EPSASA, 2012?

PROBLEMAS SECUNDARIOS

- a. ¿De qué manera localizar la conexión de un usuario de agua potable?
- b. ¿Qué características de conexión se debe de registrar para un usuario de agua potable?
- c. ¿Qué tipo de servicio de agua potable se da en el predio?

1.3 OBJETIVOS DE LA INVESTIGACIÓN

1.3.1 OBJETIVO GENERAL

Desarrollar una aplicación web para teléfonos inteligentes (Smartphones),

que permite actualizar permanentemente el catastro de usuario de agua potable de la Entidad Prestadora de Servicio de Saneamiento EPSASA, 2012. Mediante la metodología ágil Scrum que nos ayudará a gestionar el proyecto, también con un framework de desarrollo Rhodes y tecnologías de Internet, con el propósito de mejorar el nivel de recaudación por consumo de agua potable y con la finalidad de actualizar en tiempo real el catastro de usuario de agua potable.

1.3.2 OBJETIVOS ESPECÍFICOS

- a. Implementar la localización de la conexión de un usuario de agua potable, con la finalidad de geolocalizar automáticamente y en tiempo real.
- b. Registrar las características de conexión para un usuario de agua potable, con la finalidad de dar mantenimiento a las características de conexiones.
- c. Identificar e implementar el tipo de servicio de los predios que realiza el usuario de agua potable a fin de brindar información para la facturación del consumo de agua potable.

1.4 HIPOTESIS

1.4.1. HIPOTESIS DE LA INVESTIGACION

Si implementamos la aplicación web para teléfonos inteligentes entonces se tendrá la actualización en tiempo real del catastro de usuario de agua potable de la entidad prestadora de servicio de saneamiento EPSASA, 2012.

1.5 JUSTIFICACION Y DELIMITACION DE LA INVESTIGACION

1.5.1 IMPORTANCIA Y JUSTIFICACION

El crecimiento de la población y el precio elevado en el levantamiento, procesamiento y utilización adecuada de información catastral de usuario de agua potable, ha originado la necesidad de contar con una aplicación web para móviles inteligentes, que pueda resolver la problemática existente como la localización de la conexión, la obtención de las características de conexión y la identificación del predio. Y así mejorar el desempeño funcional, el nivel de recaudación y reducción de la clandestinidad.

La aplicación web para móviles inteligentes tiene la capacidad de localizar a los predios, así podrá sincronizar información del cliente con el servidor central de catastro para poder actualizarlo en tiempo real.

La investigación desarrollada es para apoyar en la actualización permanente de los usuarios de la entidad prestadora de servicio de saneamiento EPSASA, simplificar y hacer más rápida la tarea de localización de la conexión, de la obtención de las características de conexión y la identificación del predio, creando funcionalidades específicas en la aplicación web para móviles inteligentes.

1.5.2 DELIMITACION

La investigación se realizará en el área de catastro de la entidad prestadora de servicios de saneamiento EPSASA, utilizando una metodología ágil, con la finalidad de actualizar permanentemente el catastro de usuario de agua potable, identificando las variables de investigación para el año 2012.

CAPÍTULO II

MARCO DE REFERENCIA DE LA INVESTIGACIÓN

2.1. ANTECEDENTES

Según Santana J. Camargo S (2011), El desarrollo de nuevas tecnologías ha dado la posibilidad de realizar ciertas aplicaciones, para dar soluciones a la falta de cartografía creciente, de alta precisión y disponibilidad, prescindiendo de grandes empresas satelitales o aerografías; mediante metodologías que propongan un bajo coste y alta portabilidad a cualquier lugar.

Con el planteamiento realizado se ha conseguido diseñar un modelo que permite generar y actualizar cartografía, desde la información disponible (imágenes, mapas, SIG), que puede ser adaptado a nuevas técnicas que se desarrollen y apliquen (imágenes satelitales, imágenes Google Earth, imágenes o fotogramas del IGN o mapas vectoriales).

Gianfelici, Esteban (2011). Nos expresa que el sistema de información catastral de agua potable y alcantarillado es una aplicación de información espacial y alfanumérica a través de la intranet e internet. Esta aplicación tiene la funcionalidad propia del SIG que permite capturar, compartir, almacenar, procesar, actualizar y

analizar la información espacial, de este modo es útil para la toma de decisiones por las oficinas encargadas de la administración de EMSA Puno. La ventaja de manejar datos alfanuméricos y geográficos de una manera compacta y ordenada, permite efectuar consultas sobre los datos que se pueden reflejar en los elementos geográficos presentes en la red de agua potable y alcantarillado, permitiendo obtener los resultados de una manera rápida, fácil y al alcance de la mano.

Según Héctor Zárate Rea (2011). Los dispositivos móviles de hoy como teléfonos, tablets o iPods son el último capítulo de una larga historia de distintas tecnologías que han convergido para otorgar movilidad física a los entornos computacionales. Estos equipos tienen servicios de posicionamiento, casi es obvio señalar que un dispositivo móvil implica movimiento. La movilidad le puede dar al desarrollador nuevas posibilidades respecto a otras plataformas como la localización del usuario, que deriva en funcionalidades de mucho valor para este.

Estrems Amestoy, Manuel (2011). Manifiesta que la nueva tecnología permite aplicar técnicas de mejora continua muy extendidas en la industria, según el ciclo de Shewhart, sin generación de papel ni pérdidas de tiempo significativas. Actualmente se procura establecer la capacidad de procesos o máquinas antes de empezar a producir productos, y después de establecida la capacidad, se realizan los gráficos de control para asegurar que los productos siguen saliendo bien durante la producción. De esta forma se eliminan las inspecciones finales y se lleva el control a allí donde

realmente se juega la calidad del producto que es en la máquina.

Diego Germán de la Riva Legajo (2007). nos indica que en particular la tecnología celular tiene como objetivo brindar “movilidad”, definida como la capacidad de acceder a la información y a los servicios en cualquier momento, de cualquier manera y en cualquier parte. El auge de los servicios que se proveen hoy en día en Internet, el avance tecnológico, la tendencia hacia dispositivos más pequeños y más rápidos, junto con la necesidad de acceso a la información en cualquier momento, son los factores determinantes del surgimiento de nuevas tecnologías de acceso a Internet desde cualquier tipo de dispositivos incluyendo a los teléfonos celulares.

2.2 MARCO TEÓRICO

2.2.1 EPSASA

Es una Entidad Prestadora Municipal de derecho privado, con autonomía técnica, administrativa y económica; normada por la ley N° 26338 Ley General de Saneamiento y su Reglamento aprobado con D.S N° 09-95-PRES y la Ley N° 26887 Ley General de Sociedades, y presupuestalmente está comprendida dentro de las Empresas de tratamiento empresarial, bajo el ámbito de la Dirección Nacional del Presupuesto Público del Ministerio de Economía y Finanzas. Tiene como objetivo principal brindar los servicios de agua potable y alcantarillado a las localidades de Huamanga y Huanta, su actividad principal es la captación, tratamiento,

almacenamiento y distribución del agua potable para uso doméstico, comercial e industrial en su ámbito de jurisdicción. De igual manera programa y ejecuta obras y proyectos referidos al sector como son ampliación y mejoramiento del sistema de saneamiento. La entidad que regula principalmente las tarifas cobradas a los usuarios es la Superintendencia Nacional de Servicios de Saneamiento (SUNASS).

Visión.- Ser una empresa organizada con sostenibilidad económica y social, logrando el bienestar y la confianza de nuestros clientes.

Misión.- Brindar servicios de agua potable y alcantarillado, contribuyendo a mejorar la calidad de vida de la población, con honestidad, preservando el medio ambiente y buen trato a nuestros usuarios, dentro de los estándares de calidad.

2.2.2 CATASTRO DE USUARIO

El catastro de usuarios comprende el conjunto de registros y procedimientos que permiten la exacta identificación y localización de los usuarios de los servicios de agua potable y desagüe¹. Posee toda la información necesaria de los usuarios activos, factibles, potenciales y clandestinos. Este registro contiene datos del usuario y del predio, las características técnicas de las conexiones de agua potable y desagüe, de las cajas de registro, de los marcos, tapas y micro medidores, así como datos complementarios de los servicios y del predio.

Existen diferentes tipos de usuarios que se tienen en cuenta en un catastro las cuales son:

¹ Catastro de usuarios de agua potable y desagüe; Manual para entidades prestadoras de servicios - SUM Canadá

- a. Activo.- Usuario que dispone de una conexión de agua potable y/o desagüe, independientemente de la situación de esta (activa, inactiva o cortada). Se encuentra registrado por la entidad prestadora, es decir, es un usuario legalmente conectado.
- b. Factible.- Usuario que no dispone de conexiones, pero tiene la posibilidad de instalar una conexión de agua potable y/o desagüe debido a que frente a su predio existe una red.
- c. Potencial.- Usuario que no dispone de ningún tipo de conexiones y que para tenerla requeriría de obras de ampliación debido a que frente a su predio no pasa una red.
- d. Clandestino.- Usuario que hace uso de los servicios de agua potable y/o desagüe sin autorización de la entidad prestadora.

A. COMPONENTES

El catastro tiene tres componentes entrelazados, los cuales no deben ser manejados en forma separada. Estos son: los planos, las informaciones y el software (sistema informático) que permite administrar la base de datos en forma eficaz y eficiente.

Los planos son la base cartográfica que posibilita la localización física de los predios sobre el cual se levantan las informaciones. La base cartográfica permite generar el código catastral. Los tipos y niveles de complejidad de los planos dependerán mucho de las necesidades reales de la entidad prestadora.

Las informaciones son los datos recogidos a través de una ficha y deben ser todos los

que requiera la entidad prestadora para realizar la gestión comercial. Los tipos y cantidad de datos dependerán de las necesidades reales. Las informaciones deben dar cuenta de los usuarios activos, factibles, potenciales y clandestinos a fin de registrarlos y tener un conocimiento real del mercado actual y potencial.

El tercer componente lo constituye el software o sistema informático, que permite procesar, almacenar y administrar la información de las fichas e incluso los planos.

B. METODOLOGÍA PARA REALIZAR LA ACTUALIZACIÓN DE CATASTRO DE USUARIO

La realización o actualización del catastro de usuarios es un proceso que determina una serie de actividades o pasos secuenciales para alcanzar los beneficios esperados. Los cuales son:

- a. Definición de objetivos y alcances.
- b. Organización de actividades recopilación y revisión de planos disponibles.
- c. Sectorización.
- d. Difusión e información a la población.
- e. Levantamiento de información catastral.
- f. Supervisión del trabajo de campo y primer control de calidad.
- g. Procesamiento de fichas catastrales.
- h. Proceso de inconsistencias.
- i. Pintado de códigos catastrales en cada predio.

2.2.3 MANIFIESTO ÁGIL

En 2001, 17 representantes de nuevas metodologías y críticos de los modelos de mejora basados en procesos se reunieron, convocados por Kent Beck, para discutir sobre el desarrollo de software. Fue un grito de ¡basta ya! a las prácticas tradicionales. Estos profesionales, con una dilatada experiencia como aval, llevaban ya alrededor de una década utilizando técnicas que les fueron posicionando como líderes de la industria del desarrollo software. Conocían perfectamente las desventajas del clásico modelo en cascada donde primero se analiza, luego se diseña, luego se implementa y por último se escriben algunos tests y se martiriza a un grupo de personas para que ejecuten manualmente el software, una y otra vez hasta la saciedad. Después de esta reunión expresaron y mostraron los cuatro valores de manifiesto ágil: Estamos descubriendo mejores maneras de desarrollar software tanto por nuestra propia experiencia como ayudando a terceros. A través de esta experiencia hemos aprendido a valorar²:

1. Individuos e interacciones sobre procesos y herramientas.
2. Software que funciona sobre documentación exhaustiva.
3. Colaboración con el cliente sobre negociación de contratos.
4. Responder ante el cambio sobre seguimiento de un plan.

Esto es, aunque los elementos a la derecha tienen valor, nosotros valoramos por encima de ellos los que están a la izquierda.

² Manifiesto por el Desarrollo Ágil de Software; <http://agilemanifesto.org/iso/es/manifesto.html>

Tras este manifiesto se encuentran 12 principios de vital importancia para entender su filosofía³:

1. Nuestra mayor prioridad es satisfacer al cliente mediante la entrega temprana y continua de software con valor.
2. Aceptamos que los requisitos cambien, incluso en etapas tardías del desarrollo. Los procesos Ágiles aprovechan el cambio para proporcionar ventaja competitiva al cliente.
3. Entregamos software funcional frecuentemente, entre dos semanas y dos meses, con preferencia al periodo de tiempo más corto posible.
4. Los responsables de negocio y los desarrolladores trabajamos juntos de forma cotidiana durante todo el proyecto.
5. Los proyectos se desarrollan en torno a individuos motivados. Hay que darles el entorno y el apoyo que necesitan, y confiarles la ejecución del trabajo.
6. El método más eficiente y efectivo de comunicar información al equipo de desarrollo y entre sus miembros es la conversación cara a cara.
7. El software funcionando es la medida principal de progreso.
8. Los procesos Ágiles promueven el desarrollo sostenible. Los promotores, desarrolladores y usuarios debemos ser capaces de mantener un ritmo constante de forma indefinida.
9. La atención continua a la excelencia técnica y al buen diseño mejora la Agilidad.

³ Doce Principios Del Software Ágil, <http://agilemanifesto.org/iso/es/principles.html>

10. La simplicidad, o el arte de maximizar la cantidad de trabajo no realizado, es esencial.
11. Las mejores arquitecturas, requisitos y diseños emergen de equipos auto-organizados.
12. A intervalos regulares el equipo reflexiona sobre cómo ser más efectivo para ajustar y perfeccionar su comportamiento.

2.2.4 METODOLOGIA SCRUM

A. LA HISTORIA DE SCRUM

Scrum es una metodología ágil para gestionar proyectos de software, que toma su nombre y principios de los estudios realizados sobre nuevas prácticas de producción por Hirotaka Takeuchi e Ikujiro Nonaka a mediados de los 80 escrito en su artículo titulado "The new new product development game". "Aunque surgió como práctica en el desarrollo de productos tecnológicos, resulta válido en los entornos que trabajan con requisitos inestables, y necesitan rapidez y flexibilidad; situaciones habituales en el desarrollo de algunos sistemas de software" (Keith, 2010, p. 20).

En 1993, Jeff Sutherland aplicó el modelo Scrum al desarrollo de software en Easel Corporation. En 1996 presentó, junto con Ken Schwaber, las prácticas que empleaba como proceso formal, para gestión del desarrollo de software en OOPSLA 96. En 2001 formaron parte de los firmantes del Manifiesto Ágil. Las prácticas diseñadas por Schwaber y Sutherland para gestionar el desarrollo de software están incluidas en la lista de modelos ágiles de Agile Alliance.

B. DEFINICIÓN DE SCRUM

“La metodología Scrum tiene pequeñas y simples prácticas que el equipo puede usar en el desarrollo; estas prácticas no son todas envolventes o perfectas para cada producto” (Keith, 2010, p. 40). Scrum se considera “una manera simple de manejar problemas complejos”, proporcionando un marco de trabajo para soportar la innovación y permitir que equipos auto-organizados entreguen resultados de alta calidad en tiempos cortos.

Según Keith (2010) es importante conservar los principios de Scrum:

- a. **El empirismo.-** Scrum usa la frase “revise y adapte” ciclo que posibilita al equipo y al cliente responder y a emerger conocimiento, cambiando condiciones en el tiempo y usando datos reales. Un ejemplo de esto puede verse en la práctica diaria de Scrum, lo cual faculta al equipo a reaccionar para los asuntos diarios.
- b. **El surgimiento.-** Como desarrollamos un juego, aprendemos más acerca de lo que le hace divertido y lo que es posible crearlo. El propósito de Scrum no es desarrollar de forma “up front“, Admiten que no podemos saber todo acerca de un juego desde el principio. La revisión del sprint y la planificación de un ciclo es diseñado para maximizar surgimiento.
- c. **Timeboxing.-** Scrum es iterativo, da valor de forma regular y faculta a los clientes y los desarrolladores a funcionar sincronizadamente. Los sprints son un ejemplo de una práctica del timeboxed.
- d. **Priorización.-** Algunas funcionalidades son más importantes para los clientes que otros. En vez de “implementar todo lo que se estipuló en el documento de

diseño”, solo trabajaremos en lo que da más valor el cliente. El product backlog es una expresión de este principio.

e. Auto organización.- Los equipos son multidisciplinarios y tienen la facultad de organizar a sus miembros, manejar sus procesos y crear un mejor producto dentro del tiempo establecido. Usando la frase “revise y adapte” mejorar continuamente a menudo se consigue realizando la retrospectiva.

Conservando estos principios, los equipos de Scrum pueden alterar sus prácticas y puede mejorar las prestaciones de Scrum.

B. PARTES DE SCRUM

En la figura N° 2.1 se muestran los artefactos y reuniones de Scrum en detalle como también algunas prácticas adicionales.

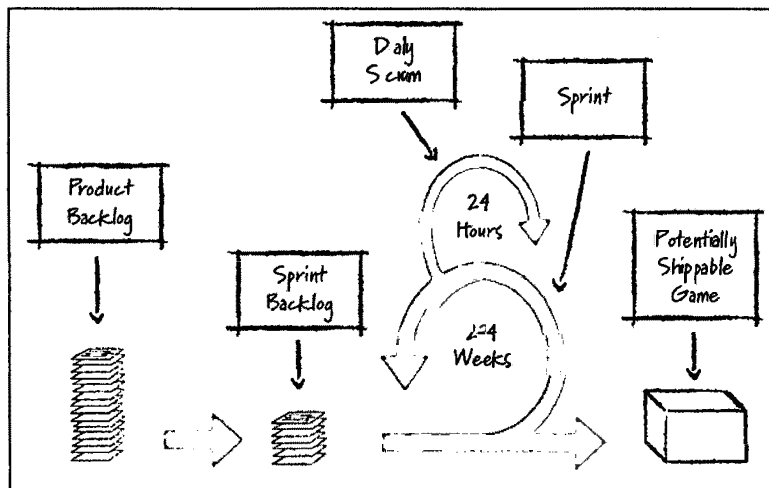


Figura N° 2.1: Artefactos y Reuniones de SCRUM (Keith, 2010, p. 39).

C. ARTEFACTOS

- a. **Product backlog.-** “El product backlog es un documento de alto nivel para todo el proyecto” (Cohn, 2004, p. 34). Contiene descripciones genéricas de todos los requerimientos, funcionalidades deseables y priorizadas según su retorno de inversión (ROI). Es el qué va a ser construido, es abierto y cualquiera puede modificarlo. Contiene estimaciones a groso modo, tanto del valor para el negocio, como del esfuerzo de desarrollo requerido.
- b. **Sprint backlog.-** El sprint backlog es un documento detallado donde se describe cómo el equipo va a implementar los requisitos durante el siguiente sprint. ”Las historias de usuario se dividen en tareas y si estas historias usuarios tienen una mayor duración superior a 16 horas se deberá dividir en otras menores” (Cohn, 2004, p. 35). Las tareas en el sprint backlog nunca son asignadas, son tomadas por los miembros del equipo del modo que les parezca oportuno.
- c. **Burn down.-** La burn down chart es una gráfica mostrada públicamente que mide la cantidad de requisitos en el Backlog del proyecto pendientes al comienzo de cada Sprint. ”Dibujando una línea que conecte los puntos de todos los Sprints completados, podremos ver el progreso del proyecto” (Cohn, 2004, p. 36). Lo normal es que esta línea sea descendente (en casos en que todo va bien en el sentido de que los requisitos están bien definidos desde el principio y no varían nunca) hasta llegar al eje horizontal, momento en el cual el proyecto se ha terminado (no hay más requisitos pendientes de ser completados en el Backlog). Si durante el proceso se añaden nuevos requisitos la recta tendrá pendiente ascendente en determinados segmentos, y si se

modifican algunos requisitos la pendiente variará o incluso valdrá cero en algunos tramos.

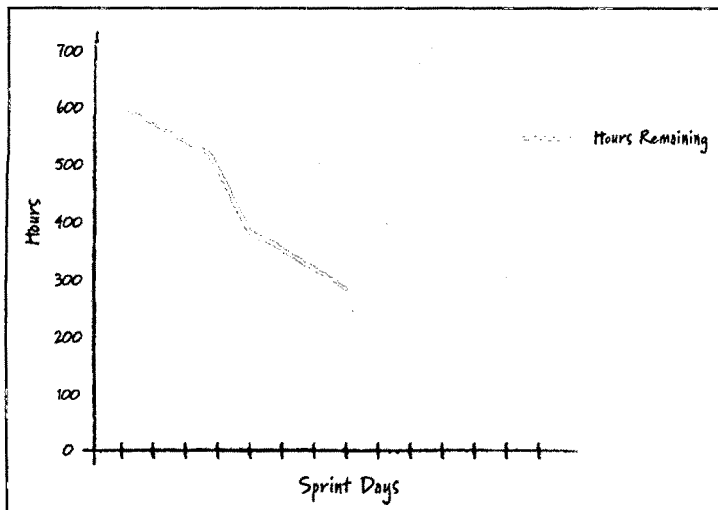


Figura N° 2.1: Burn down chart (Keith, 2010, p. 69).

D. REUNIONES EN SCRUM

a. **Reunión Diaria (Daily Scrum).**- Cada día de un sprint, se realiza la reunión sobre el estado de un proyecto. Esto se llama "daily Scrum". El scrum tiene unas guías específicas:

- i. La reunión comienza puntualmente a su hora.
- ii. Todos son bienvenidos, pero sólo los responsables pueden hablar.
- iii. La reunión tiene una duración fija de 15 minutos, de forma independiente del tamaño del equipo.
- iv. Todos los asistentes deben mantenerse de pie.
- v. La reunión debe ocurrir en la misma ubicación y a la misma hora todos los días.
- vi. Durante la reunión, cada miembro del equipo contesta a tres preguntas:

- ¿Qué has hecho desde ayer?
- ¿Qué es lo que estás planeando hacer hoy?
- ¿Has tenido algún problema que te haya impedido alcanzar tu objetivo? (Es el papel del ScrumMaster recordar estos impedimentos).

b. Reunión de Planificación del Sprint (Sprint Planning Meeting).- Al inicio del ciclo Sprint (cada 15 o 30 días), una “reunión de planificación del sprint“ se lleva a cabo para:

- i. Seleccionar qué trabajo se hará.
- ii. Preparar con el equipo completo el Sprint Backlog y que detalle el tiempo que tomará hacer el trabajo.
- iii. Identificar y comunicar cuánto del trabajo es probable que se realice durante el actual Sprint.
- iv. Ocho horas como límite.

c. Reunión de Revisión del Sprint (Sprint Review Meeting).-

- i. Revisar el trabajo que fue completado y no completado.
- ii. Presentar el trabajo completado a los interesados (alias “demo”).
- iii. El trabajo incompleto no puede ser demostrado.
- iv. Cuatro horas como límite.

d. Retrospectiva del Sprint (Sprint Retrospective).- Después de cada sprint, se lleva a cabo una retrospectiva del sprint, en la cual todos los miembros del equipo

dejan sus impresiones sobre el sprint recién superado. El propósito de la retrospectiva es realizar una mejora continua del proceso. Esta reunión tiene un tiempo fijo de cuatro horas.

e. Sprint Backlog.- El Sprint es el período en el cual se lleva a cabo el trabajo en sí. Es recomendado que la duración de los sprints sea constante y definida por el equipo con base en su propia experiencia. Se puede comenzar con una duración de sprint en particular (2 o 3 semanas) e ir ajustándolo con base en el ritmo del equipo, aunque sin relajarlo demasiado. Al final de cada sprint, el equipo deberá presentar los avances logrados, y deberán entregar un producto con características de utilizable por el cliente. Asimismo, se recomienda no cambiar los objetivos del sprint o sprint backlog a menos que la falta de estos cambios amenace al éxito del proyecto. La constancia permite la concentración y mejora la productividad del equipo de trabajo.

E. ROLES PRINCIPALES

a. Product Owner.- El product owner representa la voz del cliente. “Se asegura de que el equipo Scrum trabaja de forma adecuada desde la perspectiva del negocio” (Cohn, 2004, p. 36). El product owner escribe historias de usuario, las prioriza, y las coloca en el Product Backlog.

b. Scrum Master.- El Scrum es facilitado por un ScrumMaster, cuyo trabajo primario es eliminar los obstáculos que impiden que el equipo alcance el objetivo del sprint. El ScrumMaster no es el líder del equipo, sino que actúa como una protección entre el equipo y cualquier influencia que le distraiga. El ScrumMaster se asegura de

que el proceso Scrum se utiliza como es debido. El ScrumMaster es el que hace que las reglas se cumplan.

c. **Equipo de desarrollo.**- El equipo tiene la responsabilidad de entregar el producto. Un pequeño equipo de 3 a 9 personas con las habilidades transversales necesarias para realizar el trabajo (análisis, diseño, desarrollo, pruebas, documentación, etc).

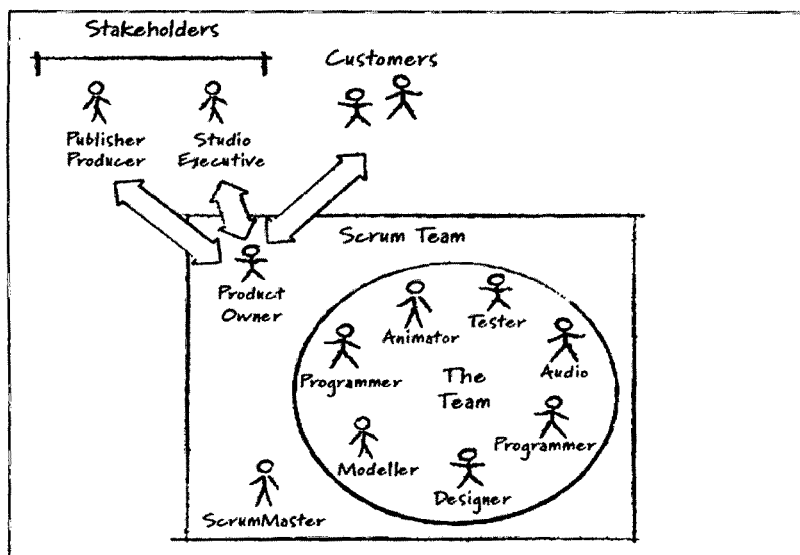


Figura Nº 2.2: Roles de Scrum (Keith, 2010, p. 73).

2.2.5 ESTIMACIÓN Y PLANIFICACIÓN

Cohn, M. (2004) nos muestra la forma de estimar las historia de usuario y cómo crear un release plan de alto nivel para la entrega de las historia de usuario de mayor prioridad. También vamos a buscar formas de medir y monitorear el progreso del proyecto, de modo que constantemente se puede ajustar el plan para incorporar los conocimientos que obtenemos de cada iteración.

A. STORY POINTS

Un enfoque que satisface cada uno de estos objetivos es el de estimar en puntos las historias de usuario. Una característica interesante de los story points es que cada equipo los define como mejor les parezca. Un equipo puede decidir para definir un punto de la historia como un día ideal de trabajo (es decir, un día sin interrupciones, ni reuniones, ni correo electrónico, sin llamadas telefónicas, y así sucesivamente). Dos ventajas que ofrece: en primer lugar, es más fácil la estimación en el tiempo transcurrido. La estimación en el tiempo transcurrido nos obliga a considerar todas las demás posibles impactos en nuestro tiempo. En segundo lugar, la estimación de story points en el momento ideal le da a nuestros cálculos una base un poco mejor que cuando se estima que son unidades totalmente nebulosas. Dado que uno de los principales propósitos de estimación es ser capaz de responder a preguntas acerca del esfuerzo general que se espera en un proyecto, que finalmente tendrá que convertir las estimaciones en el tiempo. Comenzando con el tiempo ideal se hace que la conversión sea un poco más simple para poder comenzar con una unidad totalmente nebuloso.

B. ESTIMACIÓN DE HISTORIAS DE USUARIO

El enfoque que hace uso Cohn, M. (2004) en la estimación es un método iterativo para el desarrollo de estimaciones. Así es como funciona: En primer lugar, reunir a los clientes y los desarrolladores que van a participar en la creación de las estimaciones. Traiga las tarjetas de historia de usuario y una pila de fichas adicionales

en blanco. Distribuya un puñado de tarjetas en blanco a cada participante. El cliente selecciona un artículo al azar de la colección y lo lee a los desarrolladores, los desarrolladores después de hacer tantas preguntas, ya que necesitan que el cliente responda todas, si no conoce la respuesta, se toma una suposición o le pide al equipo de diferir la estimación de esa historia. Cuando no hay más preguntas acerca de la historia de usuario, cada desarrollador escribe una estimación de la historia de usuario, aún no muestra la estimación de los otros. Si el equipo ha definido un punto de la historia de usuario como un día de trabajo ideal, los desarrolladores piensan en cuántos días ideales se llevará a completar la historia de usuario. Cuando todos hayan terminado de escribir una estimación, deben mostrar para que lo vean todos. Si las estimaciones son diferentes, alta y baja debe explicar porque creen que deben tener ese punto. El objetivo es aclarar y aprender de lo que se piensa de la historia de usuario. El cliente aclara las dudas que vayan surgiendo. En muchos casos, las estimaciones se aclaran en la segunda ronda, el objetivo es que los estimadores puedan converger en una sola estimación que se puede utilizar para la historia de usuario. La precisión disminuye a medida que aumenta el tamaño de la historia de usuario, esto es un problema con la estimación de puntos de la historia de usuario es que las diferencias entre algunos números pueden ser difíciles de justificar. Supongamos que los desarrolladores están teniendo en cuenta una historia de usuario y un desarrollador sugiere que la historia de usuario vale dos puntos. Otro replica que la historia de usuario es de tres puntos de la historia. Esta discusión tiene sentido: tres puntos de la historia de usuario representa Trabajo 50% más de dos puntos de la

historia. Sin embargo, ahora supongo que los desarrolladores están discutiendo sobre si la historia de usuario debe ser siete u ocho puntos de la historia de usuario, en la mayoría de los casos un punto de diferencia entre el número tan grande es demasiado pequeño para ser discutido con alguna relevancia. Argumentando acerca de si una historia de usuario vale siete u ocho puntos de la historia de usuario implica una mayor precisión en nuestro proceso de estimación de lo que tienen. Para evitar esta situación y simplificar las cosas, por el bien del equipo, puede llegar a un acuerdo para limitar las estimaciones de concreto pre-definido valores tales como:

$\frac{1}{2}$, 1, 2, 3, 5, 8, 13, 20, 40, 80

Esto es atractivo porque refleja la verdad que a medida que las estimaciones se hacen más grandes, sabemos menos acerca de ellos. Si el equipo tiene un EPIC a considerar que tendrán que decidir si se trata de un 40 o 80, pero no van a tener que pensar en si es un 79 o un 80.

C. PLANNING RELEASE

Cohn, M. (2004) nos expresa que lo ideal sería que los desarrolladores y el cliente pueden hablar de un rango de fechas, en lugar de una fecha específica, en un proceso iterativo de historia de usuario basada en el proceso hace que sea fácil fijar una fecha. Si un equipo puede empezar a planificar la liberación con un intervalo de fechas aceptables tendrán más flexibilidad en el calendario lanzamientos. Se recomienda que todas las release tenga una duración de 2 a 4 semanas y la reunión de la plannig release dure 4 horas, 2 horas indicando el “que“ y las otras 2 horas

especificando el “como”.

D. PRIORIZACIÓN DE HISTORIA DE USUARIO

Hay muchas dimensiones por las cuales podemos clasificar las historias de usuario. Entre los factores técnicos que podemos utilizar son los siguientes:

- a. El riesgo de que la historia no se puede completar según se desee.
- b. El impacto que ésta tendrá sobre otras historia de usuario.
- c. Además, los clientes y los usuarios tienen su propio conjunto de factores que podrían utilizar para ordenar las historias de usuario, entre ellas las siguientes:
 - i. La conveniencia de la historia de usuario a una amplia base de usuarios o clientes.
 - ii. La conveniencia de la historia de usuario de un pequeño número de usuarios o clientes importantes.
 - iii. La cohesión de la historia de usuario en relación con otras historias de usuario.

C. LA VELOCIDAD INICIAL: Hay tres maneras de obtener un valor inicial de la velocidad:

1. Utilice los valores históricos.
2. Ejecutar una iteración inicial y utilizar la velocidad de esa iteración.
3. Tome una conjetura.

Utilizando los valores históricos es la mejor opción, pero sólo es viable si tenemos un

equipo existente que está rodando fuera de un proyecto similar al nuevo proyecto y si no hay nadie que entre o se retire en el equipo. Por desgracia, es raro que el equipo sea exactamente la misma la que ponga a trabajar en dos años consecutivos proyectos similares.

Ejecución de una iteración inicial es una gran manera de conseguir una velocidad de arranque. Sin embargo, hay muchas veces cuando esto no es viable. Adivinar a una velocidad si un punto de la historia de usuario es un día ideal de trabajo, se puede estimar la velocidad inicial mediante la estimación de la cantidad de días reales que nos llevará a completar un todo el día ideal de trabajo.

D. CREANDO EL RELEASE PLAN

Por lo tanto, si el proyecto cuenta con 100 puntos de la historia de usuario y la estimación de la velocidad es 20 puntos de la historia de usuario por iteración, entonces podemos esperar que tome cinco iteraciones. El paso final en la planificación de liberación es asignar historia de usuario en cada una de las iteraciones. Con colaboración del cliente y los desarrolladores, se seleccionan veinte puntos de la historia de usuario de más alta prioridad y los situará en la primera iteración. La próxima historia de usuario de veinte puntos que se va a desarrollar en la segunda iteración y así sucesivamente hasta que todas las historia de usuario ha sido asignada.

E. HISTORIA DE USUARIO

Una historia de usuario es una narración breve de lo que el product owner desea que se desarrolle, una historia de usuario comprende tres aspectos:

- Una descripción que la identifica.
- Criterio de aceptación utilizada para saber cuándo una historia de usuario está realmente terminada.
- Conversaciones con el cliente para ahondar detalles.

Convención de redacción de las historia de usuario:

As a [type of user]

I want to [perform some task]

so that I can [achieve some goal.]

Para crear buenas historia de usuario centrase en los seis atributos INVEST:

- a. Independiente.- Tanto como sea posible, se debe tener cuidado para evitar la introducción de las dependencias entre las historia de usuario. Las dependencias nos causan problemas de priorización y planificación.
- b. Negociable.- Tarjetas de la historia de usuario son descripciones breves de funcionalidad, cuyos detalles serán negociados en una conversación entre el cliente y el equipo de desarrollo. Porque las tarjetas son un recordatorio de la historia de usuario para tener una conversación en lugar de totalmente los requisitos detallados en sí, no es necesario incluir todos los detalles pertinentes.
- c. Valorable.- representante del cliente o el cliente en sí. No siempre lo que valora el

cliente es valorable por los usuarios. Lo que debemos de evitar es utilizar jerga técnica que solo comprenda el desarrollador.

d. Estimable.- las historias de usuario deben de ser estimables por los desarrolladores. Hay tres razones más comunes por las que una historia no puede ser estimable:

1. Los desarrolladores no tienen conocimiento del dominio.
2. Los desarrolladores carecen de conocimientos técnicos.
3. La historia es demasiado grande.

e. Pequeño.- las historias de usuario deberían ser lo suficientemente chicas para poder estimarlas y puedan realizarse en lo que dura una iteración, si las historia de usuario son muy grandes o chicas no se pueden utilizar en la planificación.

f. Testeable.- Las historia de usuario deberán ser escritos con el fin de ser testeable. Pasar con éxito sus pruebas demuestra que una historia de usuario se ha desarrollado con éxito. Si el user story no se puede probar, ¿cómo pueden los desarrolladores saber cuando han terminado de codificar?

F. PRUEBAS DE ACEPTACIÓN

Una de las razones para escribir pruebas de aceptación es la de expresar muchos los detalles que resultan de las conversaciones entre clientes y los desarrolladores. En lugar de escribir largas listas de requisitos como: "El sistema deberá ...", las pruebas se utilizan para rellenar los detalles de un historia de usuario .

La prueba se ve mejor como un proceso de dos pasos: En primer lugar, las notas

sobre las pruebas se anotan en el posterior de las tarjetas. En segundo lugar, sirve para verificar si se ha terminado de programar. Una buena manera de hacer esto es buscar en cada historia y pedir preguntas similares a lo siguiente:

¿Qué otra cosa que los programadores necesitan saber acerca de esta historia de usuario?

¿Qué estoy asumiendo acerca de cómo esta historia de usuario se llevará a cabo?

¿Hay circunstancias en las que esta historia de usuario puede comportarse de manera diferente?

¿Qué puede salir mal durante la historia de usuario?

El cliente debe seguir escribiendo las pruebas, siempre y cuando agregue valor y la clarificación de la historia de usuario. El cliente debe enfocar sus esfuerzos en escribir pruebas que aclaren la intención de la historia de usuario y que los desarrolladores puedan entender. Sin embargo, otros tipos de pruebas para ser consideradas pueden ser las siguientes:

- User interface testing, que asegura que todos los componentes de la interfaz de usuario se comportan como se esperaba.
- Usability testing, que se hace para asegurar una aplicación que puede ser fácilmente utilizado.
- Performance testing, que se realiza para medir qué tan bien la aplicación se realice en virtud de diversas cargas de trabajo.

Un buen patrón para las pruebas debe de tener esta convención:

GIVEN: startingcriterial AND startingcriteria2 AND startingcriteria3 ...

WHEN: some event

THEN: outcome1 AND outcome2 AND outcome3 ...

G. MODELADO DE ROL

Es evidente que no podemos escribir las historias de usuario de una sola perspectiva y tener esas historias de usuario que refleje las experiencias, antecedentes y objetivos de cada usuario. Vamos a utilizar los siguientes pasos para identificar y seleccionar a los usuarios:

- a. Identificación, utilizar fichas donde escribir el nombre del rol, y pégalo en la mesa o pared, solo tener presente una regla no discutir sobre las ideas de los demás.
- b. Organización, se utiliza la técnica de solapado; si los roles se solapan mucho entonces solapar mucho las fichas y si los roles se solapan poco entonces solapar poco las fichas.
- c. Consolidación, después de que los roles se han agrupado, tratar de consolidar y condensar. Comience con las tarjetas que son totalmente coincidentes.
- d. Refinamiento, una vez que hemos consolidado y tienen un conocimiento básico de cómo los roles se relacionan entre sí, es posible modelar dichas funciones por definir los atributos de cada rol. Un atributo de rol es un hecho o un poco de información útil sobre los usuarios que cumplan con el rol.

modelo bidimensional, ayuda a pensar al producto desde los procesos de negocio y las necesidades de los usuarios. En este modelo es más simple asegurar que el backlog está completo y se visualizan más claramente las funcionalidades alternativas y la relación entre las historias de usuario de alto nivel y los detalles. Los story maps son útiles a la hora de definir el backlog inicial, pero también resultan una herramienta interesante para la creación de un plan de releases que aproveche al máximo el aspecto iterativo y evolutivo del desarrollo ágil.

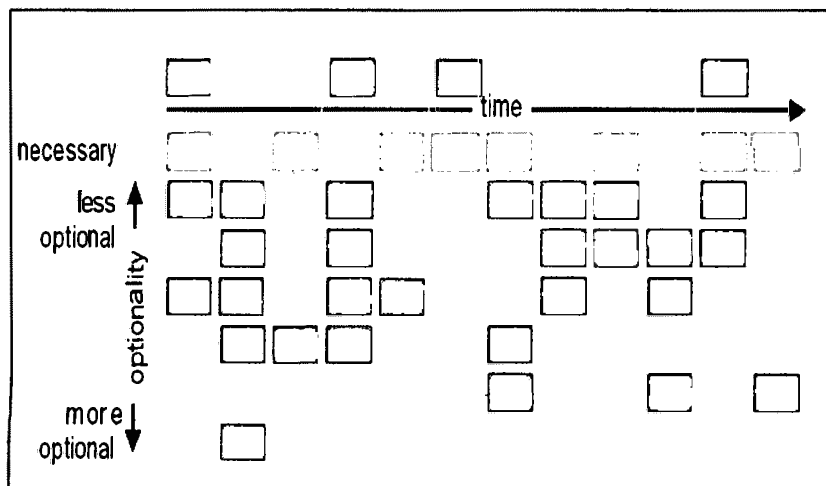


Figura N° 2.4 Jeff Patton Building Better Products Using User Story Mapping

2.2.6 DESARROLLO ÁGIL DE SOFTWARE

Jim Shore and Shane Warden (2007). Manifiestan que el desarrollo de software requiere la cooperación de todos en el equipo. Los programadores a menudo se llaman "Desarrolladores", pero en realidad todos en el equipo son parte del esfuerzo de desarrollo. ¿Cómo compartimos el trabajo? los clientes identifican los requisitos siguientes, mientras que los programadores trabajan en los actuales. Los testers

ayudan al equipo a encontrar la manera de dejar de introducir errores. Los programadores a distribuir el costo de la technical infrastructure a lo largo de toda la vida del proyecto. Por encima de todo, todos ayudan a mantener todo limpio. La mejor forma que conocen para reducir el costo de la escritura de software es mejorar la calidad interna del código y el diseño. Siempre reduce el coste de desarrollo en el corto plazo, así como a largo plazo.

Aquí hay cinco prácticas que mantener el código limpio y permitir que todo el equipo para contribuir al desarrollo:

- a. Incremental Requirements permite que el equipo empiecen a trabajar mientras que los clientes a especificar los detalles de los requisitos.
- b. Customer Tests para comunicar reglas difíciles de dominio.
- c. Test-Driven Development permite a los programadores tener confianza en que su código hace lo que piensa.
- d. Refactoring permite a los programadores para mejorar la calidad del código sin cambiar su comportamiento.
- e. Incremental Design and Architecture permite a los programadores a trabajar en las características en paralelo con la técnica infraestructura.

A. DESARROLLO DIRIGIDO POR PRUEBA (TDD)

El Desarrollo Dirigido por Tests (Test Driven Development), es una técnica de diseño e implementación de software incluida dentro de la metodología XP. TDD es una técnica para diseñar software que se centra en tres pilares fundamentales:

- La implementación de las funciones justas que el cliente necesita y no más.
- La minimización del número de defectos que llegan al software en fase de producción.
- La producción de software modular, altamente reutilizable y preparado para el cambio.

a. EL ALGORITMO TDD

La esencia de TDD es sencilla pero ponerla en práctica correctamente es cuestión de entrenamiento, como tantas otras cosas. El algoritmo TDD sólo tiene tres pasos:

- Escribir la especificación del requisito.
- Implementar el código según dicho ejemplo.
- Refactorizar para eliminar duplicidad y hacer mejoras.

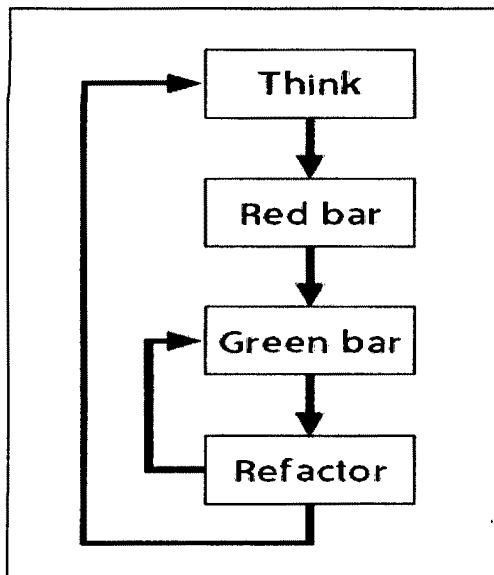


Figura Nº 2.4: Ciclo TTD (Jim Shore and Shane Warden, p. 287).

2.2.7 TECNOLOGÍA

A. AQUITECTURA CLIENTE SERVIDOR

Uno de los modelos consiste en 2 capas; que es una arquitectura constituida por Front-End y Back-End.

- a. Front-End: consiste en la capa donde el usuario interactúa con su terminal.
- b. Back-End: es el servidor de bases de datos como Sqlite3 o Postgres.

B.RUBY

Ruby es un lenguaje de programación interpretado, reflexivo y orientado a objetos, creado por el programador japonés Yukihiro "Matz" Matsumoto, quien comenzó a trabajar en Ruby en 1993, y lo presentó públicamente en 1995. Combina una sintaxis inspirada en Python y Perl con características de programación orientada a objetos similares a Smalltalk. Comparte también funcionalidad con otros lenguajes de programación como Lisp, Lua, Dylan y CLU. Ruby es un lenguaje de programación interpretado en una sola pasada y su implementación oficial es distribuida bajo una licencia de software libre.

C. RUBY ON RAILS

Ruby on Rails, también conocido como RoR o Rails es un framework de aplicaciones web de código abierto escrito en el lenguaje de programación Ruby, siguiendo el paradigma de la arquitectura Modelo Vista Controlador (MVC). Trata de combinar la simplicidad con la posibilidad de desarrollar aplicaciones del mundo real

escribiendo menos código que con otros frameworks y con un mínimo de configuración. El lenguaje de programación Ruby permite la metaprogramación, de la cual Rails hace uso, lo que resulta en una sintaxis que muchos de sus usuarios encuentran muy legible. Rails se distribuye a través de RubyGems, que es el formato oficial de paquete y canal de distribución de bibliotecas y aplicaciones Ruby.

D.RHODES

Rhodes es un framework open source de Rhomobile. Se desarrolla aplicaciones nativas para casi todos los Smartphones. Las aplicaciones construidas a través de Rhodes son puras aplicaciones nativas y utilizar las capacidades del dispositivo, tales como GPS, contactos PIM y el calendario, cámara, native mapping, push, barcode, signature capture, and Bluetooth.

Rhodes acelera el desarrollo de aplicaciones móviles sin comprometer en su portabilidad. Este framework es similar al framework Rails. Se basa en Modelo Vista Controlador y ha incorporado Administrador de objetos relacionales (ORM) denominado Rhom que es similar al active Record en Rails. La mayoría de personalización de la interfaz de usuario se puede hacer en plantillas HTML (ERB, los archivos eruby). Un conjunto único de fuente escrita con Rhodes puede ser compilado para ejecutarse a través de todos los smartphones. Esto significa que vamos a tener el mismo código fuente para todos los dispositivos.

E.RHOSYNC

RhoSync es un servidor independiente de sincronización móvil que mantiene los

datos de aplicaciones empresariales hasta la fecha y disponible en los teléfonos inteligentes de los usuarios. Aplicaciones locales de las empresas necesitan datos sincronizados para ser usado la mayor parte del tiempo. La información se almacena localmente en el dispositivo del usuario y está disponible para ellos, incluso en el modo fuera de línea. Es muy fácil escribir un adaptador de fuente como RhoSync genera la mayor parte del código al crear el adaptador de fuente. El adaptador de fuente también se puede utilizar para Create, Read, Update, and Delete (CRUD) en un modelo.

Redis usos Rhosync, lo cual es una tienda de NoSQL valor clave para el almacenamiento en caché de datos. Esto hace que Rhosync más escalable. RhoSync lleva a cabo su impulso basado en sincronización con los SDK de teléfonos inteligentes nativos de inserción. Utiliza nuevas y avanzadas de BlackBerry Enterprise Server and iPhone 3.0 SDKs para Push. Utiliza BES y Push para la sincronización de iPhone, lo que permite actualizaciones en tiempo real de información de la aplicación back-end.

E.ANDROID

Android es un sistema operativo móvil basado en Linux, que junto con aplicaciones middleware, está enfocado para ser utilizado en dispositivos móviles como teléfonos inteligentes, tabletas, Google TV y otros dispositivos. Es desarrollado por la Open Handset Alliance, la cual es liderada por Google. Este sistema por lo general maneja aplicaciones como Market (Marketing). Fue desarrollado inicialmente

por Android Inc., una firma comprada por Google en 2005. Es el principal producto de la Open Handset Alliance, un conglomerado de fabricantes y desarrolladores de hardware, software y operadores de servicio.

La estructura del sistema operativo Android se compone de aplicaciones que se ejecutan en un framework Java de aplicaciones orientadas a objetos sobre el núcleo de las bibliotecas de Java en una máquina virtual Dalvik con compilación en tiempo de ejecución.

F.SINATRA

Sinatra es un framework para aplicaciones web de software libre y código abierto, y Lenguaje específico del dominio escrita en Ruby. Es una alternativa a otros frameworks para aplicaciones webs como Ruby on Rails, Merb, Nitro, Camping, and Rango. Sinatra depende de Rack interfaz de servidor web.

Diseñado y desarrollado por Blake Mizerany, Sinatra es pequeño y flexible. Sinatra no sigue el típico patrón modelo vista controlador que se ve en otros frameworks, como Ruby on Rails. En su lugar, Sinatra se enfoca en la rápida creación de aplicaciones web en Ruby con el mínimo esfuerzo.

G.SQLITE

SQLite es un sistema de gestión de bases de datos relacional compatible con ACID, contenida en una relativamente pequeña, biblioteca escrita en C. SQLite es un proyecto de dominio público creado por D. Richard Hipp. A diferencia de los

sistemas de gestión de bases de datos cliente-servidor, el motor de SQLite no es un proceso independiente con el que el programa principal se comunica. En lugar de eso, la biblioteca SQLite se enlaza con el programa pasando a ser parte integral del mismo. El programa utiliza la funcionalidad de SQLite a través de llamadas simples a subrutinas y funciones. Esto reduce la latencia en el acceso a la base de datos, debido a que las llamadas a funciones son más eficientes que la comunicación entre procesos.

H.RSPECT

RSpec fue creado por Steven Baker, en 2005. Es un framework de TDD para el lenguaje de programación ruby que alienta a centrarse en el comportamiento. RSpec desde que nació ha evolucionado sus detalles sintácticos, la premisa básica sigue siendo la misma. Usamos RSpec para escribir ejemplos ejecutables del comportamiento, que se espera de un pequeño fragmento de código en un controlado contexto.

CAPITULO III

METODOLOGIA DE LA INVESTIGACION

3.1. TIPO DE INVESTIGACION

El tipo de investigación es aplicada. Se empleo la metodología Scrum para obtener los artefactos necesarios que se requieren para llegar a implementar y probar la aplicación web para teléfonos inteligentes.

3.2. DISEÑO DE LA INVESTIGACION

El diseño de la investigación esta en función de los objetivos que se desean, existiendo varios tipos de diseño según su estudio. así, existen: estudios exploratorios para averiguar si existe o no un fenómeno se realiza como primer paso de la investigación, estudios descriptivos que examinan un fenómeno para describirlo mas completamente o para diferenciarlo de otro, estudios predictivos para identificar las relaciones que permiten especular acerca de una cosa sabiendo algo de otra, estudios explicativos que procuran encontrar la relación causa-efecto entre dos o mas fenómenos y estudios de acción que orientan a solucionar un problema de tipo social, político, de mercado, etc. (José Cegarra, 2004). Entonces la investigación se de tipo descriptivo.

3.3. POBLACION Y MUESTRA

POBLACION

Toda la información necesaria de los usuarios de agua potable para la actualización del catastro de usuario de agua potable de la entidad prestadora de servicio de saneamiento EPSASA, el año 2012.

MUESTRA

La información necesaria de los usuarios de agua potable para la actualización del catastro de usuario de agua potable de la entidad prestadora de servicio de saneamiento EPSASA, el año 2012.

3.4. VARIABLES E INDICADORES

VARIABLE INDEPENDIENTE

CATASTRO

El catastro de usuarios comprende el conjunto de registros y procedimientos que permiten la exacta identificación y localización de los usuarios de agua potable y desagüe.



INDICADORES DE LA VARIABLE INDEPENDIENTE

A. LOCALIZACION

Es un neologismo que refiere al posicionamiento con el que se define la localización de un objeto espacial (representado mediante punto, vector, área, volumen) en un sistema de coordenadas y datum determinado. Este proceso es utilizado frecuentemente en los Sistemas de Información Geográfica.

B. CARACTERÍSTICAS DE CONEXION

Las características técnicas de las conexiones domiciliarias de agua potable son datos que se refieren a fecha de instalación, existencia de caja del medidor o conexión directa, diámetro de conexiones, material, estado, situación, entre otros.

C. PREDIO

Es toda vivienda, terreno sin construir, unidad habitacional, local, oficina o unidad catastral según la Base de Datos catastral.

VARIABLE DEPENDIENTE

USUARIO DE AGUA POTABLE

Son aquellas persona que requieran el consumo de agua potable, entre los tipos de usuario tenemos activo, factible, clandestino y potencial.

INDICADORES DE LA VARIABLE DEPENDIENTE

A. ACTIVO

Usuario que dispone de una conexión de agua potable y/o desagüe, independientemente de la situación de esta (activa, inactiva o cortada). Se encuentra registrado por la entidad prestadora, es decir, es un usuario legalmente conectado.

B. FACTIBLE

Usuario que no dispone de conexiones, pero tiene la posibilidad de instalar una conexión de agua potable y/o desagüe debido a que frente a su predio existe una red.

C. POTENCIAL

Usuario que no dispone de ningún tipo de conexiones y que para tenerla requeriría de

obras de ampliación debido a que frente a su predio no pasa una red.

D. CLANDESTINO

Usuario que hace uso de los servicios de agua potable y/o desagüe sin autorización de la entidad prestadora.

3.5. TECNICAS E INSTRUMENTOS

3.5.1. INSTRUMENTO PARA RECOLECTAR INFORMACION

Se empleó la técnica de entrevista, que tiene como objeto el estudio descriptivo de la información necesaria obtenida de los usuarios, se realizó la entrevista al personal de catastro de usuario de la entidad prestadora de servicios de saneamiento EPSASA. Se hizo una revisión documentaria para la recolección de datos e información necesaria para comprender en profundidad el funcionamiento del catastro de usuario. Los instrumentos necesarios que se utilizan para la recolección de información se muestra en el anexo G.

3.5.2. HERRAMIENTAS PARA EL TRATAMIENTO DE DATOS E INFORMACIÓN

Las herramientas que se han seleccionado para el desarrollo de la investigación, están en función de la metodología Scrum, y también en utilizar el framework Rhodes que nos permite implementar aplicaciones web para móviles que se ejecuta localmente en su dispositivo móvil y esta implementado con el estándar de arquitectura MVC, para satisfacer los requisitos del sistema, tal como se muestra en la siguiente tabla:

Nombre	Fabricante	Licencia	Servicio
Ubuntu 11.10	Equipos comerciales y la comunidad de colaboradores	Lincencia Libre	Sistema operativo
Gedit	TheGNOME Project	GNU General Public License	Editor de texto de propósito general, se utilizara como entorno de desarrollo
Ruby on Rails	O'Reilly Media	Lincencia Libre	Framework para aplicación web
Rhodes	Rhomobile, Inc.	open source under the MIT License	Framework para aplicaciones web para móviles.
Rhom	Rhomobile, Inc.	open source under the MIT License	mini database object mapper (ORM) para Rhodes
RhoSync	Rhomobile, Inc.	open source under the MIT License	servidor de sincronización de datos
Cucumber	Dan North, Chris Matts, Liz Keogh, David Chelimsky	open source	Framework para automatizar los criterios de aceptación
Rspec	Steven Baker	open source	Framework para automatizar las pruebas unitarias
Sinatra	Blake Mizerany	open source	Servidor de aplicaciones
Sqlite	D. Richard Hipp	open source	Sistema de gestión de base de datos relacional
Android	Open Handset Alliance	open source	sistema operativo móvil basado en Linux

Tabla Nº 3.1. Materiales tecnológicas para el tratamiento de datos.

3.5.3. TECNICAS PARA APLICAR SCRUM

Para obtener los artefactos durante el desarrollo de la investigación se emplea la metodología Scrum, se utiliza las tablas que están divididas por cada reunión celebrada, artefactos que se obtienen, técnicas que se aplican y los responsables de cada actividad.

1. PLANING MEETING (Timebox = 4 horas)

- Modelado de roles.
- Identificación de requerimientos y escritura de historia de usuario.

a. Modelado de roles

Actividades	Entregables	Técnica	Responsable
Identificación de roles	Lista de roles	1. Identificación 2. Organización 3. Consolidación 4. Refinamiento	Equipo de desarrollo y Scrum Master

Tabla Nº 3.2. Modelado de roles

b. Identificación de requerimientos y escritura de historia de usuario

Actividades	Entregables	Técnica	Responsable
Identificación de requerimiento	Product backlog inicial	Entrevistas Observaciones	Product owner, equipo de desarrollo y scrum master
Priorización de las historias de usuarios	Product backlog priorizado	Hi-Low Showdown	Product owner, equipo de desarrollo y scrum master

Convertir requerimientos a historia de usuario	Historia de usuario	Workshop de escritura de historia de usuario y propiedad INVEST.	Product owner, equipo de desarrollo y scrum master
Escribir detalle de las historias de usuario	Detalle de las historias de usuario	Convención de redacción: Como<rol> Necesito <actividad> Con el <objetivo>	Product owner, equipo de desarrollo y scrum master
Escribir criterios de aceptación de las historias de usuario	Criterios de aceptación de la historia de usuario	Conveccion de criterios de aceptcion: Given <>	Product owner, equipo de desarrollo y scrum master
Estimación de historia de usuario	Peso de la historia de usuario	story points	Equipo de desarrollo

Tabla Nº 3.3. Identificación de requerimientos y escritura de historia de usuario

2. RELEASE PLAN

Actividades	Entregables	Técnica	Responsable
Identificación de la velocidad del equipo de desarrollo	Velocidad del equipo	valores históricos	Equipo de desarrollo
Armado de la reléase plan	Fechas de entregas	Story mapping	Scrum Master

Tabla Nº 3.4. Relase plan

3. SPRINT (Timebox = 2 a 4 semanas)

- Dayli meeting
- Desarrollo ágil
- Review meeting
- Retrospectiva

a. **Dayli meeting** (Timebox = 15 minutos)

Actividades	Entregables	Técnica	Responsable
Comunicación y Sincronización	Lista de Impedimentos	¿Qué se hizo ayer? ¿Qué se hará hoy? ¿Cuáles son los impedimentos que tengo para terminar mi trabajo?	Scrum Master, equipo de desarrollo

Tabla Nº 3.5. Dayli Meeting

b. **Desarrollo**

Actividades	Entregables	Técnica	Responsable
Convertir historias de usuario en tareas	Tareas	Desagregación	Equipo de desarrollo
Escribir las pruebas unitarias	Automatización de las Pruebas unitarias	Rspec	Equipo de desarrollo
Programación	Código	Solo necesario para cumplir con la pruebas unitarias	Equipo de desarrollo
Refactoring	Diseño simple	Eliminación de código duplicado	Equipo de desarrollo
Actualizar el taskboard y el burn-down chart	Taskboard Burn Down Chart	Kanban	Scrum Master

Tabla Nº 3.6. Desarrollo

c. Review meeting

Actividades	Entregables	Técnica	Responsable
Verificar el cumplimiento del product backlog comprometido	Demo	Validación de los criterios de aceptación	Stakeholder, product owner, scrum master, equipo de desarrollo
Identificación de nuevas funcionalidades	Product backlog actualizado	identificación	Stakeholder, product owner, scrum master, equipo de desarrollo

Tabla Nº 3.7. Review Meeting

d. Retrospectiva (Timebox = 3 a 4 horas)

Actividades	Entregables	Técnica	Responsable
Mejorar los procesos de Scrum	¿Qué funciona bien? ¿Qué funciona mal?	Asociación muda	Scrum Master, equipo de desarrollo
Solucionar los impedimentos	Acciones a aplicar para mejora		Scrum Master, equipo de desarrollo

Tabla Nº 3.8. Retrospectiva.

CAPITULO IV

RESULTADO DE LA INVESTIGACION

4.1. PROCESOS DEFINIDOS POR METODOLOGUIA SCRUM

Tal como en el capítulo III de la sección 3.5.3. TECNICAS PARA APLICAR SCRUM, muestra la planinng meeting (modelado de roles, identificación de requerimientos y escritura de historia de usuario), relase plan y el sprint (dayli meeting, desarrollo ágil, review meeting, retrospectiva) que nos a permitió generar artefactos para el desarrollo de la aplicación web para teléfonos inteligentes.

4.1.1. PLANINNG MEETING (Timebox = 4 horas)

En la planificación se determino que la culminación del proyecto se realizara en 3 Sprint, para ello se realizo el modelado de roles, la identificación de los requisitos y la escritura de historias de usuario.

a. MODELADO DE ROLES

Utilizando la técnica de identificación obtuvimos los siguientes roles, ver Anexo A.

Usuario	Abogado
Prediador	director
Encuestador	Administrador
Digitador	Cliente

Supervisor	cliente clandestinos
Guía	cliente potencial
jefe de proyecto	cliente factible
Coordinador	cliente activo
jefe de comisión	Topógrafo
Asesor	auxiliar topógrafo
Entrenador	Auxiliar
Técnico de autocad	Dibujante

Tabla N° 4.1 resultado de la técnica de identificación.

Utilizando la técnica de organización obtuvimos, ver anexo A.

Abogado	Cliente	director
	Cliente clandestinos	Administrador
Supervisor	Cliente potencial	
Guía	Cliente factible	Topógrafo
Jefe de proyecto	Cliente activo	Auxiliar topógrafo
Coordinador		Auxiliar
Jefe de comisión	Usuario	Dibujante
Asesor	Predador	técnico autocat
Entrenador	Encuestador	
	Digitador	

Tabla N° 4.2 resultado de la técnica de organización.

Utilizando las técnicas de Consolidación y Refinamiento, ver Anexo A

Nombre	Descripción	Nivel de responsabilidad	Frecuencia de uso
Administrador		Elevado	Semana
Predador	Encargado de actualizar las fichas catastrales.	Bajo	Hora
Supervisor		Medio	Día
Cliente		Bajo	Nunca

Tabla N° 4.3 resultado de la técnica de consolidación y refinamiento.

b. IDENTIFICACION DE REQUERIMIENTOS Y ESCRITURA DE HISTORIA DE USUARIO

b.1. IDENTIFICACION DE REQUERIMIENTO

La identificación del requerimiento se obtuvo de las entrevistas, conversaciones y observaciones que se hizo en la reunión de planificación, ver anexo B.

Product backlog inicial
registrar estado de conexión
iniciar sesión
registrar conexión agua
definir la arquitectura
registrar rol
ubicar predio
localizar predio
registrar medidor
registrar unidad de uso
registrar predio
identificar cliente
registrar usuario

Tabla N° 4.2 resultado de la técnica de organización.

b.2. PRIORIZACIÓN DE LAS HISTORIAS DE USUARIOS

En la priorización de los requerimientos se utilizó la técnica Hi-Low Story Showdown en el visual story mapping. También se identificó las funcionalidades, epics, historias de usuario, ver anexo B.

Prioridad	Procesos	Product Backlog Ítem
1	no-funcional	definir la arquitectura

2	localización de predio	ubicar predio
3		localizar predio
4	características de conexión	identificar cliente
5		registrar predio
6		registrar estado de conexión
7		registrar medidor
8		registrar conexión agua
9	usos del predio	registrar unidad de uso
10	Administración	iniciar sesión
11		registrar usuario
12		registrar rol

Tabla N° 4.2. Priorización de las historias de usuarios.

b.3. CONVERTIR REQUERIMIENTOS A HISTORIA DE USUARIO

La conversión de requerimientos a historia de usuario se obtuvo en el taller de workshop de escritura de historias de usuario, ver el anexo H. Se lista a continuación:

Historia de usuario 01: definir la arquitectura.

Historia de usuario 02: obtener localización.

Historia de usuario 03: localizar predio.

Historia de usuario 04: identificar cliente.

Historia de usuario 05: registrar predio.

Historia de usuario 06: registrar estado de conexión.

Historia de usuario 07: registrar medidor.

Historia de usuario 08: registrar conexión agua.

Historia de usuario 09: registrar unidad de uso.

Historia de usuario 10: iniciar sesión.

Historia de usuario 11: registrar usuario.

Historia de usuario 12: registrar rol.

b.4. ESCRIBIR DETALLE DE LAS HISTORIAS DE USUARIO

En la escritura de la descripción de las historias de usuario se empleó el formato que se propone en el anexo G.

HISTORIA DE USUARIO	
Título: Definir Arquitectura	Prioridad 01
	Estimación
Descripción: Definir la configuración de RHODES con las capas Modelo-Vista-controlador definir la configuración de RHOM la capa de persistencia con ORM embebido definir la configuración de RHOSYNC para sincronizar la base de datos cliente-server.	

Figura Nº 4.1. Historia de usuario "Definir Arquitectura".

HISTORIA DE USUARIO	
Título: Obtener localización	Prioridad 02
	Estimación
Descripción: como prediador, puedo utilizar el GPS con la finalidad de obtener la latitud y longitud y convertir en coordenadas UTM.	

Figura Nº 4.2. Historia de usuario "Obtener localización".

HISTORIA DE USUARIO	
Título: Localizar Predio	Prioridad
	03
Estimación	
Descripción: Como prediador, puedo localizar el predio con la finalidad de obtener datos del predio.	

Figura Nº 4.3. Historia de usuario "Localizar Predio"

HISTORIA DE USUARIO	
Título: Identificar Cliente	Prioridad
	04
Estimación	
Descripción: como prediador, puedo verificar los datos del dueño del predio.	

Figura Nº 4.4. Historia de usuario "Identificar Cliente"

HISTORIA DE USUARIO	
Título: Registrar Predio	Prioridad
	05
Estimación	
Descripción: Como prediador, puedo registrar los detalles del predio.	

Figura Nº 4.5. Historia de usuario "Registrar Predio"

HISTORIA DE USUARIO	
Título: Registrar Estado Conexión	Prioridad 06
	Estimación
Descripción: Como prediador, puedo registrar el estado de conexión.	

Figura Nº 4.6. Historia de usuario “Registrar Estado Conexión”

HISTORIA DE USUARIO	
Título: Registrar Medidor	Prioridad 07
	Estimación
Descripción: Como prediador, puedo registrar los datos del medidor	

Figura Nº 4.7. Historia de usuario “Registrar Medidor”

HISTORIA DE USUARIO	
Título: Registrar Conexión Agua	Prioridad 08
	Estimación
Descripción: Como prediador, puedo registrar los detalles de la conexión de agua.	

Figura Nº 4.8. Historia de usuario “Registrar Conexión Agua”

HISTORIA DE USUARIO	
Título: Registrar Unidad de Uso	Prioridad
	09
	Estimación
Descripción: como prediador, puedo registrar los detalles de la unidad de uso.	

Figura N° 4.9. Historia de usuario “Registrar Unidad de Uso”

HISTORIA DE USUARIO	
Título: Iniciar Sesión	Prioridad
	10
	Estimación
Descripción: como usuario, puedo ingresar con mi cuenta y password.	

Figura N° 4.10. Historia de usuario “Iniciar Sesión”

HISTORIA DE USUARIO	
Título: Registrar Usuario	Prioridad
	11
	Estimación

<p>Descripción:</p> <p>como administrador, puedo crear un nuevo usuario.</p>
--

Figura N° 4.11. Historia de usuario “Registrar Usuario”

HISTORIA DE USUARIO	
Título: Registra Rol	Prioridad 12
	Estimación
<p>Descripción:</p> <p>como administrador, puedo asignar a cada usuario su rol.</p>	

Figura N° 4.12. Historia de usuario “Registra Rol”

b.5. ESCRIBIR CRITERIOS DE ACEPTACIÓN DE LAS HISTORIAS DE USUARIO

En la escritura de los criterios de aceptación de las historias de usuario se empleo el formato que se propone en el anexo G. La historia de usuario 01 no tiene criterio de aceptación por que es de carácter técnico, por lo tanto el equipo de desarrollo se encargara de definirlo. Ver anexo C.

HISTORIA DE USUARIO	
Título: Obtener localización	Prioridad 02
	Estimación 13
Descripción: <p>como prediador, puedo utilizar el GPS con la finalidad de obtener la latitud y longitud y convertir en coordenadas UTM.</p>	
Criterio de aceptación: 1.- Cuando seleccione ubicar GPS, entonces debería poder obtener la latitud y la longitud. 2.- Cuando obtenga la latitud y la longitud, entonces se debería calcular las coordenadas con la formula UTM(este, norte).	

Figura Nº 4.13. Historia de usuario "Obtener localización"

HISTORIA DE USUARIO	
Título: Localizar Predio	Prioridad 03
	Estimación 20
Descripción: <p>Como prediador, puedo localizar el predio con la finalidad de obtener datos del predio.</p>	
Criterio de aceptación: 1.- Cuando compare UTM, entonces debería comparar las coordenadas UTM de catastro con las coordenadas UTM calculado.	

2.- Cuando localice el predio, entonces debería ver: actualizar predio o registrar nuevo predio.

Figura N° 4.14. Historia de usuario “Localizar Predio”

HISTORIA DE USUARIO	
Título: Identificar Cliente	Prioridad
	04
	Estimación
	20
<p>Descripción:</p> <p>como prediador, puedo verificar los datos del dueño del predio.</p>	
<p>Criterio de aceptación:</p> <p>1.- Cuando ubique el predio, entonces debería poder consultar los datos (nombres y apellido, DNI/RUC, teléfono, dirección, N°, Dpto., piso) y poder modificar.</p>	

Figura N° 4.15. Historia de usuario “Identificar Cliente”

HISTORIA DE USUARIO	
Título: Registrar Predio	Prioridad
	05
	Estimación
	05
<p>Descripción:</p> <p>Como prediador, puedo registrar los detalles del predio.</p>	

Criterio de aceptación:

- 1.- Cuando ingrese el tipo de propiedad, entonces debería poder seleccionar entre: 1.particular, 2.mixto, 3.público municipal, 4. público nacional.
- 2.- Cuando ingrese el tipo de construcción, entonces debería poder seleccionar entre: 1.lote sin construcción, 2.casa, 3. Edificio, 4. Multifamiliar, 5.abandonado.
- 3.- Cuando ingrese material de construcción, entonces debería poder seleccionar entre: 1.material de construcción, 2.adobe, 3.tierra y piedra, 4.madera, 5.otros.
- 4.- Cuando ingrese el N° de piso de la construcción, entonces debería validar que sea número.
- 5.- Cuando ingrese reservorio, entonces debería poder seleccionar entre: 1.no tiene reservorio, 2.cisterna, 3.fanque elevado.
- 6.- Cuando ingrese piscina, entonces debería poder seleccionar entre: 1.normal, 2.enterrado, 3.profundo.

Figura N° 4.16. Historia de usuario “Registrar Predio”

HISTORIA DE USUARIO	
Título: Registrar Estado Conexión	Prioridad 06
	Estimación 05
Descripción: Como prediador, puedo registrar el estado de conexión.	
Criterio de aceptación: 1.- Cuando ingrese distribución de facturación, entonces debería poder seleccionar entre: 1.dirección inmueble, 2.dirección de correspondencia, 3.banco/agencia. 2.- Cuando ingrese estado de conexión, entonces debería poder seleccionar entre 1.agua en uso, 2.agua cortado, 3.factible, 4.potencial, 5.clandestino.	

- 3.- Cuando el estado de conexión de agua sea potencial, entonces solo registrar los datos del predio.
- 4.- Cuando ingrese placa catastral, entonces debería poder seleccionar entre: 1.sí, 2.no, 3.crayola.

Figura N° 4.17. Historia de usuario “Registrar Estado Conexión”

HISTORIA DE USUARIO	
Título: Registrar Medidor	Prioridad 07
	Estimación 05
<p>Descripción:</p> <p>Como prediador, puedo registrar los datos del medidor</p>	
<p>Criterio de aceptación:</p> <p>1.- Cuando defina medidor instalado, debería poder seleccionar “Sí” O “NO” y ...</p> <p>2.- Cuando ingrese el estado medidor, entonces debería poder seleccionar entre: 1.operativo, 2.inoperativo, 3.alterado, 4.otros.</p> <p>3.- Cuando ingrese el dispositivo de seguridad medidor, entonces debería poder seleccionar entre: 1.anclaje metálico, 2.dado concreto, 3.otros, 4.ninguno.</p> <p>4.- Cuando ingrese llave de paso, entonces debía poder seleccionar entre: 1.una llave, 2.dos llaves, 3.ninguno.</p> <p>5.- Cuando ingrese pavimentación, entonces debería poder seleccionar entre: 1.tierra, 2.piedra, 3.adoquin, 4.asfaltado, 5.concreto.</p> <p>6.- Cuando ingrese vereda, entonces debería poder seleccionar entre: 1.tierra, 2.loseta, 3.concreto, 4.loza de concreto.</p>	

Figura N° 4.18. Historia de usuario “Registrar Medidor”

HISTORIA DE USUARIO	
Título: Registrar Conexión Agua	Prioridad 08
	Estimación 05
Descripción: <p>Como prediador, puedo registrar los detalles de la conexión de agua.</p>	
Criterio de aceptación: <p>1.- Cuando ingrese localización de caja medidor, entonces debería seleccionar entre: 1.en vereda, 2.en pared, 3.interno, 4.pistas, 5.no ubicado.</p> <p>2.- Cuando ingrese material caja, entonces debería seleccionar entre: 1.concreto, 2.ladrillo, 3.sin caja, 4.otros.</p> <p>3.- Cuando ingrese seguridad de caja, entonces debería poder seleccionar entre: 1.exagonal, 2.longitud, 3.otros.</p> <p>4.- Cuando ingrese tapa de la caja, entonces debería poder seleccionar entre: 1.fierro fundido, 2.fierro galvanizado, 3.termoplástico, 4.sin tapa.</p> <p>5.- cuando ingrese estado de tapa, entonces debería poder seleccionar entre: 1.bueno, 2.malo, 3.regular, 4.colapasado.</p> <p>6.- cuando ingrese profundidad de la caja, entonces debería poder seleccionar entre: 1.normal, 2.enterrado, 3.profundo.</p>	

Figura Nº 4.19. Historia de usuario "Registrar Conexión Agua"

HISTORIA DE USUARIO	
Título: Registrar Unidad de Uso	Prioridad 09
	Estimación 05

<p>Descripción:</p> <p>como prediador, puedo registrar los detalles de la unidad de uso.</p>
<p>Criterio de aceptación:</p> <p>1.- cuando ingrese el ámbito, entonces debería poder utilizar entre: 1.Huamanga, 2.Huanta.</p> <p>2.- cuando ingrese categoría, entonces debería poder seleccionar entre: 1.domestico, 2.comercial, 3.industrial, 4.estatal, 5.social, 6.mixto.</p> <p>3.- cuando ingrese cantidad de instalaciones, entonces debería validar que sea número y menor de 10.</p> <p>4.- cuando ingrese el tipo de uso, entonces debería poder relacionar entre: 1.empresa, 2.vivienda, 3.restaurant, 4.hotel, 5.colegio, 6.consultoria, 7.cuartel, 8.comisaria, 9.templos, 10.mercados, etc.</p> <p>5.- cuando ingrese número de personas, entonces debería validar un número.</p> <p>6.- cuando seleccione agregar unidad de uso, entonces debería poder agregar a la lista de unidad de uso.</p>

Figura Nº 4.20. Historia de usuario “Registrar Unidad de Uso”

HISTORIA DE USUARIO	
Título: Iniciar Sesión	Prioridad 10
	Estimación 03
<p>Descripción:</p> <p>como usuario, puedo ingresar con mi cuenta y password.</p>	
<p>Criterio de aceptación:</p> <p>1.- cuando ingrese mi password correcto, entonces debiera mostrar la página principal.</p>	

2.- cuando ingrese mi cuenta y mi PASSWORD incorrecto, entonces debiera mostrar mensaje de error.

Figura Nº 4.21. Historia de usuario “Iniciar Sesión”

HISTORIA DE USUARIO	
Título: Registrar Usuario	Prioridad 11
	Estimación 03
<p>Descripción:</p> <p>como administrador, puedo crear un nuevo usuario.</p>	
<p>Criterio de aceptación:</p> <p>1.- Cuando ingrese el nombre, entonces debe validar que sean caracteres. 2.- cuando ingrese el apellido, entonces debiera validar que sean caracteres. 3.- cuando ingrese el DNI, entonces debiera validar que sean números y solo de 8 digitos. 4.- Debería poder ingresar la dirección. 5.- Cuando seleccione el sexo, entonces debiera poder seleccionar “masculino” o “femenino”</p>	

Figura Nº 4.22. Historia de usuario “Registrar Usuario”

HISTORIA DE USUARIO	
Título: Registra Rol	Prioridad 12
	Estimación 03

<p>Descripción:</p> <p>como administrador, puedo asignar a cada usuario su rol.</p>
<p>Criterio de aceptación:</p> <p>1.- Cuando defina el rol del empleado, entonces debería poder seleccionar administrador, predador, supervisor.</p> <p>2.- cuando ingrese la denominación del usuario, entonces debería verificar si es único y mostrar error si ya existe la denominación del usuario.</p> <p>3.- cuando yo ingrese el PASSWORD, entonces debería verificar que tenga caracteres y número.</p>

Figura Nº 4.23. Historia de usuario “Registra Rol”

b.6. ESTIMACION DE HISTORIA DE USUARIO

Para la estimación se utilizó la técnica story points donde se determino el peso de cada historia de usuario. Tal como se muestra en el visual story mapping, ver anexo B.

Prioridad	Product Backlog Ítem	Estimación
01	definir la arquitectura	08
02	ubicar predio	13
03	localizar predio	20
04	identificar cliente	20
05	registrar predio	05
06	registrar estado de conexión	05
07	registrar de medidor	05
08	registrar conexión agua	05

09	registrar unidad de uso	05
10	iniciar sesión	03
11	registrar usuario	03
12	registrar rol	03

Tabla N° 4.3. Estimación de historia de usuario.

4.1.2. RELASE PLAN

A. IDENTIFICACION DE LA VELOCIDAD DEL EQUIPO DE DESARROLLO

Utilizamos los valores históricos de cada Sprint para ajustar la velocidad estimada hasta aproximarnos a la velocidad real del equipo, como se muestra el born down chart del anexo D, anexo E y anexo F. Compromisos de entrega:

Sprint 01: 41 puntos.

Sprint 02: 40 puntos.

Sprint 03: 14 puntos.

B. ARMADO DE LA RELÉASE PLAN

El release plan se obtuvo con ayuda de visual story mapping, ver Anexo B.

Sprint	Objetivo	Act. Date	Fin date	Dura c. (dias)	Veloc. Est.	Real Veloc.
1	Construir la plataforma de desarrollo y la funcionalidad de localización del predio.	18-abr-11	29-abr-11	14	41	41

2	Implementar la funcionalidad de características de conexión.	30-abr-11	14-may-11	14	40	40
3	Implementar la funcionalidad de usos del predio y administración.	14-may-11	25-may-11	10	14	16

Tabla N° 4.3. Estimación de historia de usuario.

4.1.3. SPRINT 1

El objetivo de este Sprint es construir la plataforma de desarrollo y desarrollar la funcionalidad de localización del predio. Las historias de usuario que nos comprometimos a entregar en este Sprint, son divididas en tareas por el equipo de desarrollo, para ayudarnos en la gestión de la tareas se utilizo el taskbord lo cual nos permite ver el estado (esperando, en progreso y terminado) de las tareas, ver anexo D.

Historia de usuario 01: “definir la arquitectura“.

Ver el anexo C, la arquitectura incremental propuesta consiste en:

- Cada Controlador tiene varias acciones basadas en CRUD (créate, read, update y delete).
- La clase del Modelo utiliza mini-ORM para asociar las clases con la base de datos Sqlite3.

- Cada acción del controlador esta asociado con una plantilla de Vista (index.erb, edit.erb, show.erb y new.erb)
- Se creo "catastro" que funcionará en el teléfono inteligente y "catastro-server" que funcionará como servidor, con las siguientes líneas de comando:

```
# rhogen app catastro
#rhoconnect app catastro-server
```

a. DAYLI MEETING

a.1. Comunicación y Sincronización

Se muestra todo los impedimentos que se tuvo en cada historia de usuario:

¿Qué se hizo ayer?	¿Qué se hará hoy?	¿Cuáles son los impedimentos que tengo para terminar mi trabajo?
Historia de usuario 1	Historia de usuario 1	Configurar emulador android con framework rhodes
Historia de usuario 2	Historia de usuario 2	Configurar mapkey google
Historia de usuario 3	Historia de usuario 3	Formula de conversión latitud, longitud a UTM

Tabla N° 4.4. Lista de Impedimentos.

b. DESARROLLO

Se empleo la siguiente línea de comando para crear controlador fuente:

```
#rhoconnect source utm
```

b.1. Convertir historias de usuario “obtener localización” en tareas

H.U. 02	Obtener Localización
Ítem	Tarea
1	escribir controlador controller.rb (método geolocalizacion, método visualizar mapa)
2	escribir vista new.erb

Tabla Nº 4.4. Lista de tareas de la historia de usuario “obtener localización”.

b.2. Programación

Tarea 01.- escribir controlador controller.rb (método geolocalizacion, método visualizar mapa)

```

controller.rb
3 # encoding: utf-8
4 # frozen_string_literal: true
5
6 class SettingsController < ActionController::Base
7
8
9   show_location
10
11   def show_location
12     @known_position = true
13     set_notification(:location, :known_position)
14     redirect_to :action => :new
15   end
16
17   def show_location_params
18     @lat = params[:latitude]
19     @long = params[:longitude]
20     @lat = @lat.to_s.split(".").first
21     @long = @long.to_s.split(".").first
22     map_params = {
23       :lat => @lat,
24       :long => @long,
25       :known_position => @known_position,
26       :show_location => true
27     }
28     @lat = @lat.to_f
29     @long = @long.to_f
30     @lat = @lat.to_s
31     @long = @long.to_s
32     @map_params = map_params.merge(@lat => @lat, @long => @long)
33   end
34
35   def create_map_params
36
37
38
39

```

Figura Nº 4.24. Clase controller.rb

Visualización en emulador android de la localización del predio:

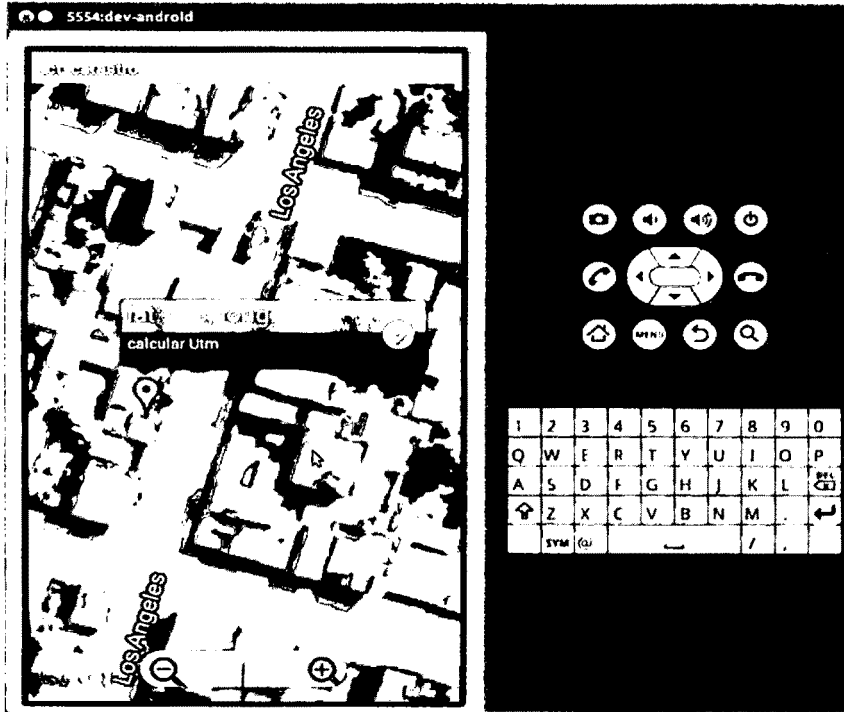


Figura Nº 4.25. Localización del predio.

Tarea 02.- escribir vista new.erb

```

new.erb
1 <div data-role="page">
2   <div data-role="header" data-position="inline">
3     <h1>New <%= @localizacion.title %></h1>
4     <a href="#<%= url_for :action => :index %>" class="ui-btn-left" data-icon="back" data-
5       direction="reverse">
6       Cancel
7     </a>
8   </div>
9   <div data-role="content">
10    <form method="POST" action="#<%= url_for :action => :create %>">
11      <input type="hidden" name="id" value="#<%= @localizacion.object %>" />
12
13      <div data-role="fieldcontain">
14        <label for="localizacion[lattitud]" class="fieldLabel">Lattitud</label>
15        <input type="text" id="localizacion[lattitud]" name="localizacion[lattitud]"
16          <%= placeholder("Lattitud") %> />
17      </div>
18
19      <div data-role="fieldcontain">
20        <label for="localizacion[longitud]" class="fieldLabel">Longitud</label>
21        <input type="text" id="localizacion[longitud]" name="localizacion[longitud]"
22          <%= placeholder("Longitud") %> />
23      </div>
24
25      <div data-role="fieldcontain">
26        <label for="localizacion[esteUtm]" class="fieldLabel">EsteUtm</label>
27        <input type="text" id="localizacion[esteUtm]" name="localizacion[esteUtm]"
28          <%= placeholder("EsteUtm") %> />
29      </div>
30
31      <div data-role="fieldcontain">
32        <label for="localizacion[oesteUtm]" class="fieldLabel">OesteUtm</label>
33        <input type="text" id="localizacion[oesteUtm]" name="localizacion[oesteUtm]"
34          <%= placeholder("OesteUtm") %> />
35      </div>
36      <input type="submit" value="Create" />
37    </form>
38  </div>
39 </div>

```

Figura Nº 4.26. Vista new.erb.

Visualización en emulador android de la latitud y longitud obtenida:

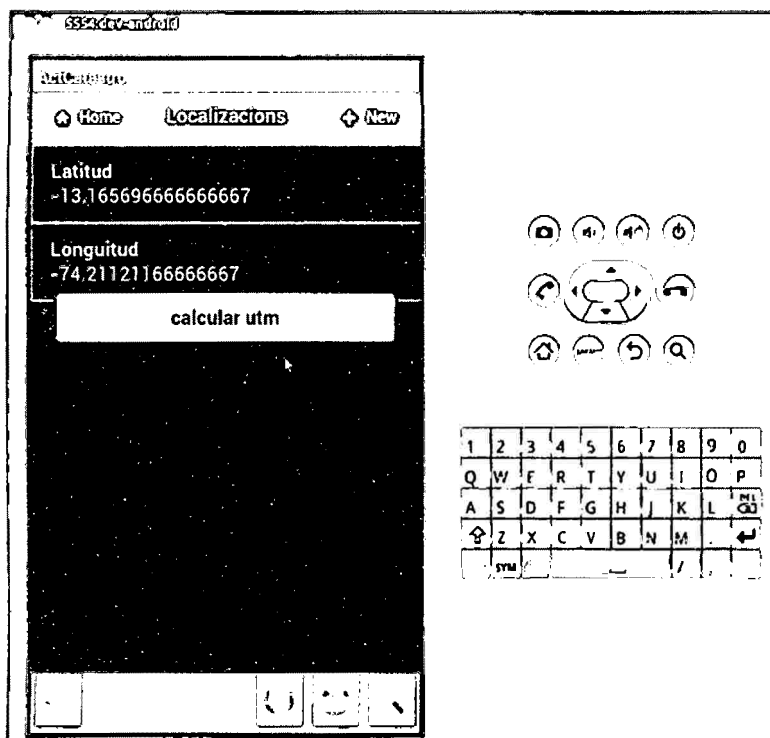


Figura N° 4.27. Latitud y longitud obtenida

b.1. Convertir historias de usuario “localizar predio” en tareas

H.U.03	Localizar Predio
Ítem	Tarea
1	escribir test utm_spect.rb (catastro-server)
2	escribir controlador fuente utm.rb (catastro-server)
3	escribir vista calcular_utm.erb
4	escribir controlador localizacion_controller.rb (método buscar predio)
5	escribir test unitario localizacion_spec.rb
6	escribir modelo localizacion.rb (método calcular utm)
7	escribir controller umt_controller(método index,new,show,delete)
8	escribir vista utm (edit.erb, index.erb, new.erb, show.erb)
9	escribir modelo umt.rb

Tabla N° 4.5. Lista de tareas de la historia de usuario “localizar predio”.

b.2. Programación

Tarea 01.- escribir test utm_spect.rb

```
utm_spect.rb x
1  .join(      .dirname(      ),      , :sync => false )
2
3  describe "UTM"
4    it "should behave like a utm object"
5      before(:all) do
6        setup_test_for(:utm)
7        @utm = {
8          :utm => "UTM",
9          :name => "UTM"
10       }
11     end
12
13     it "should present an utm object"
14       test_create(@utm)
15       test_query_size.should == 1
16       query_errors.should == {}
17     end
18
19     it "has a id attribute"
20       new_utm_id = test_create(@utm)
21       new_utm_id.should not be nil
22       create_errors.should == {}
23       rd[new_utm_id].should == @utm
24     end
25
26     it "should be able to update"
27       utm_id = test_create(@utm)
28       rd.should { |utm_id => @utm }
29       test_update({utm_id => { :utm => "UTM" }})
30       update_errors.should == {}
31       test_query
32       rd[utm_id][ :utm ].should == "UTM"
33     end
34
35     it "should be possible to delete"
36       utm_id = test_create(@utm)
37       rd.should { |utm_id => @utm }
38     end
39   end
40 end
```

Figura N° 4.28. Escribir test utm_spect.rb

Tarea 02.- escribir controlador utm.rb

```
utm.rb x
1  class Utm
2    attr_reader :source
3
4    def initialize(source)
5      @base = "http://utms.com/utm/utm/#{source}"
6      @source = source
7    end
8
9
10   def query(params = {})
11     parsed = parse( get("#{ @base }", params).body )
12     @result = {}
13     parsed.each { |item|
14       @result[item[:id]] = item[:utm]
15     }
16     @result
17   end
18
19   def create(create_hash)
20     res = post(@base, :utm => create_hash)
21     parse(
22       get(
23         "#{ @base }/#{ res[:utm][:id] }", {}
24       ).body
25     )[:utm][:id]
26   end
27
28   def update(update_hash)
29     obj_id = update_hash[:id]
30     update_hash.delete(:id)
31     put(
32       "#{ @base }/#{ obj_id }", :utm => update_hash
33     )
34   end
35
36   def delete(delete_hash)
37     delete(
38       "#{ @base }/#{ delete_hash[:id] }"
39     )
40   end
41
42   def login
43     # ...
44   end
45 end
```

Figura N° 4.29. Escribir controlador utm.rb

Tarea 03.- escribir vista calcular_utm.erb

```
calcula_utm.erb %
1 <div data-role="page" data-add-back-btn="false">
2   <div data-role="header" data-position="inline">
3     -h1-lat,long -vs- utm- /h1-
4   </div>
5   <div data-role="content">
6     <form method="POST" action="<%= url_for :controller => :Predio, :action
7
8     <ul data-role="listview">
9       -li-
10        <div class="itemLabel">Latitud</div>
11        <div class="itemValue"><%= @lati %></div>
12      </li>
13
14      <li>
15        <div class="itemLabel">Longitud</div>
16        <div class="itemValue"><%= @longi %></div>
17      </li>
18
19      <li>
20        <input type="hidden" form="est" name="est" value="<%= @est %>
21        <div class="itemLabel">Este Utm</div>
22        <div class="itemValue"><%= @est %></div>
23      </li>
24
25      <li>
26        <input type="hidden" form="north" name="north" value="<%= @
27        <div class="itemLabel">Norte Utm</div>
28        <div class="itemValue"><%= @north %></div>
29      </li>
30    </ul>
31    <input type="submit" value="buscar predio">
32
33  </form>
34 </div>
35 </div>
36
37 </div>
```

Figura N° 4.30. Vista calcular_utm.erb

Visualización en emulador android de la latitud-longitud y este-norte utm:

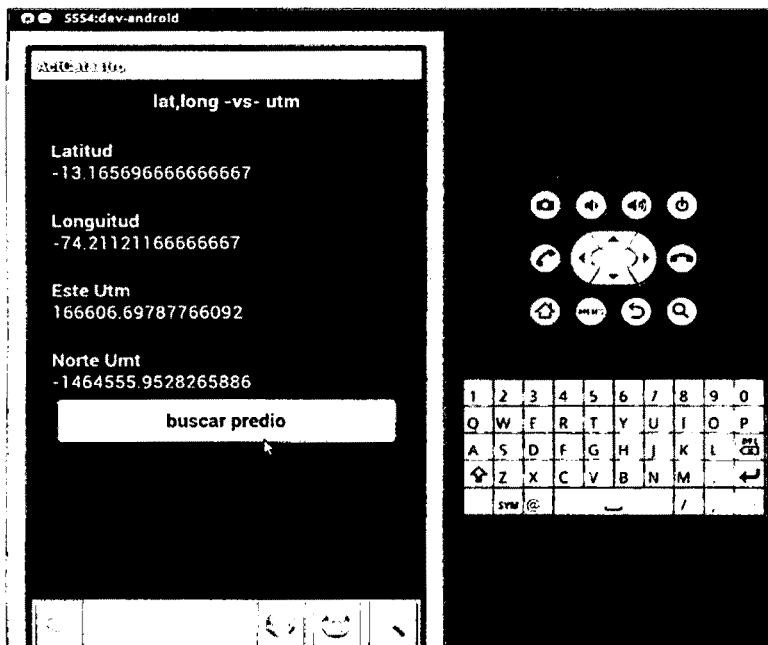


Figura N° 4.31. Vista calcular_utm.erb

Tarea 04.- escribir controlador localizacion_controller.rb (método buscar predio)

```

localizacion_controller.rb ✖
1 module Rho::RhoController
2   include Rho::Helpers/Streamer/Helper
3
4   class LocalizacionController < Rho::RhoController
5     #
6
7     # GET /localizacion/:id
8     def index
9       @localizaciones = find(:all)
10      @lat = @params['lat']
11      @long = @params['long']
12      @busca => /?pp/
13    end
14
15    # GET /localizacion/:id/est
16    def calcular_esta
17      @latl = @params['latl']
18      @longl = @params['longl']
19      @localizacion = find(@latl.to_f, @longl.to_f)
20      @est = @localizacion.est
21      @north = @localizacion.north
22      @distancia => /?distancia/
23    end
24
25    # GET /localizacion/:id/show
26    def show
27      @localizacion = find(@params[:id])
28      @localizacion
29      @latl => @lat, @longl => @long, @distancia => /?distancia/
30    end
31    redirect_to :action => :index
32  end
33
34
35  # GET /localizacion/new
36  def new
37    @localizacion = Localizacion.new
38  end
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

Figura Nº 4.32. Controlador localizacion_controller.rb

Tarea 05.- escribir test unitario localizacion_spec.rb

```

localizacion_spec.rb ✖
1 describe Localizacion do
2   # before(:all) { Localizacion.delete_all }
3   before(:each) { Localizacion.delete_all }
4
5   # before(:each) { Localizacion.delete_all }
6
7
8   it "should return 0.22413474915217572 of latitude radian of localization" do
9     localization = Localizacion.find(74, 112716)
10    localization.latRad.should == 0.22413474915217572
11  end
12
13
14  it "should return 74.112716 of longitude radian of localization" do
15    localization = Localizacion.find(74, 112716)
16    localization.longRad.should == 74.112716
17  end
18
19  it "should return 10.631484727777777 of zone" do
20    localization = Localizacion.find(74, 112716)
21    localization.zone.should == 10.631484727777777
22  end
23
24  it "should return 0.22413474915217572 of sinlat of localization" do
25    localization = Localizacion.find(74, 112716)
26    sinlat = localization.sinlat()
27    sinlat.should == 0.22413474915217572
28  end
29
30  it "should return normalized distance" do
31    localization = Localizacion.find(74, 112716)
32    localization.distOverMeridian.should == 0.22413474915217572
33  end
34
35  it "should return the rho of localization" do
36    localization = Localizacion.find(74, 112716)
37    localization.rho.should == 0.22413474915217572
38  end
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

Figura Nº 4.33. Test unitario localizacion_spec.rb

Visualización en emulador android del test unitario:

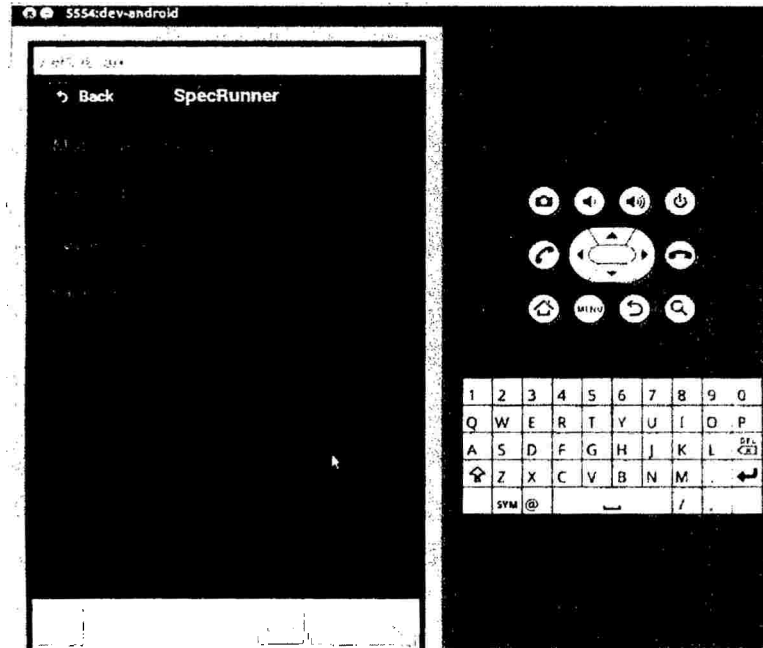


Figura Nº 4.34. Test unitario.

Tarea 06.- escribir modelo localizacion.rb (método calcular utm)

```
# rhogen model localizacion latitud,longitud,este,oeste
```

```

localizacion.rb $2
83 def calcular(n)
84   return n ** tan(latRad)
85 end
86 #Diferencias de longitud y sus potencias
87 def centralMeridian
88   return (zone * 6) + 15
89 end
90
91 def diflong(n)
92   return ((long - centralMeridian) / 180 * 2) ** n
93 end
94
95 #X (eastings)
96 def east
97   east1 = diflong(1) * coslat(1)
98   east2 = diflong(3) * coslat(3) * (psi(1) * tanlat(2)) / 6
99   east3 = diflong(5) * coslat(5) * (psi(3) * (1 - 6 * tanlat(2)) + psi(2)
100  psi(1) * (2 * tanlat(2)) - tanlat(4)) / 120
101   east4 = diflong(7) * coslat(7) * (psi(5) * 179 * tanlat(2) + 179 * tanlat(4)
102   x = nu * (east1 + east2 + east3 + east4)
103   return x
104 end
105
106 #Y (northings)
107 def north
108   north1 = sinlat(1) * diflong(2) * coslat(1) / 2
109   north2 = sinlat(1) * diflong(4) * coslat(3) * (psi(2) + psi(1) * tanlat(
110   north3 = sinlat(1) * diflong(6) * coslat(5) * (psi(4) * (1 - 24 * tanlat
111   - 6 * tanlat(2)) +
112   psi(2) * (1 - 32 * tanlat(2)) - psi(1) * (1 + tanlat(2)) + tanlat(4)) / 20
113   north4 = sinlat(1) * diflong(8) * coslat(7) * (psi(6) * 1111 * tanlat(2) + 543
114   (6)) / 40320
115   y =
116   return y
117 end
118 end

```

Figura Nº 4.35. Modelo localizacion.rb.

Tarea 07.- escribir controlador `utm_controller`(método `index,new,show,delete`)

```

utm_controller.rb ✖
1 class UtmController < ApplicationController
2   def index
3
4     @utms = Utm.find(:all)
5
6   end
7
8   def show
9     @utm = Utm.find(@params[:id])
10
11     respond_to do |format|
12       format.html { render :show }
13     end
14
15   end
16
17   def new
18     @utm = Utm.new
19
20     respond_to do |format|
21       format.html { render :new }
22     end
23
24   end
25
26   def edit
27     @utm = Utm.find(@params[:id])
28
29     respond_to do |format|
30       format.html { render :edit }
31     end
32
33   end
34
35   def delete
36     @utm = Utm.find(@params[:id])
37     @utm.destroy
38   end
39 end

```

Figura Nº 4.36. Controlador `utm_controller`.

Tarea 08.- escribir vista `utm` (`edit.erb`, `index.erb`, `new.erb`, `show.erb`)

```

edit.erb ✖
1 <%= data-role="page" %>
2
3
4 <%= data-role="header" data-position="inline" %>
5 <%= data-role="header" data-position="inline" %>
6 <%= data-role="header" data-position="inline" %>
7 <%= data-role="header" data-position="inline" %>
8 <%= data-role="header" data-position="inline" %>
9 <%= data-role="header" data-position="inline" %>
10 <%= data-role="header" data-position="inline" %>
11 <%= data-role="header" data-position="inline" %>
12 <%= data-role="header" data-position="inline" %>
13 <%= data-role="header" data-position="inline" %>
14 <%= data-role="content" %>
15 <%= data-role="content" %>
16 <%= data-role="content" %>
17 <%= data-role="content" %>
18 <%= data-role="content" %>
19 <%= data-role="content" %>
20 <%= data-role="content" %>
21 <%= data-role="content" %>
22 <%= data-role="content" %>
23 <%= data-role="content" %>
24 <%= data-role="content" %>
25 <%= data-role="content" %>
26 <%= data-role="content" %>
27 <%= data-role="content" %>
28 <%= data-role="content" %>
29 <%= data-role="content" %>
30 <%= data-role="content" %>
31 <%= data-role="content" %>
32 <%= data-role="content" %>

```

```

index.erb ✖
1 <%= data-role="page" %>
2
3
4 <%= data-role="header" data-position="inline" %>
5 <%= data-role="header" data-position="inline" %>
6 <%= data-role="header" data-position="inline" %>
7 <%= data-role="header" data-position="inline" %>
8 <%= data-role="header" data-position="inline" %>
9 <%= data-role="header" data-position="inline" %>
10 <%= data-role="header" data-position="inline" %>
11 <%= data-role="header" data-position="inline" %>
12 <%= data-role="header" data-position="inline" %>
13 <%= data-role="header" data-position="inline" %>
14 <%= data-role="content" %>
15 <%= data-role="content" %>
16 <%= data-role="content" %>
17 <%= data-role="content" %>
18 <%= data-role="content" %>
19 <%= data-role="content" %>
20 <%= data-role="content" %>
21 <%= data-role="content" %>
22 <%= data-role="content" %>
23 <%= data-role="content" %>
24 <%= data-role="content" %>
25 <%= data-role="content" %>
26 <%= data-role="content" %>
27 <%= data-role="content" %>
28 <%= data-role="content" %>

```

Figura Nº 4.37. Vista `utm` `edit.erb` e `index.erb`

```

new.erb X
1
2 <div data-role="page">
3
4 <div data-role="header" data-position="inline">
5 <h1>New <%= @utm.title %></h1>
6 <a href="#<%= url_for :action => :index %>" class="ui-btn-left" d
direction="reverse">
7   Cancel
8 </a>
9 </div>
10
11 <div data-role="content">
12 <form method="POST" action="#<%= url_for :action => :create %>">
13 <input type="hidden" name="id" value="#<%= @utm.object %>" >
14
15
16 <div data-role="fieldcontain">
17 <label for="utm[este]" class="fieldLabel">Este</label>
18 <input type="text" id="utm[este]" name="utm[este]" <%= p
19 </div>
20
21 <div data-role="fieldcontain">
22 <label for="utm[norte]" class="fieldLabel">Norte</label>
23 <input type="text" id="utm[norte]" name="utm[norte]" <%=
24 </div>
25
26 <input type="submit" value="Create" >
27 </form>
28 </div>
29 </div>
30 </div>
31 </div>
RHTML/ERB - AN...

show.erb X
1
2 <div data-role="page" data-add-back-btn="false">
3
4 <div data-role="header" data-position="inline">
5 <h1><%= @utm.este %></h1>
6 <a href="#<%= url_for :action => :index %>" class="ui-btn-
direction="reverse">
7   Back
8 </a>
9 <a href="#<%= url_for :action => :edit, :id => @utm.object
10   Edit
11 </a>
12 </div>
13
14 <div data-role="content">
15 <ul data-role="listview">
16
17 <li>
18 <div class="item-label">Este</div>
19 <div class="item-value"><%= @utm.este %></div>
20 </li>
21
22 <li>
23 <div class="item-label">Norte</div>
24 <div class="item-value"><%= @utm.norte %></div>
25 </li>
26
27 </ul>
28 </div>
29
30 </div>
Escalstro/app/utm/show.erb...
RHTML/ERB

```

Figura Nº 4.38. Vista utm new.erb e show.erb

Tarea 09.- escribir modelo umt.rb

rhogen model utm este,norte

```

utm.rb X
1 # The model has already been created by
2 # You can add more methods here
3 class Utm
4   include :
5
6   # Uncomment the following line to enable
7   # :sync
8
9   #add model specific code here
10 end

```

Figura Nº 4.39. Clase utm.rb

c. REVIEW MEETING

c.1. Verificar el cumplimiento del product backlog comprometido

H.U. N°	Criterio de aceptación	Estado
1	Cuando seleccione ubicar GPS, entonces debería poder obtener la latitud y la longitud.	OK
1	Cuando obtenga la latitud y la longitud, entonces se debería calcular las coordenadas con la formula UTM (latitud, longitud).	OK
2	Cuando compare UTM, entonces debería comparar las coordenadas UTM de catastro con las coordenadas UTM calculado.	OK
2	Cuando localice el predio, entonces debería ver: Provincia, Distrito, Sector, Ruta, Manzana y Local.	OK

Tabla N° 4.38. Review meeting.

d. RETROSPECTIVA

d.1. Mejorar los procesos de Scrum

¿Qué funciono bien?	Conforme
Planinng meeting	OK
Relase plan	OK
Sprint	OK
Dayli meeting	OK
Review meeting	OK
Retrospectiva	OK
¿Qué funciono mal?	Conforme
en el desarrollo no se pudo la automatización de test de cliente	FAIL

Tabla N° 4.7. Verificación de los procesos de Scrum.

4.1.4. SPRINT 2

El objetivo de este Sprint es implementar la funcionalidad de características de conexión. Las historias de usuario que nos comprometimos a entregar en este Sprint son divididas en tareas por el equipo de desarrollo, para ayudarnos en la gestión de las tareas se utilizó el taskbord lo cual nos permite ver el estado (esperando, en progreso y terminado) de las tareas, ver anexo E.

a. DAYLI MEETING

a.1. Comunicación y Sincronización

Identificando los impedimentos:

¿Qué se hizo ayer?	¿Qué se hará hoy?	¿Cuáles son los impedimentos que tengo para terminar mi trabajo?
Historia de usuario 4	Historia de usuario 4	manejo de JQuery
Historia de usuario 5	Historia de usuario 5	manejo de JQuery
Historia de usuario 6	Historia de usuario 6	Ninguno
Historia de usuario 7	Historia de usuario 7	Ninguno
Historia de usuario 8	Historia de usuario 8	Ninguno

Tabla N° 4.8. Lista de Impedimentos.

b. DESARROLLO

Se empleo las siguientes líneas de comando para crear el controlador

fuelle:

```
#rhoconnect source cliente  
#rhoconnect source predio  
#rhoconnect source conexion  
#rhoconnect source conexagua  
#rhoconnect source medidor
```

b.1. Convertir historias de usuario “identificar cliente” en tareas

H.U.04	Identificar Cliente
Ítem	Tarea
1	escribir test cliente_spec.rb
2	escribir controlador fuente cliente.rb
3	escribir modelo cliente.rb
4	escribir controlador cliente_controller.rb
5	escribir vista (edit.erb, index.erb, new.erb, show.erb)

Tabla Nº 4.9. Lista de tareas de la historia de usuario “identificar cliente”.

b.2. Programación

Tarea 01.- escribir test cliente_spec.rb

```
cliente_spec.rb ✖
1      .join(      .dirname(      ), .., 'pet helper')
2
3 describe 'cliente'
4   it_should_behave_like 'pet_helper'
5   before(:each)
6   setup_test_for      , :test_name
7   @cliente = {
8     nombre => 'Luisito',
9     direccion => 'Calle 123',
10    telefono => '42121177',
11    telefono2 => '91111111',
12    direccion2 => 'Calle Libertad',
13    edad => 120,
14    idata => 1,
15    ipita => '1'
16  }
17
18
19  it_should_process_cliente que:
20    test_create(@cliente)
21    test_query.size.should == 0
22    query_errors.should == {}
23
24
25  it_should_process_cliente create:
26    new_cliente_id = test_create(@cliente)
27    new_cliente_id.should_not be_nil
28    create_errors.should == {}
29    md[new_cliente_id].should == @cliente
30
31
32  it_should_process_cliente update:
33    cliente_id = test_create(@cliente)
34    md.should == {cliente_id => @cliente}
35    test_update({cliente_id => {telefono => '91111111'}})
36    update_errors.should == {}
37    test_query
38  end
39 end
```

Figura Nº 4.40. Escribir test cliente_spec.rb

Tarea 02.- escribir controlador fuente cliente.rb

```
cliente.rb
1 require 'json'
2 require 'test_client'
3
4 class Cliente < SourceAdapter
5   def initialize(source)
6     @base = "http://catastro-server.herokuapp.com/cliente"
7     super(source)
8   end
9
10  def query(params=nil)
11    parsed = parse( get( @base + "?#{params}", :json ).body )
12    @result = {}
13    parsed.each do |item|
14      @result[item[ "cliente ][ id ].to_s ] = item[ "cliente " ]
15    end
16    @result
17  end
18  def create(create_hash)
19    res = post( @base, :cliente => create_hash )
20    parse(
21      get( @base + "?#{params}", :json ).body
22    )[ "cliente ][ id ]"
23  end
24
25  def update(update_hash)
26    obj_id = update_hash[ :id ]
27    update_hash.delete( :id )
28    put( @base + "?#{params}", :cliente => update_hash )
29  end
30
31  def delete(delete_hash)
32    delete( @base + "?#{params}" )
33  end
34
35
36  def login
37    # TODO: login to your data source here if necessary.
38  end
39 end
```

Figura N° 4.41. Escribir controlador fuente cliente.rb

Tarea 03.- escribir modelo cliente.rb

```
# rhogen model cliente
```

```
nombres,apellidos,dniRuc,telefono,direccion,nro,dpto,plsa
```

```
cliente.rb
1 # The model has already been created by the framework
2 # You can add more methods here
3 class Cliente
4   include ActiveRecord::Base
5
6   # Uncomment the following line to enable sync with
7   enable :sync
8
9   #add model specific code here
10 end
```

Figura N° 4.42. Clase cliente.rb

Tarea 04.- escribir controlador cliente_controller.rb

```

cliente_controller.rb
1 class ClienteController < ActionController::Base
2   def index
3
4   end
5
6   def show
7     @cliente = Cliente.find(params[:id])
8     respond_to do |format|
9       format.html { render :show }
10      format.json { render :show }
11    end
12  end
13
14   def new
15     @cliente = Cliente.new
16     respond_to do |format|
17       format.html { render :new }
18       format.json { render :new }
19    end
20  end
21
22   def edit
23     @cliente = Cliente.find(params[:id])
24     respond_to do |format|
25       format.html { render :edit }
26       format.json { render :edit }
27    end
28  end
29
30   def create
31     @cliente = Cliente.new(params[:cliente])
32     @cliente.save
33     respond_to do |format|
34       format.html { redirect_to @cliente }
35       format.json { render :show }
36    end
37  end
38
39   def update
40     @cliente = Cliente.find(params[:id])
41     @cliente.update(params[:cliente])
42     respond_to do |format|
43       format.html { redirect_to @cliente }
44       format.json { render :show }
45    end
46  end
47
48   def destroy
49     @cliente = Cliente.find(params[:id])
50     @cliente.destroy
51     respond_to do |format|
52       format.html { redirect_to ClienteController.action_path(:index) }
53       format.json { render :nothing }
54    end
55  end
56 end

```

Figura N° 4.43. Controlador cliente_controller.rb

Tarea 05.- escribir vista cliente (edit.erb, index.erb, new.erb, show.erb)

```

edit.erb
1 <%= data-role="header" %>
2 <%= data-role="header" data-position="header" %>
3 <%= data-role="header" data-position="header" %>
4 <%= data-role="header" data-position="header" %>
5 <%= data-role="header" data-position="header" %>
6 <%= data-role="header" data-position="header" %>
7 <%= data-role="header" data-position="header" %>
8 <%= data-role="header" data-position="header" %>
9 <%= data-role="header" data-position="header" %>
10 <%= data-role="header" data-position="header" %>
11 <%= data-role="header" data-position="header" %>
12 <%= data-role="header" data-position="header" %>
13 <%= data-role="header" data-position="header" %>
14 <%= data-role="header" data-position="header" %>
15 <%= data-role="header" data-position="header" %>
16 <%= data-role="header" data-position="header" %>
17 <%= data-role="header" data-position="header" %>
18 <%= data-role="header" data-position="header" %>
19 <%= data-role="header" data-position="header" %>
20 <%= data-role="header" data-position="header" %>
21 <%= data-role="header" data-position="header" %>
22 <%= data-role="header" data-position="header" %>
23 <%= data-role="header" data-position="header" %>
24 <%= data-role="header" data-position="header" %>
25 <%= data-role="header" data-position="header" %>
26 <%= data-role="header" data-position="header" %>
27 <%= data-role="header" data-position="header" %>
28 <%= data-role="header" data-position="header" %>
29 <%= data-role="header" data-position="header" %>
30 <%= data-role="header" data-position="header" %>
31 <%= data-role="header" data-position="header" %>
32 <%= data-role="header" data-position="header" %>
33 <%= data-role="header" data-position="header" %>
34 <%= data-role="header" data-position="header" %>
35 <%= data-role="header" data-position="header" %>

```

```

index.erb
1
2 <div data-role="page">
3
4 <div data-role="header" data-position="header">
5 <%= data-role="header" %>
6 <%= data-role="header" data-position="header" %>
7 <%= data-role="header" data-position="header" %>
8 <%= data-role="header" data-position="header" %>
9 <%= data-role="header" data-position="header" %>
10 <%= data-role="header" data-position="header" %>
11 <%= data-role="header" data-position="header" %>
12 <%= data-role="header" data-position="header" %>
13 <%= data-role="header" data-position="header" %>
14 <div data-role="content">
15 <%= data-role="content" %>
16 <%= data-role="content" %>
17 <%= data-role="content" %>
18 <%= data-role="content" %>
19 <%= data-role="content" %>
20 <%= data-role="content" %>
21 <%= data-role="content" %>
22 <%= data-role="content" %>
23 <%= data-role="content" %>
24 <%= data-role="content" %>
25 <%= data-role="content" %>
26 <%= data-role="content" %>
27 <%= data-role="content" %>
28 <%= data-role="content" %>
29 <%= data-role="content" %>
30 <%= data-role="content" %>
31 <%= data-role="content" %>
32 <%= data-role="content" %>
33 <%= data-role="content" %>
34 <%= data-role="content" %>
35 <%= data-role="content" %>

```

Figura N° 4.44. Vista cliente edit.erb e index.erb

```

1 new.erb
2 <div data-role="page">
3   <div data-role="header" data-position="inline">
4     <h1>New <%= @cliente.title %>/h1>
5     <a href="#<%= url_for :action => :index %>" class="ui-btn-left" data-icon="back" data-direction="reverse">
6       Cancel
7     </a>
8   </div>
9   <div data-role="content">
10    <form method="POST" action="#<%= url_for :controller => :profile, :action => :create %>">
11      <input type="hidden" name="id" value="#<%= @cliente.object %>" />
12
13      <div data-role="fieldcontain">
14        <label for="cliente[nombres]" class="fieldlabel">Nombres</label>
15        <input type="text" id="cliente[nombres]" name="cliente[nombres]" />
16        <%= placeholder("Nombres") %>
17      </div>
18
19      <div data-role="fieldcontain">
20        <label for="cliente[apellidos]" class="fieldlabel">Apellidos</label>
21        <input type="text" id="cliente[apellidos]" name="cliente[apellidos]" />
22        <%= placeholder("Apellidos") %>
23      </div>
24
25      <div data-role="fieldcontain">
26        <label for="cliente[dniruc]" class="fieldlabel">Dniruc</label>
27        <input type="number" id="cliente[dniruc]" name="cliente[dniruc]" />
28        <%= placeholder("Dniruc") %>
29      </div>
30
31      <div data-role="fieldcontain">
32        <label for="cliente[telefono]" class="fieldlabel">Telefono</label>
33        <input type="number" id="cliente[telefono]" name="cliente[telefono]" />
34        <%= placeholder("Telefono") %>
35      </div>
36
37      <div data-role="fieldcontain">
38        <label for="cliente[direccion]" class="fieldlabel">Direccion</label>
39        <input type="text" id="cliente[direccion]" name="cliente[direccion]" />
40        <%= placeholder("Direccion") %>
41      </div>
42
43      <input type="submit" value="Crear" />
44    </form>
45  </div>
46 </div>

```

```

1 show.erb
2 <div data-role="page" data-add-back-btn="false">
3   <div data-role="header" data-position="inline">
4     <h1>New <%= @cliente.title %>/h1>
5     <a href="#<%= url_for :action => :index %>" class="ui-btn-left" data-icon="back" data-direction="reverse">
6       Back
7     </a>
8     <a href="#<%= url_for :action => :edit, :id => @cliente.object %>" class="ui-btn-right" data-icon="star">
9       Edit
10    </a>
11  </div>
12  <div data-role="content">
13    <ul data-role="listview">
14      <li>
15        <div class="itemLabel">Nombres</div>
16        <div class="itemValue"><%= @cliente.nombres %></div>
17      </li>
18      <li>
19        <div class="itemLabel">Apellidos</div>
20        <div class="itemValue"><%= @cliente.apellidos %></div>
21      </li>
22      <li>
23        <div class="itemLabel">Dniruc</div>
24        <div class="itemValue"><%= @cliente.dniruc %></div>
25      </li>
26      <li>
27        <div class="itemLabel">Telefono</div>
28        <div class="itemValue"><%= @cliente.telefono %></div>
29      </li>
30      <li>
31        <div class="itemLabel">Direccion</div>
32        <div class="itemValue"><%= @cliente.direccion %></div>
33      </li>
34    </ul>
35  </div>
36 </div>

```

Figura Nº 4.45. Vista cliente new.erb e show.erb

Visualización en emulador android para crear nuevo cliente:

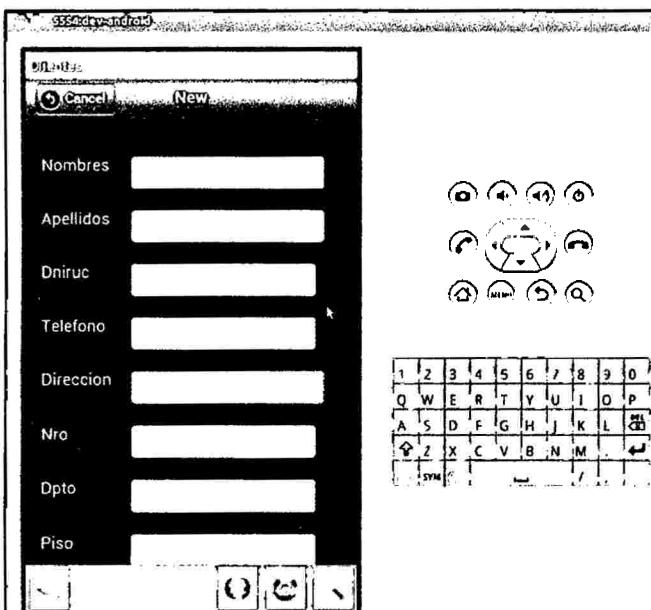


Figura Nº 4.46. Crear nuevo cliente.

b.1. Convertir historia de usuario “registrar predio” en tareas

H.U. 05	Registrar Predio
Ítem	Tarea
1	escribir test predio_spec.rb
2	escribir controlador fuente prediorb
3	escribir modelo predio.rb
4	escribir controlador predio_controller.rb
5	escribir vista (edit.erb, index.erb, new.erb, show.erb)

Tabla Nº 4.10. Lista de tareas de la historia de usuario “registrar predio”.

b.2. Programación

Tarea 01.- escribir test predio_spec.rb

```

predio_spec.rb ✖
1 require 'rails_helper'
2 require 'spec_helper'
3 describe 'Predio' do
4   before(:all) do
5     setup_test_for :test_data
6     @predio = {
7       'tipoPredio' => 'particular',
8       'direccion' => 'calle',
9       'numero' => '1234',
10      'provincia' => 'B',
11      'telefono' => '011 1234 5678',
12      'planta' => '0000'
13    }
14  end
15
16  it 'should process Predio query' do
17    test_create(@predio)
18    test_query.size.should == 0
19    query_errors.should == {}
20  end
21
22  it 'should process Predio create' do
23    new_predio_id = test_create(@predio)
24    new_predio_id.should_not be_nil
25    create_errors.should == {}
26    md[new_predio_id].should == @predio
27  end
28
29  it 'should process Predio update' do
30    predio_id = test_create(@predio)
31    md.should == {predio_id => @predio}
32    test_update({predio_id => { 'numero' => '1234' }})
33    update_errors.should == {}
34    test_query
35    md[predio_id][ 'numero' ].should == '1234'
36  end
37
38
39

```

Figura Nº 4.47. Escribir test predio_spec.rb

Tarea 02.- escribir controlador fuente predio.rb

```
predio.rb ✖
1 # require 'rest_client'
2 require 'rest_client'
3
4 Predio = SourceAdapter
5 initialize(source)
6 @base = http://localhost:3000/api/predios/
7 (source)
8
9
10 query(params=nil)
11 parsed = parse( get( @base + @predio ).body)
12 @result = {}
13 parsed.each {|item|
14   @result[item['predio_id'].to_s] = item['predio']
15   parsed
16 }
17
18 create(create_hash)
19 res = post(@base, create_hash)
20 parse(
21   get(
22     @base + @predio
23   ).body
24 )
25
26 update(update_hash)
27 obj_id = update_hash[:id]
28 update_hash.delete(:id)
29 put(
30   @base + @predio + '/' + obj_id,
31   update_hash
32 )
33
34 delete(delete_hash)
35 delete(
36   @base + @predio + '/' + delete_hash[:id]
37 )
38
39 login
```

RubyOnRails - Ancho

Figura N° 4.48. Escribir controlador fuente predio.rb

Tarea 03.- escribir modelo predio.rb

```
# rhogen model predio
```

```
tipoPropiedad,tipoConstruc,matConstruc,nroPisoConstruc,reservorio,piscina
```

```
predio.rb ✖
1 # require 'rails_helper'
2 # require 'rails_helper'
3
4 class Predio
5   attr_accessible :tipoPropiedad, :tipoConstruc, :matConstruc, :nroPisoConstruc, :reservorio, :piscina
6   attr_accessible :tipoPropiedad, :tipoConstruc, :matConstruc, :nroPisoConstruc, :reservorio, :piscina
7   enable_nested_attributes_for :predio
8
9   def initialize(attributes = {})
10     self.attributes = attributes
11   end
12
13   def save
14     # save logic
15   end
16 end
```

Figura N° 4.49. Clase predio.rb.

b.1. Convertir historia de usuario “registrar estado de conexión” en tareas

H.U.06	Registrar Estado de Conexión
Ítem	Tarea
1	escribir test conexion_spec.rb
2	escribir controlador fuente conexión.rb
3	escribir modelo conexion.rb
4	escribir controlador conexion_controller.rb
5	escribir vista (edit.erb, index.erb, new.erb, show.erb)

Tabla N° 4.11. Lista de tareas de la h. u. “registrar estado de conexión”.

b.2. Programación

Tarea 01.- escribir test conexion_spec.rb

```
conexion_spec.rb ✖
1 require 'rails_helper'
2
3 describe 'Conexion' do
4   it 'should behave like a helper' do
5     before(:each) do
6       setup_test_for :testuser
7       @conexion =
8         { distinct => 'agua en uso',
9           placa_catastral => '1'
10        }
11     end
12
13     it 'should process conexion query' do
14       test_create(@conexion)
15       test_query.size.should == 0
16       query_errors.should == {}
17     end
18
19     it 'should process conexion create' do
20       new_conexion_id = test_create(@conexion)
21       new_conexion_id.should_not be_nil
22       create_errors.should == {}
23       md[new_conexion_id].should == @conexion
24     end
25
26     it 'should process conexion update' do
27       connexion_id = test_create(@conexion)
28       nc.should == {conexion_id => @conexion}
29       test_update({conexion_id => {placa_catastral => 'crayola'}})
30       update_errors.should == {}
31       test_query
32       md[connexion_id]['placa_catastral'].should == 'crayola'
33     end
34
35     it 'should process conexion delete' do
36       connexion_id = test_create(@conexion)
37       nd.should == {conexion_id => @conexion}
38       test_delete(connexion_id)
39     end
40   end
41 end
```

Figura N° 4.54. Escribir test conexion_spec.rb.

Tarea 02.- escribir controlador fuente conexion.rb

```
conexion.rb #
1 require 'json'
2 require 'rest-client'
3
4 class Conexion < SourceAdapter
5   def initialize(source)
6     @base = "http://catastro-server.herokuapp.com/conexion"
7     super(source)
8   end
9
10  def query(params=nil)
11    parsed = parse( get("#{@base}/#{params}.json").body )
12    @result {}
13    parsed.each do |item|
14      @result[item["conexion"]["id"].to_s] = item["conexion"]
15    end
16  end
17
18  def create(create_hash)
19    res = post(@base, :conexion => create_hash)
20    parse( get("#{@base}/#{res['id']}.json").body )
21    @result[create_hash["conexion"]["id"]]
22  end
23
24  def update(update_hash)
25    obj_id = update_hash["id"]
26    update_hash.delete("id")
27    put("#{@base}/#{obj_id}.json", :conexion => update_hash)
28  end
29
30  def delete(delete_hash)
31    delete("#{@base}/#{delete_hash["id"]}.json")
32  end
33
34  def login
35    # TODO: login to the API and save the user's session
36  end
37 end
```

Figura Nº 4.55. Escribir controlador fuente conexion.rb

Tarea 03.- escribir modelo conexion.rb

```
# rhogen model conexion distFact,estConex,placaCatastral
```

```
conexion.rb #
1 # The model has already been created by the framework,
2 # You can add more methods here
3 class Conexion
4   include ActiveRecord::Base
5
6   # Uncomment the following line to enable sync with Co
7   enable :sync
8
9   belongs_to :conexagua_id, 'Conexagua'
10
11 #add model specific code here
12 end
```

Figura Nº 4.56. Clase conexión.rb.


```

1 <div data-role="page">
2
3 <div data-role="header" data-position="inline">
4 <h1>New <%= @conexion.title %></h1>
5 <a href="#url_for :action => :index %>" class="ui-btn-left" data-icon="back"
6   data-direction="reverse">
7   Cancel
8 </a>
9 </div>
10 <div data-role="content">
11 <form method="POST" action="#url_for :controller => :precios, :action => :
12   <input type="hidden" name="id" value="#{ @conexion.object %}" %>
13
14 <div data-role="fieldcontain">
15 <div data-role="fieldcontain">
16 <a href="#url_for :action => :new %>" data-role="action" data-icon="arrow-r"
17   data-transition="flip" rel="external">Registrar conexión aguas</a>
18 </div>
19 <input type="hidden" form="conexion[conexagua_id]" name="conexion[co
20   values="#{ @conexagua.object %}" %>
21 </div>
22 <div data-role="fieldcontain">
23 <label for="conexion[distfact]" class="select">Distributor Factura</label>
24 <select name="conexion[distfact]" id="conexion[distfact]">
25 <%= @listasdistfact.each { |key,value| distfact = @conexion[distfact] %>
26 <option value="#{value}">#{value}</option>
27 </select>
28 </div>
29 <div data-role="fieldcontain">
30 <label for="conexion[placaCatastral]" class="select">Placa Catastral</label>
31 <select name="conexion[placaCatastral]" id="conexion[placaCatastral]">
32 <%= @listasplacaCatastral.each { |key,value| placaCatastral = @conexion[placaCatastral] %>
33 <option value="#{value}">#{value}</option>
34 </select>
35 </div>
36 </div>
37 </div>
38 </div>
39 </div>
40 </div>
41 </div>
42 </div>
43 </div>
44 </div>
45 </div>
46 </div>
47 </div>
48 </div>
49 </div>
50 </div>
51 </div>
52 </div>
53 </div>
54 </div>
55 </div>
56 </div>
57 </div>
58 </div>
59 </div>
60 </div>
61 </div>
62 </div>
63 </div>
64 </div>
65 </div>
66 </div>
67 </div>
68 </div>
69 </div>
70 </div>
71 </div>
72 </div>
73 </div>
74 </div>
75 </div>
76 </div>
77 </div>
78 </div>
79 </div>
80 </div>
81 </div>
82 </div>
83 </div>
84 </div>
85 </div>
86 </div>
87 </div>
88 </div>
89 </div>
90 </div>
91 </div>
92 </div>
93 </div>
94 </div>
95 </div>
96 </div>
97 </div>
98 </div>
99 </div>
100 </div>

```

Figura N° 4.59. Vista conexión new.erb y show.erb

Visualización en emulador android para crear nueva conexión:

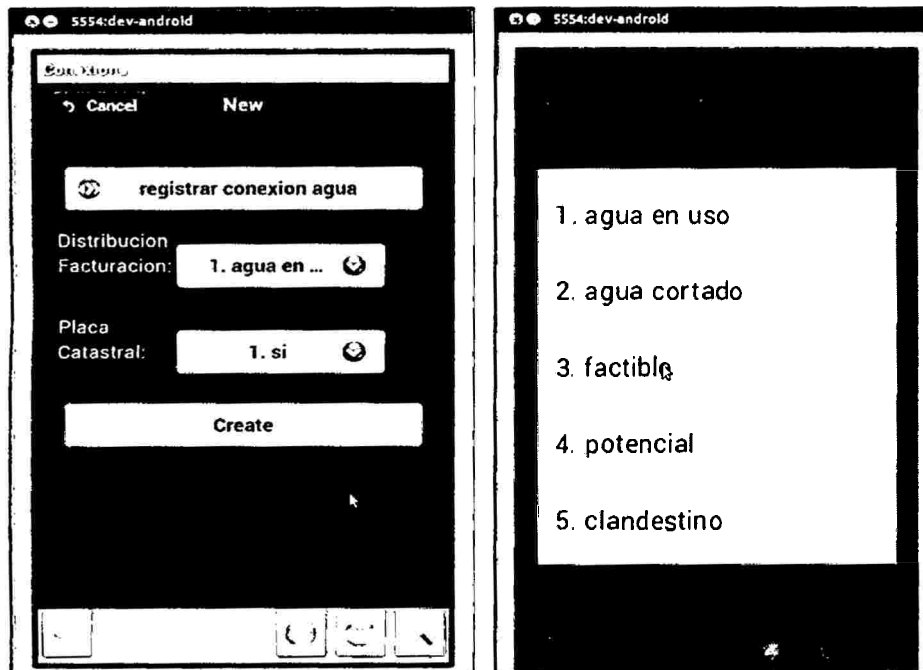


Figura N° 4.60. Crear nueva conexión.

b.1. Convertir historia de usuario “registrar medidor” en tareas

H.U.07:	Registrar Medidor
Ítem	Tarea
1	escribir test medidor_spec.rb
2	escribir controlador fuente medidor.rb
3	escribir modelo medidor.rb
4	escribir controlador medidor_controller.rb
5	escribir vista (edit.erb, index.erb, new.erb, show.erb)

Tabla Nº 4.12. Lista de tareas del h. u. “registrar medidor”.

b.2. Programación

Tarea 01.- escribir test medidor_spec.rb

```

medidor_spec.rb ✖
1   .join(      .dirname(      ), '..', 'spec_helper')
2
3 describe 'Medidor'
4   it should behave like 'SpecHelper'
5     before(:each) =>
6       setup_test_for      , test_name:
7       @medidor = {
8         medidor_id => '1',
9         medidor => 'operativa',
10        'descripcion' => 'operativa',
11        'descripcion' => 'operativa',
12        'descripcion' => 'operativa',
13        'descripcion' => 'operativa'
14      }
15
16
17   it 'should process Medidor query'
18     test_create(@medidor)
19     test_query.size.should == 0
20     query_errors.should == {}
21
22
23   it 'should process Medidor create'
24     new_medidor_id = test_create(@medidor)
25     new_medidor_id.should_not be_nil
26     create_errors.should == {}
27     nd[new_medidor_id].should == @medidor
28
29
30   it 'should process Medidor update'
31     medidor_id = test_create(@medidor)
32     nd.should { medidor_id => @medidor }
33     test_update({medidor_id => { descripcion => 'operativa' }})
34     update_errors.should == {}
35     test_query
36     nd[medidor_id]['descripcion'].should == 'operativa'
37
38

```

Figura Nº 4.61. Escribir test medidor_spec.rb.

Tarea 02.- escribir controlador fuente medidor.rb

```
medidor.rb *
1  class
2  end
3
4  Medidor < SourceAdapter
5  initialize(source)
6  @base = "http://datastra-backend.herokuapp.com/medidor"
7  (source)
8
9
10 query(params {})
11 parsed = parse( get( ).body)
12 @result = {}
13 parsed.each {|item|
14   @result[item["medidor"]["id"].to_s] = item["medidor"]
15   parsed
16 }
17
18 create(create_hash)
19 res = post(@base, create_hash => create_hash)
20 parse(
21   get( ).body
22 )["medidor"] = { id: }
23
24
25 update(update_hash)
26 obj_id = update_hash["id"]
27 update_hash.delete("id")
28 put( , medidor => update_hash)
29
30
31 delete(delete_hash)
32 delete( )
33
34
35 login
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
RubyOnRails - Ar
```

Figura Nº 4.62. Escribir controlador fuente medidor.rb.

Tarea 03.- escribir modelo medidor.rb

```
# rhogen model medidor
medInstaldo,estadoMed,dispSegMed,llavePaso,pavimentacion,vereda
```

```
medidor.rb *
1
2
3  Medidor
4  ::
5
6
7  enable :sync
8
9
10
```

Figura Nº 4.63. Clase medidor.rb.

b.1. Convertir historia de usuario “registrar conexión agua” en tareas

H.U. 08.-	Registrar Conexión Agua
Ítem	Tarea
1	escribir test conexagua_spec.rb
2	escribir controlador fuente conexagua.rb
3	escribir modelo conexagua.rb
4	escribir controlador conexagua_controller.rb
5	escribir vista (edit.erb, index.erb, new.erb, show.erb)

Tabla N° 4.13. Lista de tareas de la h. u. “registrar conexión agua”.

b.2. Programación

Tarea 01.- escribir test conexagua_spec.rb

```
conexagua_spec.rb ✖
1 require 'rails_helper'
2
3 describe 'Conexagua' do
4   it_should_behave_like 'SpecHelper' do
5     before(:each) do
6       setup_test_for :testuser
7       @conexagua = {
8         locCajaMed => 'en vereda',
9         matCaja => 'concreto',
10        tapaCaja => 'fierro fundido',
11        estTapa => 'bueno',
12        profCaja => 'normal'
13      }
14    end
15
16    it_should_process 'conexagua query' do
17      test_create(@conexagua)
18      test_query.size.should > 0
19      query_errors.should == {}
20    end
21
22    it_should_process 'conexagua create' do
23      new_conexagua_id = test_create(@conexagua)
24      new_conexagua_id.should_not be_nil
25      create_errors.should == {}
26      md[new_conexagua_id].should == @conexagua
27    end
28
29    it_should_process 'conexagua update' do
30      conexagua_id = test_create(@conexagua)
31      md.should == {conexagua_id => @conexagua}
32      test_update({conexagua_id => { profCaja => 'enterrado'}})
33      update_errors.should == {}
34      test_query
35      md[conexagua_id]['profCaja'].should == 'enterrado'
36    end
37
38  end
39 end
```

Figura N° 4.68. Escribir test conexagua_spec.rb.


```

1 <div data-role="page">
2
3 <div data-role="header" data-position="inline">
4 <h1>#{ @conexagua.ittle }</h1>
5 <a href="#<%= url_for :action => :index %>" class="ui-btn-left" data-icon="back"
direction="reverse">
6   Cancel
7 </a>
8 </div>
9
10 <div data-role="content">
11 <form method="POST" action="#<%= url_for :controller => :conexion, :action =>
12   <%= hidden_field :id, :value => @conexagua.object.id %>">
13
14   <%= if @medidor == nil %>
15     <div data-role="fieldcontain">
16       <a href="/app/medidor/new" data-role="button" data-icon="arrow-r"
17         data-transition="flip" rel="external">Registrar medidor</a>
18     </div>
19     <input type="hidden" form="conexagua[medidor_id]" name="conexagua[medidor_id]"
20       value="#<%= @medidor.object.id %>" />
21     <input type="text" />
22   </div>
23
24   <div data-role="fieldcontain">
25     <label for="conexagua[locCajamed]" class="select">Loc. caja medidor</label>
26     <select name="conexagua[locCajamed]" id="conexagua[locCajamed]">
27       <%= @locCajameds.each {|key,value| locCajamed : key, value } %>
28     </select>
29   </div>
30
31   <div data-role="fieldcontain">
32     <label for="conexagua[matCaja]" class="select">Material caja</label>
33     <select name="conexagua[matCaja]" id="conexagua[matCaja]">
34       <%= @matCajas.each {|key,value| matCaja : key, value } %>
35     </select>
36   </div>
37
38   <div data-role="fieldcontain">
39     <label for="conexagua[tapaCaja]" class="select">Tapa caja</label>
40     <select name="conexagua[tapaCaja]" id="conexagua[tapaCaja]">
41       <%= @tapaCajas.each {|key,value| tapaCaja : key, value } %>
42     </select>
43   </div>
44
45   <div data-role="fieldcontain">
46     <label for="conexagua[estadoTapa]" class="select">Estado tapa</label>
47     <select name="conexagua[estadoTapa]" id="conexagua[estadoTapa]">
48       <%= @estadoTapaCajas.each {|key,value| estadoTapa : key, value } %>
49     </select>
50   </div>
51
52   <div data-role="fieldcontain">
53     <label for="conexagua[profundidadTapa]" class="select">Profundidad tapa</label>
54     <select name="conexagua[profundidadTapa]" id="conexagua[profundidadTapa]">
55       <%= @profundidadTapaCajas.each {|key,value| profundidadTapa : key, value } %>
56     </select>
57   </div>
58
59   <input type="submit" value="Crear" />
60 </form>
61 </div>
62 </div>

```

Figura Nº 4.73. Vista conexión agua new.erb y show.erb.

Visualización en emulador android para crear conexión agua:

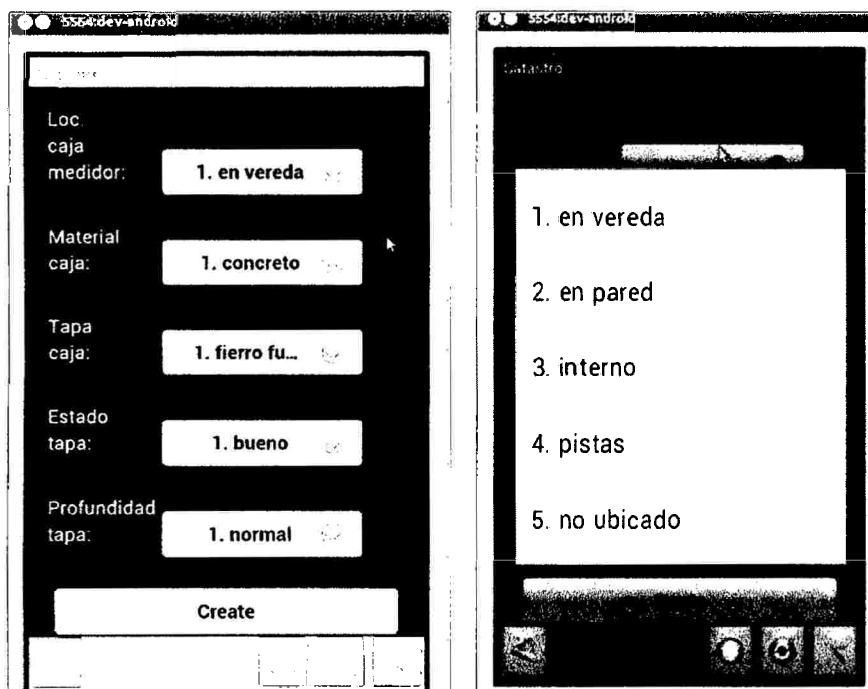


Figura Nº 4.74. Crear conexión agua.

c. REVIEW MEETING

c.1. Verificar el cumplimiento del product backlog comprometido

H.U. N°	Criterio de aceptación	Estado
4	Cuando ubique el predio, entonces debería poder consultar los datos (nombres y apellido, DNI/RUC, teléfono, dirección, N°, Dpto., piso) y poder modificar.	OK
5	Cuando ingrese el tipo de propiedad, entonces debería poder seleccionar entre: 1.particular, 2.mixto, 3.público municipal, 4. público nacional.	OK
5	Cuando ingrese el tipo de construcción, entonces debería poder seleccionar entre: 1.lote sin construcción, 2.casa, 3. Edificio, 4. Multifamiliar, 5.abandonado.	OK
5	Cuando ingrese material de construcción, entonces debería poder seleccionar entre: 1.material de construcción, 2.adobe, 3.tierra y piedra, 4.madera, 5.otros.	OK
5	Cuando ingrese el N° de piso de la construcción, entonces debería validar que sea número.	OK
5	Cuando ingrese reservorio, entonces debería poder seleccionar entre: 1.no tiene reservorio, 2.cisterna, 3.tanque elevado.	OK
5	Cuando ingrese piscina, entonces debería poder seleccionar entre: 1.normal, 2.enterrado, 3.profundo.	OK
6	Cuando ingrese distribución de facturación, entonces debería poder seleccionar entre: 1.dirección inmueble, 2.dirección de correspondencia, 3.banco/agencia.	OK
6	Cuando ingrese estado de conexión, entonces debería poder seleccionar entre 1.agua en uso, 2.agua cortado, 3.factible, 4.potencial, 5.clandestino.	OK
6	Cuando el estado de conexión de agua sea potencial, entonces solo registrar los datos del predio.	OK
6	Cuando ingrese placa catastral, entonces debería poder seleccionar entre: 1.si, 2.no, 3.crayola.	OK
7	Cuando defina medidor instalado, debería poder seleccionar "SÍ" O "NO" y ...	OK
7	Cuando ingrese el estado medidor, entonces debería poder seleccionar entre: 1.operativo, 2.inoperativo, 3.alterado, 4.otros.	OK

7	Cuando ingrese el dispositivo de seguridad medidor, entonces debería poder seleccionar entre: 1.lanclaje metálico, 2.dado concreto, 3.otros, 4.ninguno.	OK
7	Cuando ingrese llave de paso, entonces debía poder seleccionar entre: 1.una llave, 2.dos llaves, 3.ninguno.	OK
7	Cuando ingrese pavimentación, entonces debería poder seleccionar entre: 1.tierra, 2.piedra, 3.adoquin, 4.asfaltado, 5.concreto.	OK
7	Cuando ingrese vereda, entonces debería poder seleccionar entre: 1.tierra, 2.loseta, 3.concreto, 4.loza de concreto.	OK
8	Cuando ingrese localización de caja medidor, entonces debería seleccionar entre: 1.en vereda, 2.en pared, 3.interno, 4.pistas, 5.no ubicado.	OK
8	Cuando ingrese material caja, entonces debería seleccionar entre: 1.concreto, 2.ladrillo, 3.sin caja, 4.otros.	OK
8	Cuando ingrese seguridad de caja, entonces debería poder seleccionar entre: 1.exagonal, 2.longitud, 3.otros.	OK
8	Cuando ingrese tapa de la caja, entonces debería poder seleccionar entre: 1.fierro fundido, 2.fierro galvanizado, 3.termoplástico, 4.sin tapa.	OK
8	Cuando ingrese estado de tapa, entonces debería poder seleccionar entre: 1.bueno, 2.malo, 3.regular, 4.colapado.	OK
8	Cuando ingrese profundidad de la caja, entonces debería poder seleccionar entre: 1.normal, 2.enterrado, 3.profundo.	OK

Tabla N° 4.14. Review meeting.

d. RETROSPECTIVA

d.1. Mejorar los procesos de Scrum

¿QUÉ FUNCIONO BIEN?	CONFORME
Planing meeting	OK
Relase plan	OK
Sprint	OK
Dayli meeting	OK
Review meeting	OK

Retrospectiva	OK
¿QUÉ FUNCIONO MAL?	CONFORME
en el desarrollo no se pudo la automatización de test de cliente	FAIL

Tabla N° 4.15. Verificación de los procesos de Scrum.

4.1.5. SPRINT 3

El objetivo de este Sprint es implementar la funcionalidad de usos del predio y administración. Las historias de usuario que nos comprometimos a entregar en este Sprint son divididas en tareas por el equipo de desarrollo, para ayudarnos en la gestión de la tareas se utilizo el taskbord lo cual nos permite ver el estado (esperando, en progreso y terminado) de las tareas, ver anexo F.

a. DAYLI MEETING

a.1. Comunicación y Sincronización

Identificando los impedimentos:

¿Qué se hizo ayer?	¿Qué se hará hoy?	¿Cuáles son los impedimentos que tengo para terminar mi trabajo?
Historia de usuario 9	Historia de usuario 9	Ninguno
Historia de usuario 10	Historia de usuario 10	Ninguno
Historia de usuario 11	Historia de usuario 11	Ninguno
Historia de usuario 12	Historia de usuario 12	Ninguno

Tabla N° 4.16. Lista de Impedimentos.

b. DESARROLLO

Se empleo las siguientes líneas de comando para crear el modelo y el controlador fuente:

```
#rhoconnect source unidaduso
```

```
#rhoconnect source usuario
```

```
#rhoconnect source cuenta
```

```
#rhoconnect source rol
```

b.1. Convertir historia de usuario “registrar unidad de uso” en tareas

H.U. 09:	Registrar Unidad de Uso
Ítem	Tarea
1	escribir test unidadUso_spec.rb
2	escribir controlador fuente unidadUso.rb
3	escribir modelo unidaduso.rb
4	escribir controlador unidaduso_controller.rb
5	escribir vista (edit.erb, index.erb, new.erb, show.erb)

Tabla Nº 4.17. Lista de tareas de la h. u. “registrar unidad de uso“.

b.2. Programación

Tarea 01.- escribir test unidadUso_spec.rb

```
unidaduso_spec.rb ✖
1 require 'rails_helper'
2 describe 'unidaduso' do
3   it 'should behave like a resource' do
4     before(:each) do
5       setup_test_environment
6       @unidaduso =
7         FactoryBot.create(:unidaduso)
8       @unidaduso.reload
9       @unidaduso.reload
10      @unidaduso.reload
11    end
12  end
13
14  it 'should respond to create' do
15    test_create(@unidaduso)
16    test_query_size.should == 1
17    query_errors.should == []
18  end
19
20  it 'should respond to create' do
21    new_unidaduso_id = test_create(@unidaduso)
22    new_unidaduso_id.should not be nil
23    create_errors.should == []
24    md[new_unidaduso_id].should == @unidaduso
25  end
26
27  it 'should respond to update' do
28    unidaduso_id = test_create(@conexion)
29    md.should == {unidaduso_id => @conexion}
30    test_update(@unidaduso_id => {'estado' => 'suspendido'})
31    update_errors.should == []
32    test_query
33    md[unidaduso_id][@conexion].should == @conexion
34  end
35
36  it 'should respond to delete' do
37    unidaduso_id = test_create(@unidaduso)
38    test_delete(unidaduso_id)
39  end
end
```

Figura Nº 4.75. Escribir test unidadUso_spec.rb.

Tarea 02.- escribir controlador fuente unidadUso.rb

```
unidaduso.rb %
1  # require 'rest_client'
2  require 'rest_client'
3
4  class Unidaduso < SourceAdapter
5    def initialize(source)
6      @base = "http://localhost:3000/api/v1/units/#{source}"
7    end
8
9
10   def query(params = {})
11     parsed = parse( RestClient.get(@base, { :params => params }).body )
12     @result = {}
13     parsed.each { |item|
14       @result[item["unit_id"].to_s] = item["unit_name"]
15     }
16     parsed
17
18   def create(create_hash)
19     res = RestClient.post(@base, create_hash)
20     parse( RestClient.get(@base, { :params => create_hash }).body )
21   end
22   def update(update_hash)
23     obj = Unidaduso.find(update_hash[:id])
24     update_hash.delete(:id)
25     RestClient.put(@base, update_hash)
26
27   def delete(delete_hash)
28     delete(@base, delete_hash)
29
30   def login
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2640
2641

```



```

1 <div data-role="page">
2
3 <div data-role="header" data-position="inline">
4 <h1 data-role="page">unidad de uso</h1>
5 <a href="#">url_for action => index</a> <span class="ui-btn-left" data-icon="back"
6 direction="reverse">
7 Cancel
8 </a>
9 </div>
10 <div data-role="content">
11 <div data-role="form">
12 <input type="text" name="id" value="#{unidaduso.object.id}" />
13
14 <div data-role="fieldcontain">
15 <input type="text" name="categoria" class="select">
16 <input type="text" name="cantidad" id="cantidad">
17 <input type="text" name="tipo" id="tipo" />
18 <input type="text" name="numero" id="numero" />
19 <input type="text" name="unidaduso" id="unidaduso" />
20
21 </div>
22 </div>
23
24 <div data-role="fieldcontain">
25 <input type="button" value="Crear" />
26
27 </div>
28
29 </div>
30
31 </div>

```

Figura Nº 4.80. Vista unidad de uso new.erb y show.erb.

Visualización en emulador android para crear nuevo unidad de uso.

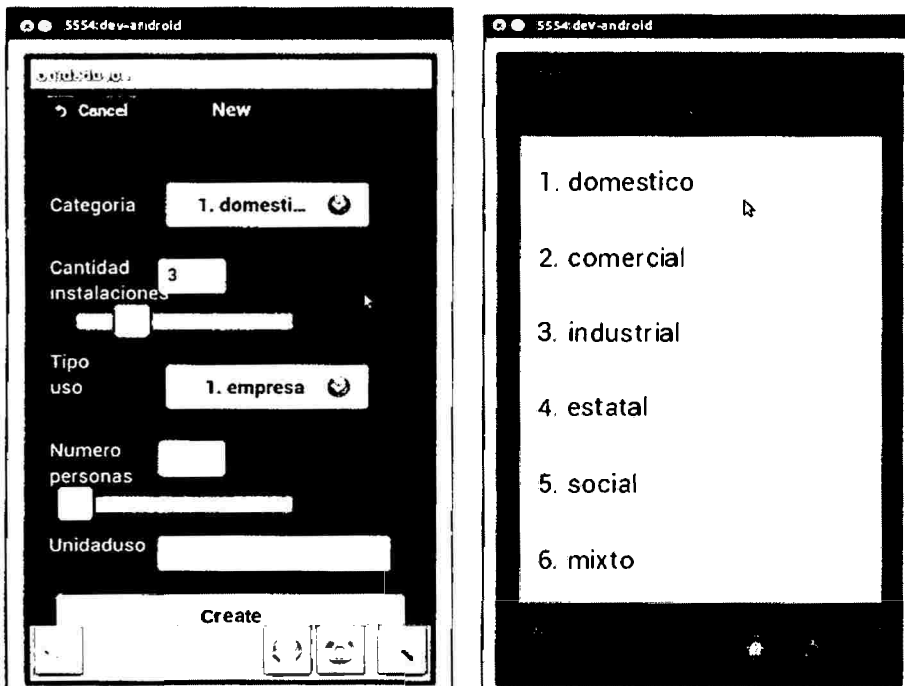


Figura Nº 4.81. Crear nuevo unidad de uso.

b.1. Convertir historia de usuario “iniciar sesión” en tareas

H.U. 10:	Iniciar Sesión
Item	Tarea
1	escribir controlador controller.rb
2	escribir vista login.erb

Tabla N° 4.18. Lista de tareas de la h. u. “iniciar sesión”.

b.2. Programación

Tarea 01.- escribir controlador controller.rb

```

controller.rb
6 | SettingsController < Rho::RhoController
7 |
8 |
9 | index
10 | @msg = @params[:msg]
11 |
12 |
13 |
14 | login
15 | @msg = @params[:msg]
16 | @url = @params[:url]
17 |
18 |
19 | login_failed
20 | errCode = @params[:errCode].to_i
21 | errCode = 0
22 |
23 |   navigate :root, options_path
24 |   dosync
25 |
26 |   errCode = 0
27 |   @msg = @params[:msg]
28 |
29 |
30 |   @msg = @msg.length
31 |   @msg = (errCode).message
32 |
33 |
34 |   navigate :root, options_path, { :url => @url, :msg => @msg }
35 |
36 |
37 |
38 | def login
39 |   @params[:login] = @params[:url]
40 |

```

Figura N° 4.82. Controlador controller.rb.

Tarea 02.- escribir vista login.erb

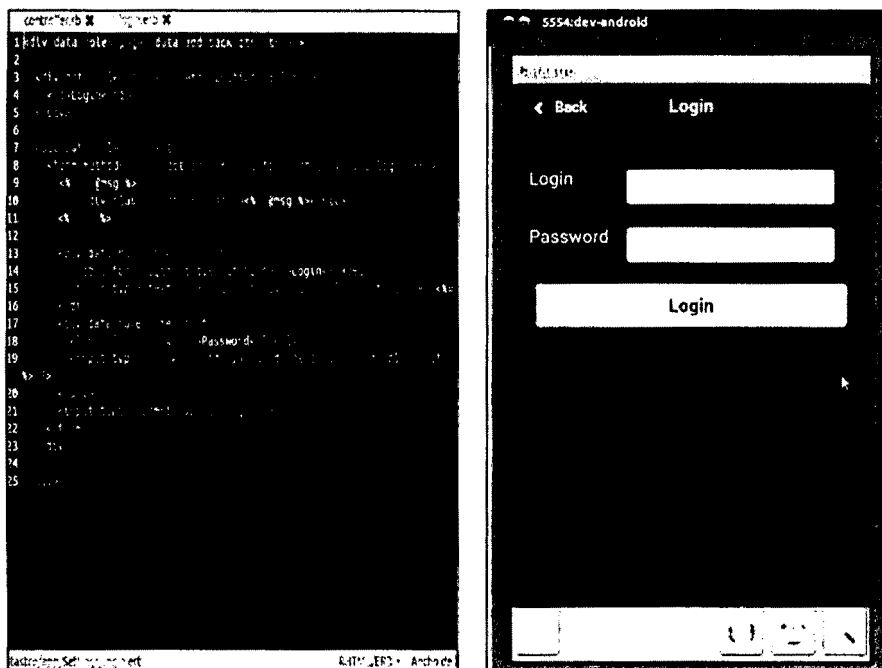


Figura N° 4.83. Vista login.erb y visualización en el emulador android.

b.1. Convertir historia de usuario “registrar usuario” en tareas

H.U. 11.-	Registrar Usuario
Ítem	Tarea
1	escribir test usuario_spec.rb
2	escribir controlador fuente usuario.rb
3	escribir modelo usuario.rb
4	escribir controlador usuario_controller.rb
5	escribir vista (edit.erb, index.erb, new.erb, show.erb)

Tabla N° 4.19. Lista de tareas de la h. u. “registrar usuario”.

b.2. Programación

Tarea 01.- escribir test usuario_spec.rb

```
1 usuario_spec.rb ✕
2   .join(      .dirname(      ), .., :app, :helpers )
3 describe 'User' do
4   it_should_behave_like 'RSpecHelper'
5   before(:each) {
6     setup_test_for 'usuario'
7     @usuario = {
8       nombre => 'Gerardo',
9       apellido => 'Medina',
10      email => 'gerardo@libertad.com',
11      direccion => 'Calle Libertad',
12      telefono => '33444444'
13    }
14  }
15
16  it 'should be able to create a new user' do
17    test_create(@usuario)
18    test_query.size.should == 1
19    query_errors.should == {}
20  end
21
22  it 'should be able to create a new user' do
23    new_usuario_id = test_create(@usuario)
24    new_usuario_id.should_not be_nil
25    create_errors.should == {}
26    md[new_usuario_id].should == @usuario
27  end
28
29  it 'should be able to update a user' do
30    usuario_id = test_create(@usuario)
31    md.should { |usuario_id| usuario_id == @usuario }
32    test_update({usuario_id => { nombre => 'Gerardo' }})
33    update_errors.should == {}
34    test_query
35    md[usuario_id][ :nombre ].should == 'Gerardo'
36  end
37 end
```

Figura N° 4.84. Escribir test usuario_spec.rb.

Tarea 02.- escribir controlador fuente usuario.rb

```
1 usuario.rb ✕
2 class Usuario
3   attr_reader :source
4
5   def initialize(source)
6     @base = ActionController::Base
7     @source = source
8   end
9
10  def query(params = {})
11    parsed = parse( get( :source ).body )
12    @result = {}
13    parsed.each { |item|
14      @result[item[:id]] = { :id => item[:id], :name => item[:nombre] }
15    }
16  end
17
18  def create(create_hash)
19    res = post(@base, create_hash => create_hash)
20    parse( res )
21  end
22  def update(update_hash)
23    obj_id = update_hash[:id]
24    update_hash.delete(:id)
25    put( obj_id, update_hash )
26  end
27  def delete(delete_hash)
28    delete(delete_hash[:id])
29  end
30
31  def login
32  end
33 end
```

Figura N° 4.85. Escribir controlador fuente usuario.rb

Tarea 03.- escribir modelo usuario.rb

rhogen model usuario nombre,apellidos,dni,direccion,sexo

```
1
2
3      Usuario
4      : :
5
6
7  enable :show
8      rol :id, 'Rol'
9      cuenta :id, 'cuenta'
10
11
12
13
```

Figura Nº 4.86. Clase usuario.rb

Tarea 04.- escribir controlador usuario_controller.rb

```
1 require 'rho/rhocontroller'
2 require 'helpers/browser_helper'
3
4 class UsuarioController < Rho::RhoController
5   include
6
7   # GET /usuarios
8   def index
9     @usuarios = find(:all)
10    @back => '/app'
11  end
12
13  # GET /usuarios/:id
14  def show
15    @usuario = find(@params[:id])
16    @rol = find(@usuario.rol_id)
17    @cuenta = find(@usuario.cuenta_id)
18    if @usuario
19      @action => :show, @back => ( :action => :index)
20    else
21      redirect :action => :index
22    end
23  end
24
25  # GET /usuarios/new
26  def new
27    rol = find(:all).first
28    if rol == nil
29      create({ denominacion => 'Prediador'})
30      create({ denominacion => 'Supervisor'})
31      create({ denominacion => 'Administrador'})
32    end
33    if @params[: cuenta ] != nil
34      @cuenta = create(@params[: cuenta ])
35    end
36
37    @rols = find(:all)
38  end
39 end
```

Figura Nº 4.87. Controlador usuario_controller.rb

Tarea 05.- escribir vista usuario (edit.erb, index.erb, new.erb, show.erb)

<pre> 1 <%= data-role="page" %> 2 3 <%= data-role="header" data-position="inline" %> 4 <%=><%= @usuario.nombre %> %> 5 show, id => @usuario.object %> class="ui- 6 data-role="button" data-direction="reverse"> 7 Show 8 </a %> 9 delete, id => @usuario.object %> class="ui- 10 data-role="button" data-direction="reverse"> 11 Delete 12 </a %> 13 </div> 14 15 <%= data-role="content" %> 16 <%= form method="POST" actions="#" for action => update %> 17 <input type="hidden" name="id" value="#{@usuario.object.id}" %> 18 19 <%= data-role="fieldcontata" %> 20 <div class="ui-field-contain" data-role="field-label">Nombre: <input 21 type="text" id="usuario_nombre" name="usuario[nombre]" 22 value="#{@usuario.nombre}" data-placeholder="Nombre" %> %> 23 </div> 24 25 <%= data-role="fieldcontata" %> 26 <div class="ui-field-contain" data-role="field-label">Apellidos: <input 27 type="text" id="usuario_apellidos" name="usuario[apellidos]" 28 value="#{@usuario.apellidos}" data-placeholder="Apellidos" %> %> 29 </div> 30 31 <%= data-role="fieldcontata" %> 32 <div class="ui-field-contain" data-role="field-label">Dir: <input 33 type="text" id="usuario_dir" name="usuario[dir]" 34 value="#{@usuario.dir}" data-placeholder="Dir" %> %> 35 </div> 36 37 <%= data-role="fieldcontata" %> </pre>	<pre> 1 <%= data-role="page" %> 2 3 <%= data-role="header" data-position="inline" %> 4 <%=><%= @usuario.nombre %> %> 5 show, id => @usuario.object %> class="ui- 6 data-role="button" data-direction="reverse" data-theme="a" %> 7 Show 8 </a %> 9 new, id => @usuario.object %> class="ui- 10 data-role="button" data-direction="reverse" data-theme="a" %> 11 New 12 </a %> 13 14 <%= data-role="content" %> 15 <%= data-role="listview" %> 16 <%=><%= @usuarios.each @usuario %> 17 18 show, id => @usuario.object %> 19 <%= @usuario.nombre %> 20 21 </a %> 22 23 </a %> 24 25 </div> 26 </div> </pre>
---	---

Figura N° 4.88. Vista usuario edit.erb e index.erb.

<pre> 1 <%= data-role="page" %> 2 3 <%= data-role="header" data-position="inline" %> 4 <%=><%= @usuario.title %> 5 show, id => @usuario.object %> class="ui- 6 data-role="button" data-direction="reverse"> 7 Show 8 </a %> 9 edit, id => @usuario.object %> class="ui- 10 data-role="button" data-direction="reverse"> 11 Edit 12 </a %> 13 14 <%= data-role="content" %> 15 <%= form method="POST" actions="#" for action => create %> 16 <input type="hidden" name="id" value="#{@usuario.object.id}" %> 17 18 <input type="text" id="usuario_nombre" name="usuario[nombre]" 19 value="#{@usuario.nombre}" data-placeholder="Nombre" %> 20 21 <input type="text" id="usuario_apellidos" name="usuario[apellidos]" 22 value="#{@usuario.apellidos}" data-placeholder="Apellidos" %> 23 24 <input type="text" id="usuario_dir" name="usuario[dir]" 25 value="#{@usuario.dir}" data-placeholder="Dir" %> 26 27 28 <input type="button" value="Registrar cuenta" %> 29 30 </div> 31 32 <%= data-role="fieldcontata" %> 33 <div class="ui-field-contain" data-role="field-label">Nombre: <input 34 type="text" id="usuario_nombre" name="usuario[nombre]" 35 value="#{@usuario.nombre}" data-placeholder="Nombre" %> %> 36 </div> 37 38 <%= data-role="fieldcontata" %> 39 <div class="ui-field-contain" data-role="field-label">Apellidos: <input 40 type="text" id="usuario_apellidos" name="usuario[apellidos]" 41 value="#{@usuario.apellidos}" data-placeholder="Apellidos" %> %> 42 </div> 43 44 <%= data-role="fieldcontata" %> 45 <div class="ui-field-contain" data-role="field-label">Dir: <input 46 type="text" id="usuario_dir" name="usuario[dir]" 47 value="#{@usuario.dir}" data-placeholder="Dir" %> %> 48 </div> 49 50 <input type="button" value="Registrar cuenta" %> 51 52 </div> </pre>	<pre> 1 <%= data-role="page" data-add-back-btn="false" %> 2 3 <%= data-role="header" data-position="inline" %> 4 <%=><%= @usuario.nombre %> %> 5 show, id => @usuario.object %> class="ui- 6 data-role="button" data-direction="reverse"> 7 Show 8 </a %> 9 edit, id => @usuario.object %> class="ui- 10 data-role="button" data-direction="reverse"> 11 Edit 12 </a %> 13 14 <%= data-role="content" %> 15 <%= data-role="listview" %> 16 17 <div class="ui-item-label">Nombre: <input 18 type="text" id="usuario_nombre" name="usuario[nombre]" 19 value="#{@usuario.nombre}" data-placeholder="Nombre" %> %> 20 </div> 21 22 <div class="ui-item-label">Apellidos: <input 23 type="text" id="usuario_apellidos" name="usuario[apellidos]" 24 value="#{@usuario.apellidos}" data-placeholder="Apellidos" %> %> 25 </div> 26 27 <div class="ui-item-label">Dir: <input 28 type="text" id="usuario_dir" name="usuario[dir]" 29 value="#{@usuario.dir}" data-placeholder="Dir" %> %> 30 </div> 31 32 <div class="ui-item-label">Direccion: <input 33 type="text" id="usuario_direccion" name="usuario[direccion]" 34 value="#{@usuario.direccion}" data-placeholder="Direccion" %> %> 35 </div> 36 37 <input type="button" value="Registrar cuenta" %> 38 39 </div> </pre>
--	--

Figura N° 4.89. Vista usuario new.erb y show.erb.

Visualización en el emulador android para crear nuevo usuario:

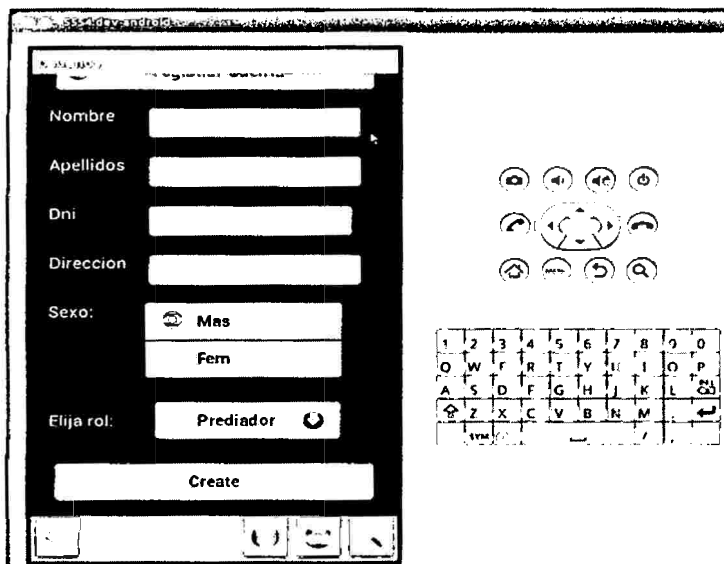


Figura N° 4.90. Crear nuevo usuario.

b.1. Convertir historia de usuario “registrar rol” en tareas

H.U. 12:	Registrar Rol
Ítem	Tarea
1	escribir test rol_spec.rb
2	escribir controlador fuente rol.rb
3	escribir modelo rol.rb
4	escribir controlador rol_controller.rb
5	escribir vista (edit.erb, index.erb, new.erb, show.erb)

Tabla N° 4.19. Lista de tareas de la h. u. “registrar rol”.

b.2. Programación

Tarea 01.- escribir test rol_spec.rb

```
rol_spec.rb ✕
1   .join( Rails.root.join('spec', 'helpers') )
2
3 describe Rol do
4   before(:each) do
5     setup_test_environment
6     @rol = Rol.new(rol_name => 'rol')
7   end
8
9
10
11   it 'habla por sí mismo rol query' do
12     test_create(@rol)
13     test_query.size.should == 1
14     query.errors.should == {}
15
16
17   it 'habla por sí mismo rol create' do
18     new_rol_id = test_create(@rol)
19     new_rol_id.should_not be_nil
20     create_errors.should == {}
21     rol[new_rol_id].should == @rol
22
23
24   it 'habla por sí mismo rol update' do
25     rol_id = test_create(@rol)
26     rol.should == {rol_id => @rol}
27     test_update({rol_id => {rol_name => 'rol update'}})
28     update_errors.should == {}
29     test_query[rol_id].should == @rol
30
31
32
33   it 'habla por sí mismo rol delete' do
34     rol_id = test_create(@rol)
35     rol.should == {rol_id => @rol}
36     test_delete(rol_id => @rol)
37     delete_errors.should == {}
38
39
40
41 end
```

Figura Nº 4.91. Escribir test rol_spec.rb

Tarea 02.- escribir controlador fuente rol.rb

```
rol.rb ✕
1   def initialize
2     @source_adapter = RolSourceAdapter.new
3
4     @rol = RolSourceAdapter.new
5     initialize(source)
6   end
7
8   def initialize(source)
9     @source_adapter = RolSourceAdapter.new
10    @source_adapter = RolSourceAdapter.new
11
12    query(params)
13    parsed = parse(response.body)
14    @result = []
15    parsed.each { |item|
16      @result[item['id'].to_s] = item['rol']
17    }
18
19    create(create_hash)
20    res = post(@base, {rol => create_hash})
21    parse(res)
22  end
23
24  def update(update_hash)
25    obj_id = update_hash['id']
26    update_hash.delete('id')
27    put(@base, {rol => update_hash})
28
29  end
30
31  def delete(delete_hash)
32    delete(@base, {rol => delete_hash})
33  end
34
35  def login
36    # ...
37  end
```

Figura Nº 4.92. Escribir controlador fuente rol.rb

Tarea 03.- escribir modelo rol.rb

rhogen model rol denominación

```
rol.rb ✕
1 # frozen_string_literal: true
2 # rubocop:disable Metrics/ClassLength
3 class Rol
4   attr_accessible :rol
5
6   # Validar que el rol sea un entero positivo y diferente de 0
7   enable :strict
8
9   # Validar que el rol sea un entero positivo
10 end
```

Figura Nº 4.93. Clase rol.rb

Tarea 04.- escribir controlador rol_controller.rb

```
rol_controller.rb ✕
1 include Rho::RhoController
2 include Helpers::BrowserHelper
3
4 class RolController < Rho::RhoController
5   attr_accessible :rol
6
7   def index
8     @rols = find(:all)
9     @rols = @rols.sort_by { |rol| rol.rol }
10    @rols = @rols.reverse
11    @rols
12  end
13
14  def show
15    @rol = find(@params[:id])
16    @rol = Rol.find(@rol)
17    respond_to do |format|
18      format.html { render :show }
19      format.json { render :show }
20    end
21  end
22
23  def new
24    @rol = Rol.new
25    @rol = Rol.new
26    respond_to do |format|
27      format.html { render :new }
28      format.json { render :new }
29    end
30  end
31
32  def edit
33    @rol = find(@params[:id])
34    @rol = Rol.find(@rol)
35    respond_to do |format|
36      format.html { render :edit }
37      format.json { render :edit }
38    end
39  end
40 end
```

Figura Nº 4.94. Controlador rol_controller.rb

Tarea 05.- escribir vista rol (edit.erb, index.erb, new.erb, show.erb)

<pre> edit.erb 1 <div data-role="page"> 2 3 <div data-role="header" data-position="inline"> 4 <h3Edit rol [denominacion] %> %> 5 <a href="#<%= @rol.denominacion %>" 6 data-icon="back" data-direction="reverse"> 7 Cancel 8 9 <a href="#<%= url_for :action => :delete, :id => @rol.object %>" 10 data-icon="delete" data-direction="reverse"> 11 Delete 12 13 </div> 14 15 <div data-role="content"> 16 <form method="POST" action="#<%= url_for :action => :update %>" 17 <input type="hidden" name="id" value="#<%= @rol.object %>" %> 18 19 <div data-role="fieldcontain"> 20 <input type="text" id="rol[denominacion]" 21 value="#<%= @rol.denominacion %>" 22 placeholder="denominacion" %> %> 23 24 <input type="submit" value="Update" %> 25 </div> 26 </div> 27 </div> </pre>	<pre> index.erb 1 <div data-role="page"> 2 3 <div data-role="header" data-position="inline"> 4 <h3Roles %> %> 5 <a href="#<%= @config.start_path %>" 6 class="ui-btn-left" data-direction="reverse" 7 data-ajax="false" if is_obj %> 8 Home 9 10 <a href="#<%= url_for :action => :new %>" 11 class="ui-btn-right" data-direction="reverse"> 12 New 13 14 </div> 15 <div data-role="content"> 16 <div data-role="listview"> 17 <% @rols.each %> <rol %> 18 19 <div> 20 <a href="#<%= url_for :action => :show, :id => @rol.object %>" 21 class="rol.denominacion" %> 22 23 <input type="text" value="#<%= @rol.denominacion %>" %> 24 25 </div> 26 </div> 27 </div> </pre>
---	--

Figura Nº 4.95. Vista rol edit.erb e index.erb.

<pre> new.erb 1 <div data-role="page"> 2 3 <div data-role="header" data-position="inline"> 4 <h3New rol [rol.title] %> %> 5 <a href="#<%= url_for :action => :index %>" 6 class="ui-btn-left" data-icon="back" 7 data-direction="reverse"> 8 Cancel 9 10 </div> 11 <div data-role="content"> 12 <form method="POST" action="#<%= url_for :action => :create %>" 13 <input type="hidden" name="id" value="#<%= @rol.object %>" %> 14 15 <div data-role="fieldcontain"> 16 <input type="text" id="rol[denominacion]" 17 value="#<%= @rol.denominacion %>" 18 placeholder="denominacion" %> %> 19 20 <input type="submit" value="Create" %> 21 </div> 22 </div> 23 </div> 24 25 </div> </pre>	<pre> show.erb 1 <div data-role="page" data-add-back-btn="false"> 2 3 <div data-role="header" data-position="inline"> 4 <h3rol [rol.denominacion] %> %> 5 <a href="#<%= url_for :action => :index %>" 6 class="ui-btn-left" data-direction="reverse"> 7 Back 8 9 <a href="#<%= url_for :action => :edit, :id => @rol.object %>" 10 class="ui-btn-right" data-direction="reverse"> 11 Edit 12 13 </div> 14 <div data-role="content"> 15 <div data-role="listview"> 16 <div> 17 <input type="text" value="#<%= @rol.denominacion %>" %> 18 19 <input type="text" value="#<%= @rol.denominacion %>" %> 20 21 </div> 22 </div> 23 </div> 24 </div> </pre>
--	---

Figura Nº 4.96. Vista rol new.erb y show.erb.

c. REVIEW MEETING

c.1. Verificar el cumplimiento del product backlog comprometido

H.U. N°	Criterio de aceptación	Estado
9	Cuando ingrese el ámbito, entonces debería poder utilizar entre: 1.Huamanga, 2.Huanta.	OK
9	Cuando ingrese categoría, entonces debería poder seleccionar entre: 1.domestico, 2.comercial, 3.industrial, 4.estatal, 5.social, 6.mixto.	OK
9	Cuando ingrese cantidad de instalaciones, entonces debería validar que sea número y menor de 10.	OK
9	cuando ingrese el tipo de uso, entonces debería poder relacionar entre: 1.empresa, 2.vivienda, 3.restaurant, 4.hotel, 5.colegio, 6.consultoria, 7.cuartel, 8.comisaria, 9.templos, 10.mercados, etc.	OK
9	Cuando ingrese número de personas, entonces debería validar un número.	OK
9	Cuando seleccione agregar unidad de uso, entonces debería poder agregar a la lista de unidad de uso.	OK
10	Cuando ingrese mi PASSWORD correcto, entonces debiera mostrar la página principal.	OK
10	Cuando ingrese mi cuenta y mi PASSWORD incorrecto, entonces debiera mostrar mensaje de error.	OK
11	Cuando ingrese el nombre, entonces debe validar que sean caracteres.	OK
11	Cuando ingrese el apellido, entonces debiera validar que sean caracteres.	OK
11	Cuando ingrese el DNI, entonces debiera validar que sean números y solo de 8digitos.	OK
11	Debería poder ingresar la dirección.	OK

11	Cuando seleccione el sexo, entonces debiera poder seleccionar “masculino” o “femenino”	OK
12	Cuando defina el rol del empleado, entonces debería poder seleccionar administrador, prediador, supervisor.	OK
12	Cuando ingrese la denominación del usuario, entonces debería verificar si es único y mostrar error si ya existe la denominación del usuario.	OK
12	Cuando yo ingrese el PASSWORD, entonces debería verificar que tenga caracteres y número.	OK

Tabla Nº 4.20. Review meeting.

d. RETROSPECTIVA

d.1. Mejorar los procesos de Scrum

¿QUÉ FUNCIONO BIEN?	CONFORME
Planinng meeting	OK
Relase plan	OK
Sprint	OK
Dayli meeting	OK
Review meeting	OK
Retrospectiva	OK
¿QUÉ FUNCIONO MAL?	CONFORME
En el desarrollo no se pudo la automatización de test de cliente	FAIL

Tabla Nº 4.21. Verificación de los procesos de Scrum.

CAPITULO V

CONCLUSIONES Y RECOMENDACIONES

5.1. CONCLUSION

- a.** Para la localización de la conexión de un usuario de agua potable se empleo el GPS o A-GPS del teléfono inteligente, las clases de Geolocalización de rhodes y también la asignación a los predios con coordenadas UTM para su identificación, que realiza la aplicación web que se implementó.

- b.** Se puede registrar las características de conexión para un usuario de agua potable a través del manejo de información de los clientes, predios, estado de conexión, medidor y conexión agua, que realiza la aplicación web que se implementó.

- c.** Se puede identificar el tipo de servicio de los predios a través del manejo de información de la unidad de uso, que realiza la aplicación web que se implementó.

5.2. RECOMENDACIÓN

- a.** Evaluar el cambio de la metodología actual de actualización de catastro de usuario de agua potable con la propuesta. Se realizaría la actualización en la lectura de los medidores con el software implementado.

- b.** Realizar el diagrama de secuencia utilizando la técnica Round-trip engineering para entender las dependencias y las responsabilidades de las clases.

- c.** Para la instalación del ambiente de desarrollo, sobre todo el repositorio y centro de integración emplear <https://github.com/> y <http://hudson-ci.org/> que son herramientas online y no requiere tiempo en su instalación y configuración.

- d.** Para adaptar y poner en practica los principios de Scrum es mejor simular y jugar con: Ball Point Game, Spec Writer, Visual Story Mapping, Personas + UX, Lego ágil, Spaguetti.

BIBLIOGRAFIA

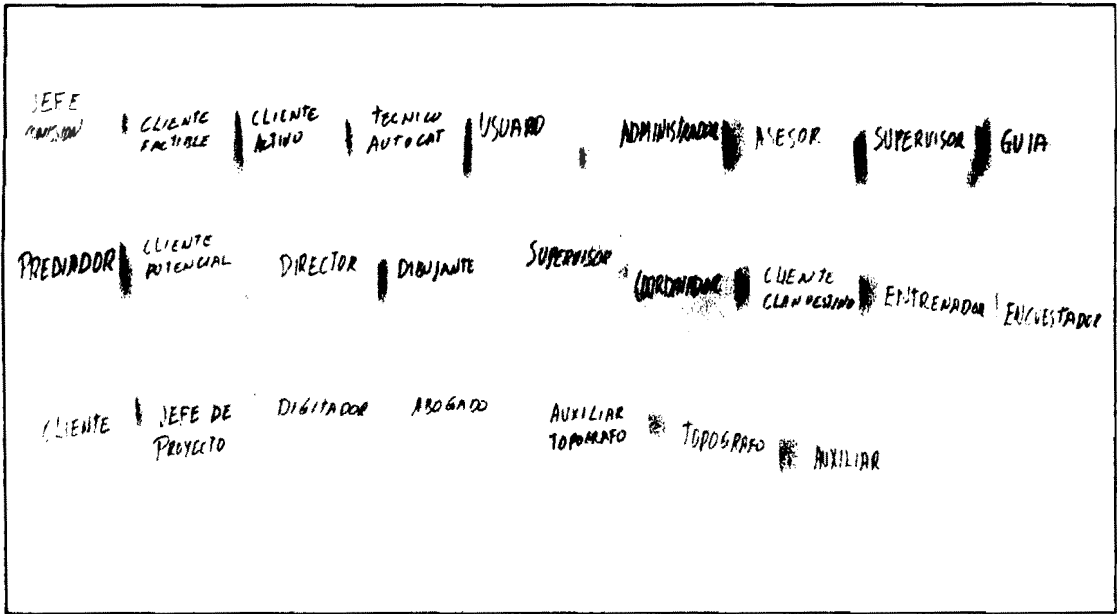
1. Cohn, M. (2004). User Stories Applied: For Agile Software Development. Boston: Pearson Education, Inc.
2. Diego Germán de la Riva Legajo (2007). Tesis de Grado "Aplicaciones Web para celulares" pag. 6,10
3. David Chelimsky (2010). "The RSpec Book, Behaviour-Driven Development with RSpec, Cucumber, and Friends" Pag. 20,28
4. EPSASA. (2009). Plan Operativo Anual – 2010. Ayacucho.
5. EPSASA. (2009). Planeamiento Estratégico 2010 –2014. Ayacucho.
6. EPSASA (2010). “Formulación del plan estratégico 2010 – 2014 y plan operativo anual (POA) 2010 de EPSASA”.
7. Estrems Amestoy, Manuel (2011). Tesis profesional "aplicación de instrumentos móviles con tecnología Wifi para la mejora del aprendizaje en clase" pag. 8,85
8. Gianfelici, Esteban (2011).Tesis profesional "sistema de información geográfica MAPSERVER" pag. 8,102
9. Gallarday, L., & Ramírez, J. (15 de Enero de 2012). <http://www.kamlov.site90.net>.
Obtenido de <http://www.kamlov.site90.net>.
10. Héctor Zárate Rea (2011). Tesis profesional “Guía de Referencia para el Desarrollo y Distribución de Software en Plataformas Móviles” pag. X,X,
11. Jim Shore and Shane Warden (2007). "The Art of Agile Development" pag. 285,270,290

12. José Cegarra Sánchez (2010). "Metodología de la investigación científica y tecnológica", Pág. 92.
13. Keith, C. (2010). Agile Game Development with Scrum. Boston: Pearson Education, Inc.
14. Palacio, J. (2007). Flexibilidad con Scrum. <http://www.safecreative.org>: safeCreative.
15. Roberto Hernández, Carlos Fernández & María del Pilar Baptista (2010). "Metodología de la investigación", Pag. 120, 121.
16. Servicio Universitario Mundial del Canadá - SUM Canadá. (2007). CATASTRO DE USUARIOS DE AGUA POTABLE Y DESAGÜE. Lima: Juan Escate Cavero.
17. Santana J. Camargo S. (2011). Tesis doctoral "Estudio y diseño de metodología con Técnicas GPS para la actualización de la cartografía catastral del Municipio Palavecino (Venezuela)" pag. 7,45
18. The RSpec Book: Behaviour-Driven Development with RSpec, Cucumber, and Friends.
19. EPSASA (2010). "Formulación del plan estratégico 2010 – 2014 y plan operativo anual (POA) 2010 de EPSASA"

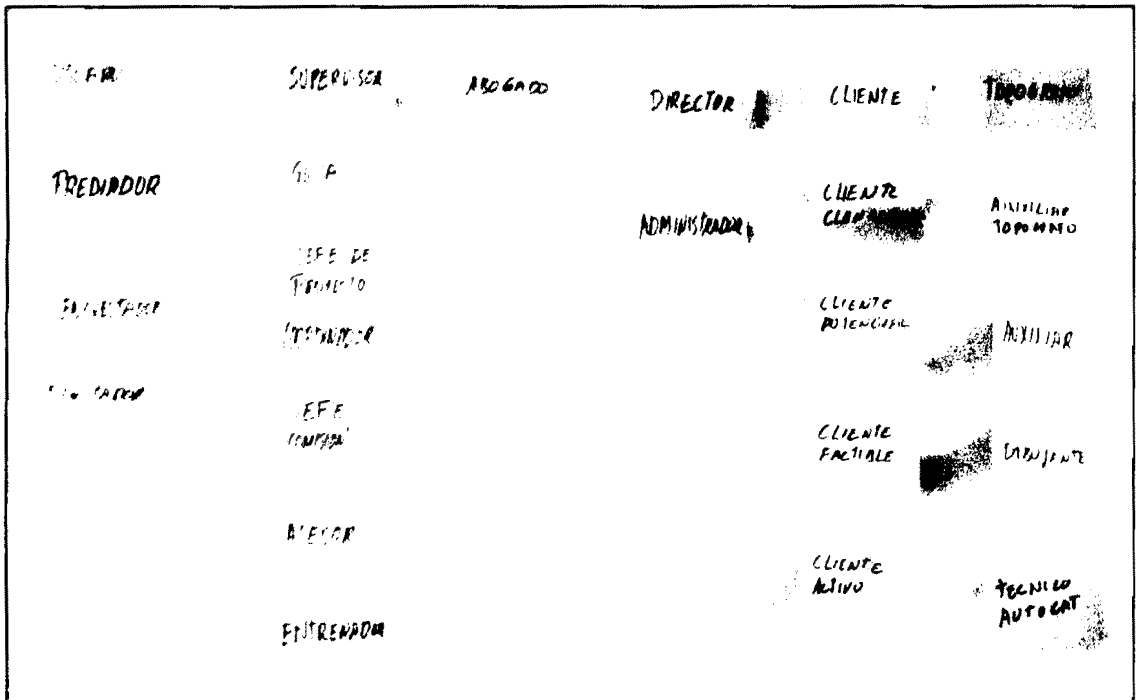
ANEXO A

MODELADO DE ROLES

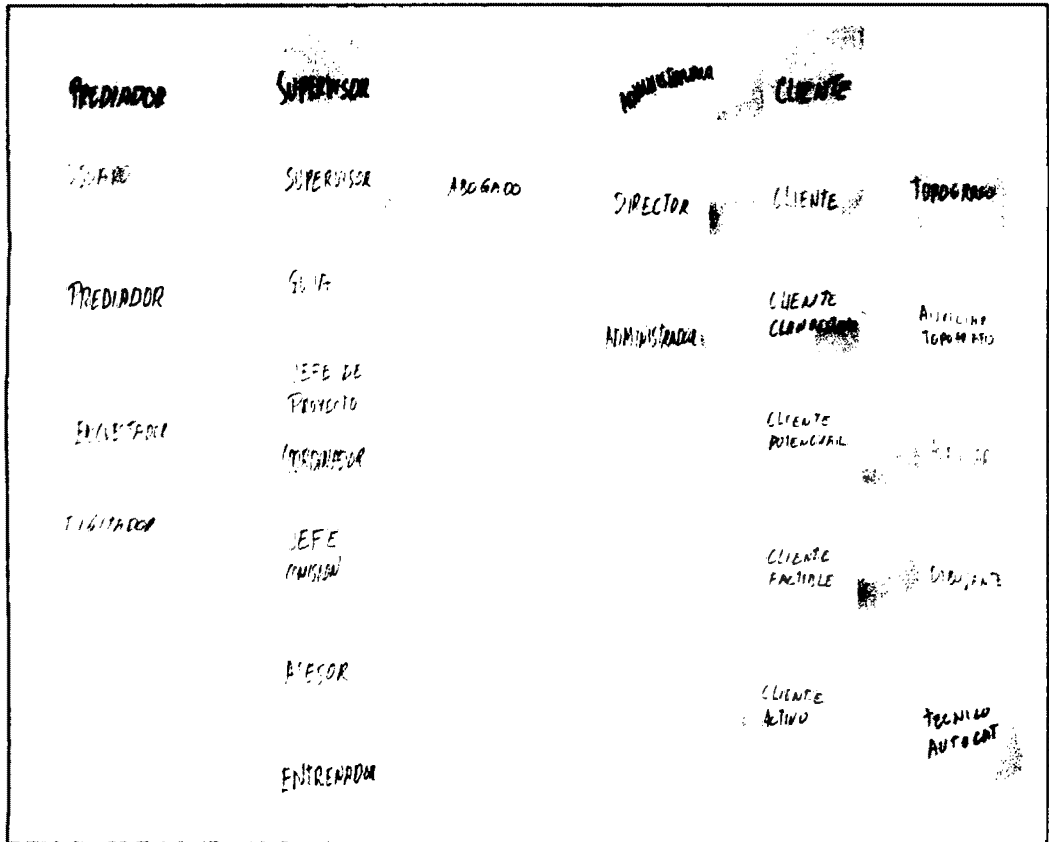
1. Identificación



2. Organización

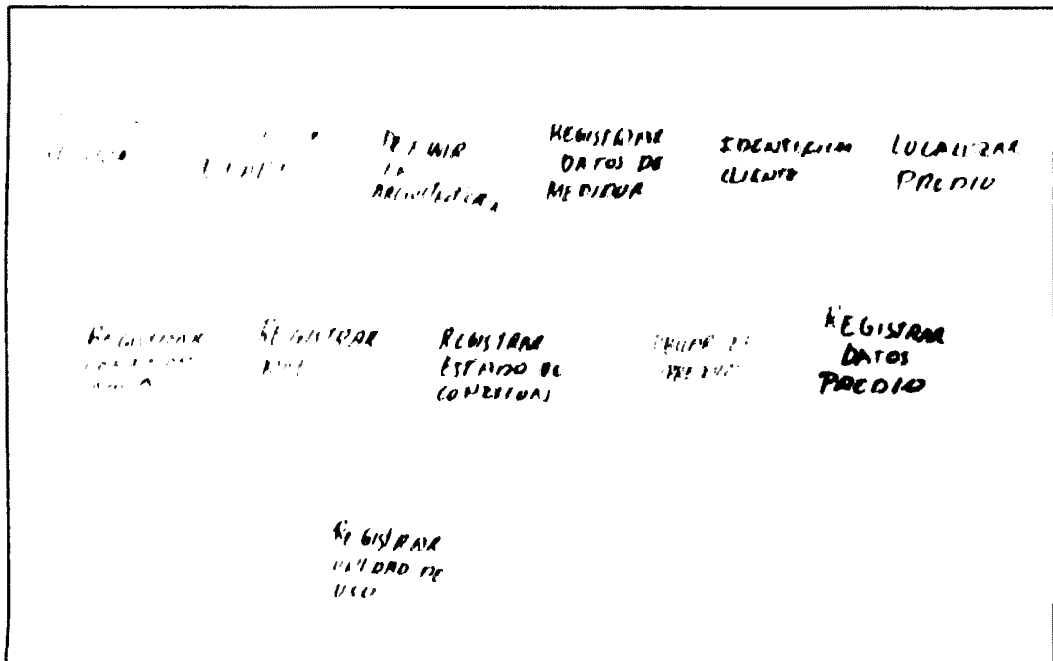


3. Consolidación y refinamiento

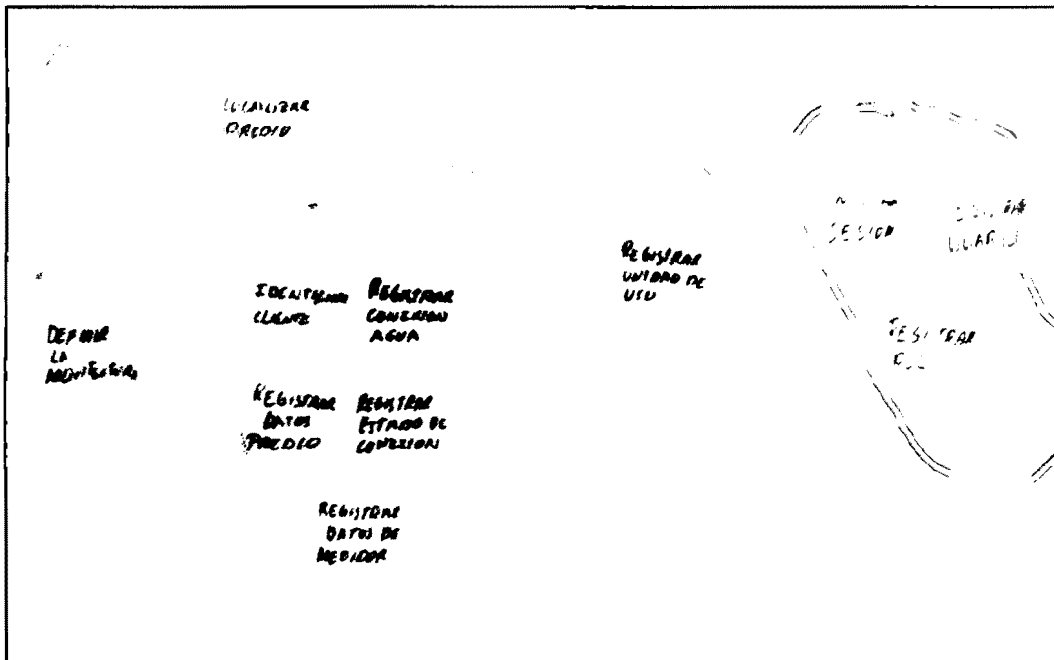


ANEXOB

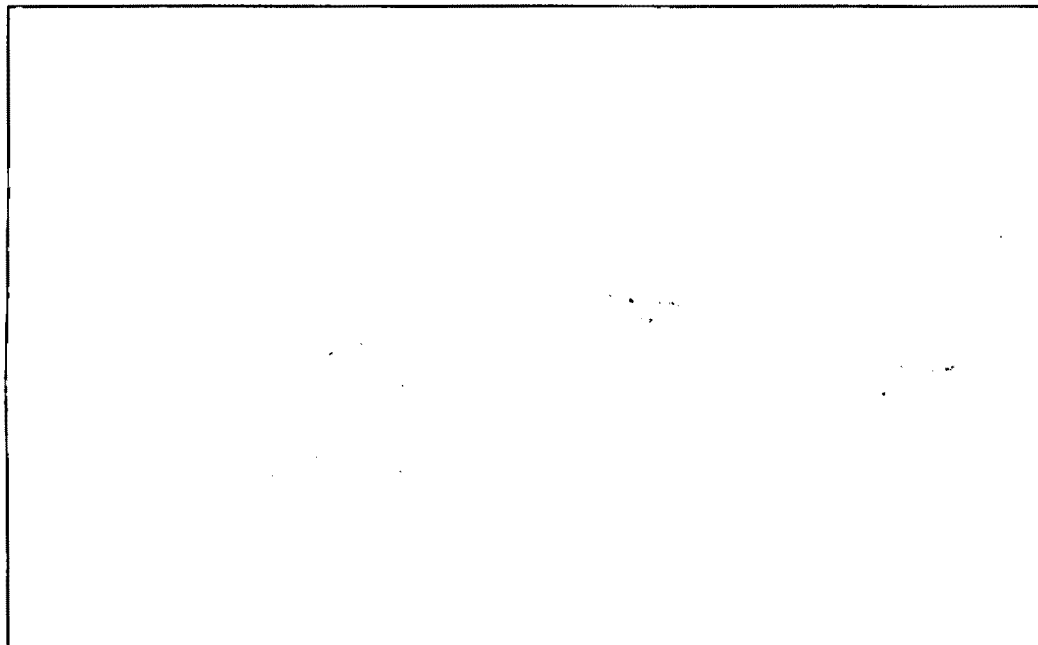
1. La identificación del requerimiento se obtuvo de las entrevistas que se hizo al product owner en la reunión de planificación que realizamos.



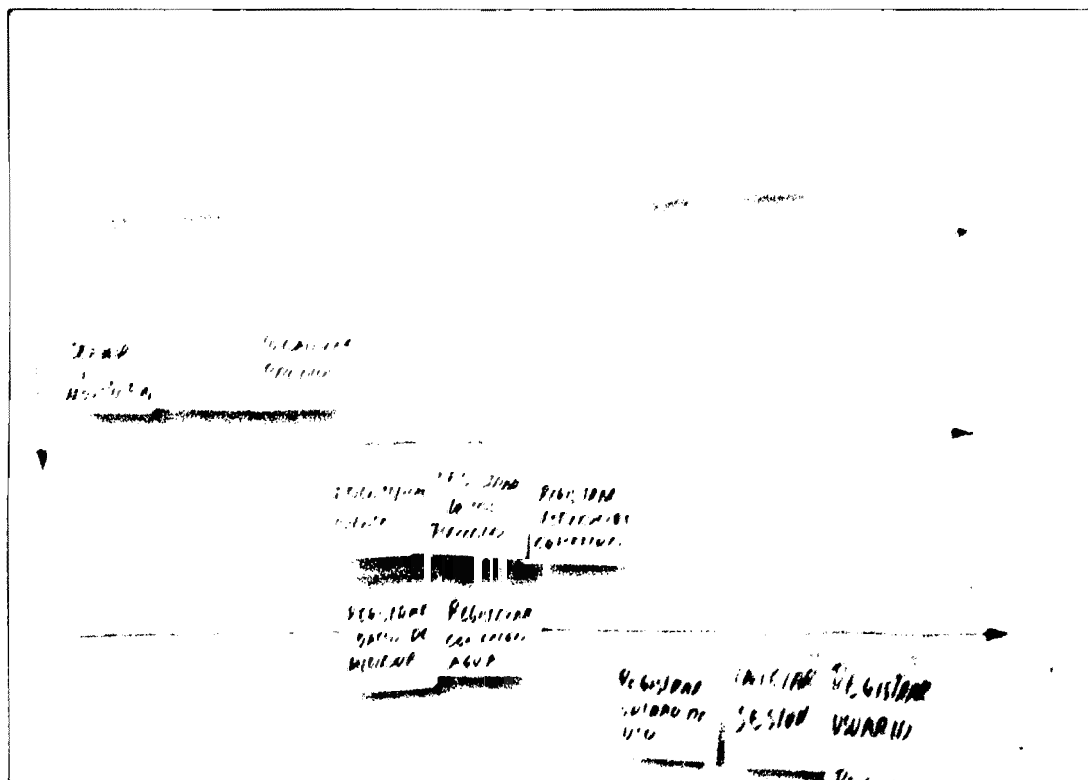
2. Creando islas para organizar conceptos.



3. Organizando por procesos.

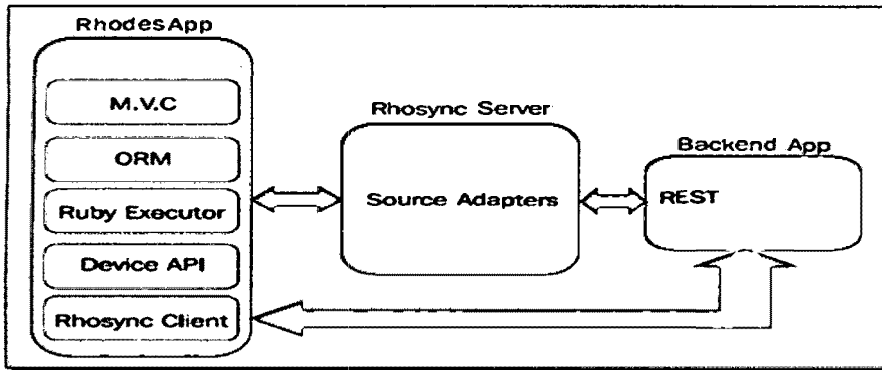


4. Priorizando los requerimientos utilizando la técnica Hi-Low Story Showdown y el visual story mapping.



ANEXOC

1. Representación grafica de la arquitectura.



2. Configuración de Rhodes y Rhosync en el archivo build.yml.

```
1 android:
2   version: 4.0.3
3   mapping: yes
4   apikey: 0Eii6t8-Y5e32-UaWLUyhQ1sk27uGl-HFtttXQA
5   emulator: dev-android
6   capabilities:
7     - network_state
8   capabilities:
9     - gps
10  extensions: ["fileutils","fileutils","mspec"]
11
```

3. creación del emulador android

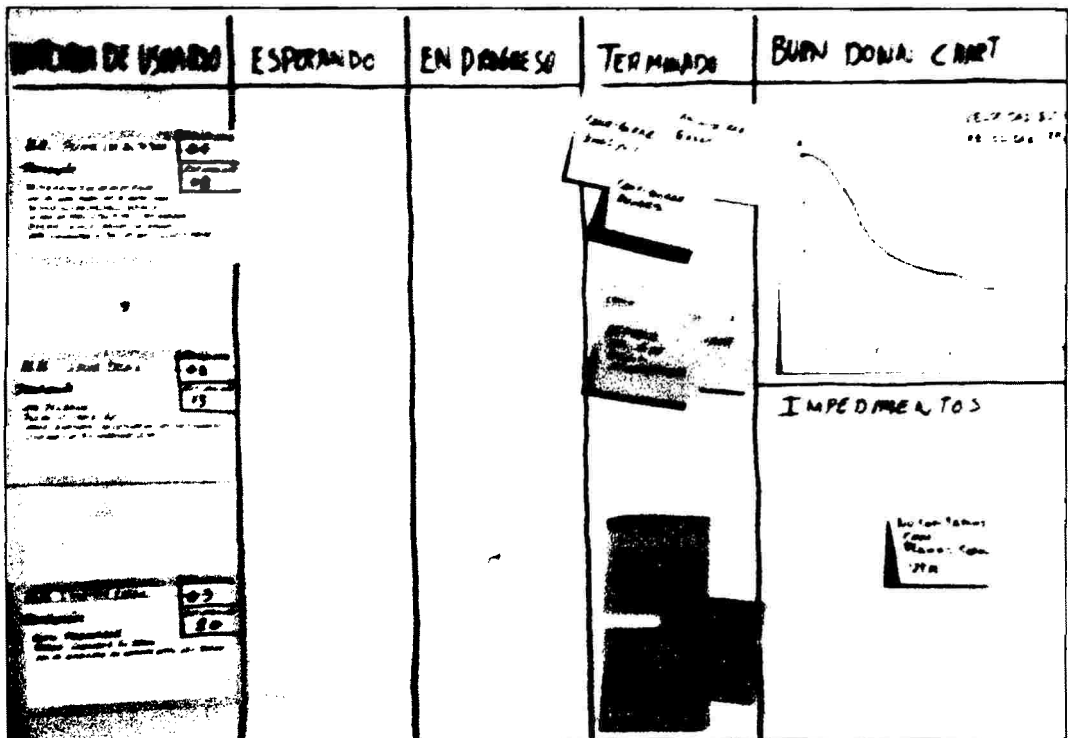
Antes de crear instalar ndk y sdk de android

```
#!/mksdcard 4096M SDCard
#!/android create avd -n emu-android -t 3 -b armeabi-v7a
#!/opt/sdk/tools/emulator @ emu-android -sdcard /opt/sdk/tools/SDCard
```

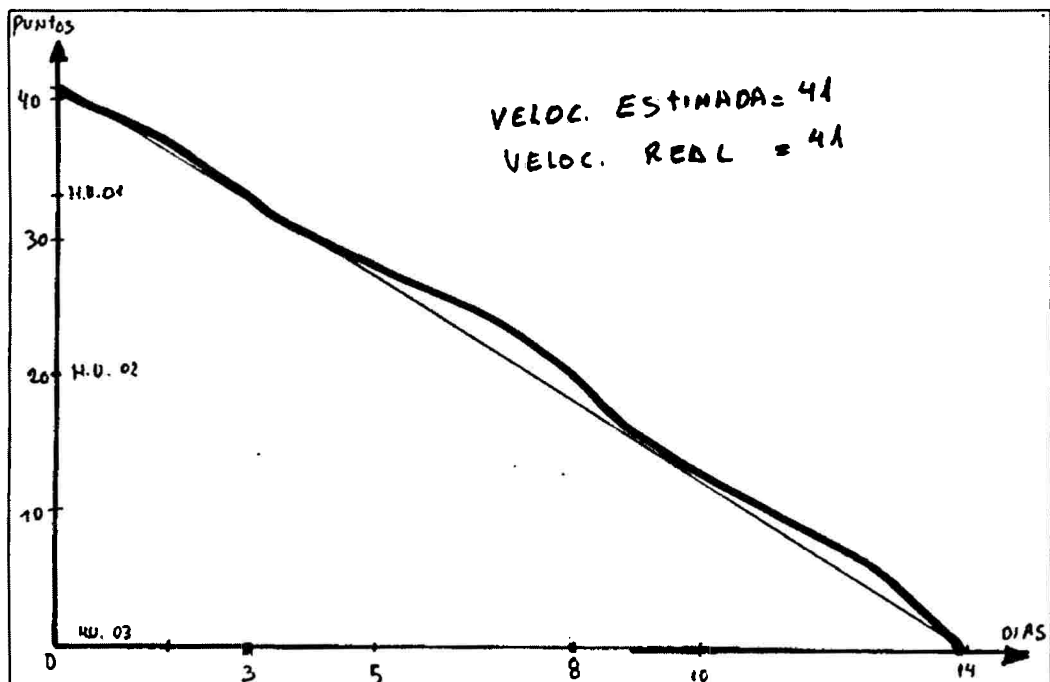
4. Instalación de Rhodes y Rhosync

```
# sudo gem install rhodes
#sudo gem install rhoconnect
```


3. Imagen del taskboard en terminado del Sprint 1



4. Born down chart del Sprint



ANEXO E

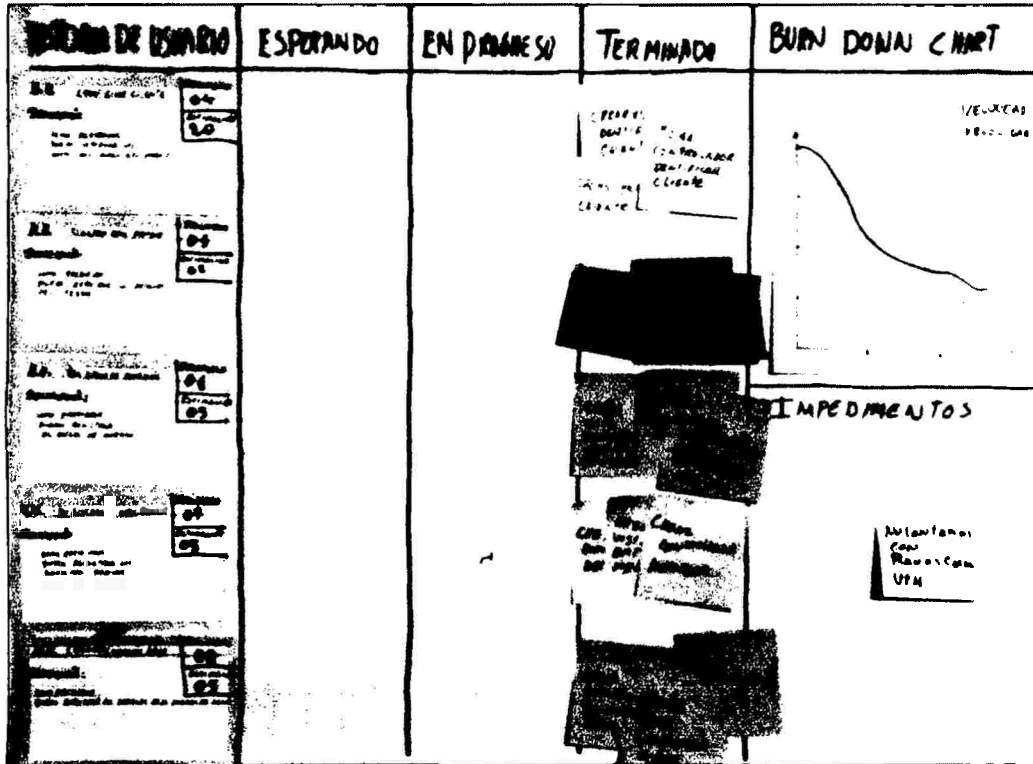
1. Imagen del taskboard capturado al comienzo del Sprint 2

Historia de Usuario	Esperando	En Progreso	Terminado	Burn Down Chart
<p>U1. Crear perfil de usuario</p> <p>U2. Autenticación de usuario</p> <p>U3. Gestión de contraseñas</p> <p>U4. Perfil de usuario</p> <p>U5. Configuración de perfil</p> <p>U6. Gestión de roles</p> <p>U7. Gestión de permisos</p> <p>U8. Gestión de usuarios</p>	<p>U1. Crear perfil de usuario</p> <p>U2. Autenticación de usuario</p> <p>U3. Gestión de contraseñas</p> <p>U4. Perfil de usuario</p> <p>U5. Configuración de perfil</p> <p>U6. Gestión de roles</p> <p>U7. Gestión de permisos</p> <p>U8. Gestión de usuarios</p>			<p>Imágenes de un gráfico de Burn Down Chart y un botón de imprimir.</p>

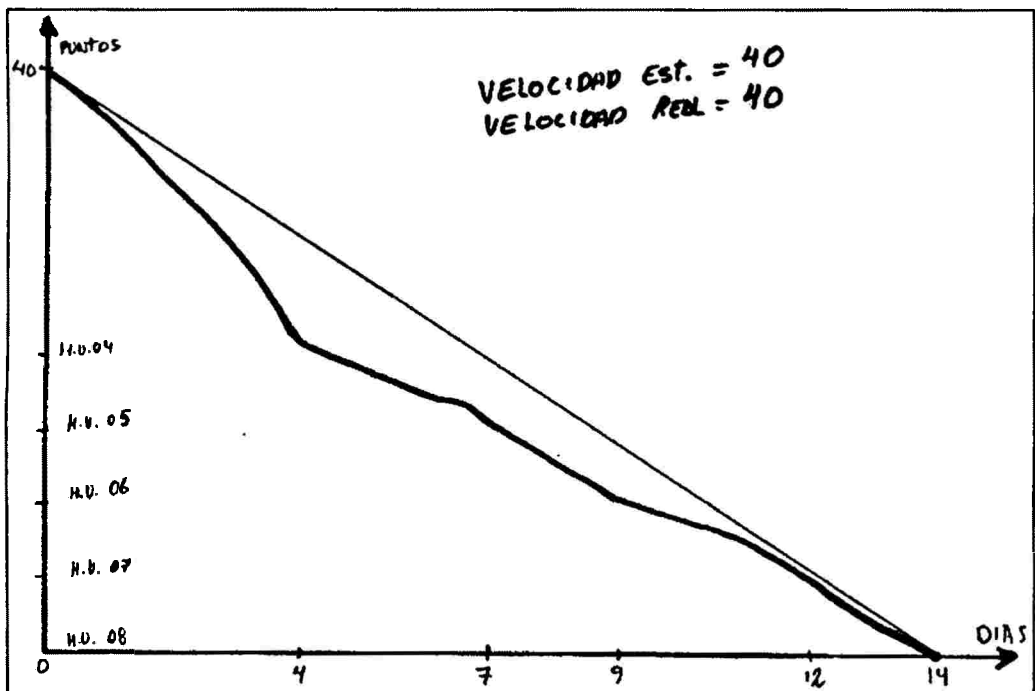
2. Imagen del taskboard en progreso del Sprint 2

Historia de Usuario	Esperando	En Progreso	Terminado	Burn Down Chart
<p>U1. Crear perfil de usuario</p> <p>U2. Autenticación de usuario</p> <p>U3. Gestión de contraseñas</p> <p>U4. Perfil de usuario</p> <p>U5. Configuración de perfil</p> <p>U6. Gestión de roles</p> <p>U7. Gestión de permisos</p> <p>U8. Gestión de usuarios</p>	<p>U1. Crear perfil de usuario</p> <p>U2. Autenticación de usuario</p> <p>U3. Gestión de contraseñas</p> <p>U4. Perfil de usuario</p> <p>U5. Configuración de perfil</p> <p>U6. Gestión de roles</p> <p>U7. Gestión de permisos</p> <p>U8. Gestión de usuarios</p>	<p>U1. Crear perfil de usuario</p> <p>U2. Autenticación de usuario</p> <p>U3. Gestión de contraseñas</p> <p>U4. Perfil de usuario</p> <p>U5. Configuración de perfil</p> <p>U6. Gestión de roles</p> <p>U7. Gestión de permisos</p> <p>U8. Gestión de usuarios</p>	<p>U1. Crear perfil de usuario</p> <p>U2. Autenticación de usuario</p> <p>U3. Gestión de contraseñas</p> <p>U4. Perfil de usuario</p> <p>U5. Configuración de perfil</p> <p>U6. Gestión de roles</p> <p>U7. Gestión de permisos</p> <p>U8. Gestión de usuarios</p>	<p>Imágenes de un gráfico de Burn Down Chart y un botón de imprimir.</p>

3. Imagen del taskboard terminado del Sprint 2



4. Born down chart del Sprint 2



ANEXOF

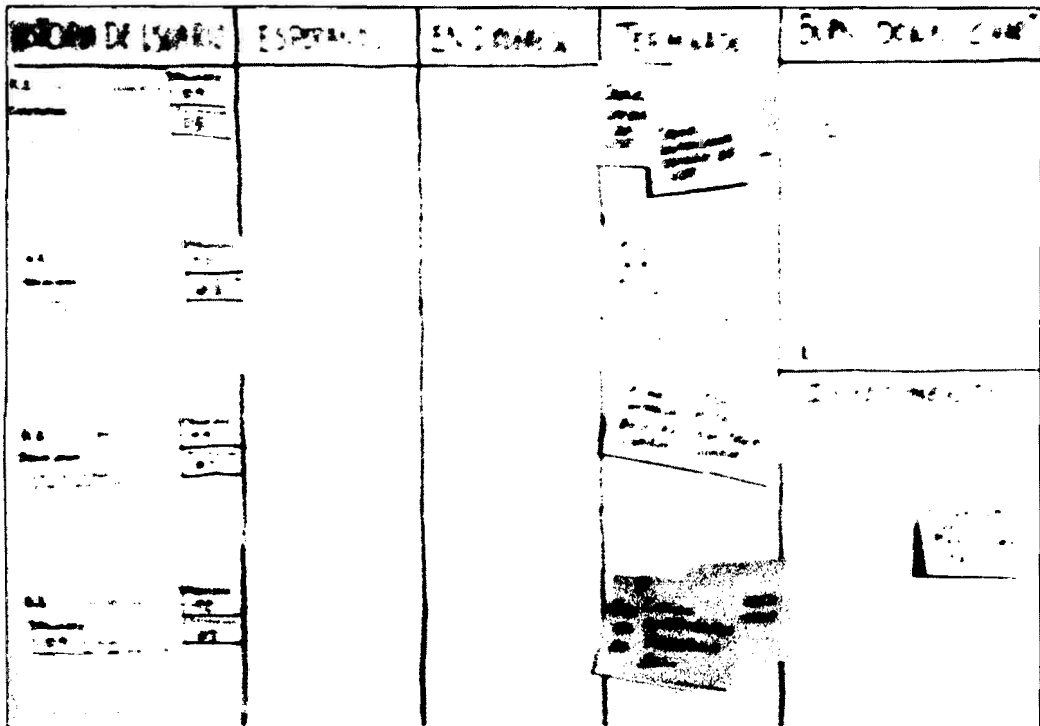
1. Imagen del taskboard capturado al empezo del Sprint 3

HISTORIA DE USUARIO	ESPERANDO	EN DESARROLLO	TERMINADO	EN PRÓXIMA VERSIÓN
<p>H.U. 01 Desarrollar el módulo de inicio de sesión.</p> <p>H.U. 02 Implementar la validación de credenciales.</p> <p>H.U. 03 Crear el flujo de registro de usuarios.</p> <p>H.U. 04 Diseñar la interfaz de usuario para el registro.</p> <p>H.U. 05 Desarrollar la lógica de negocio para el registro.</p>	<p>01 Implementar la validación de credenciales.</p> <p>02 Crear el flujo de registro de usuarios.</p> <p>03 Diseñar la interfaz de usuario para el registro.</p> <p>04 Desarrollar la lógica de negocio para el registro.</p>			

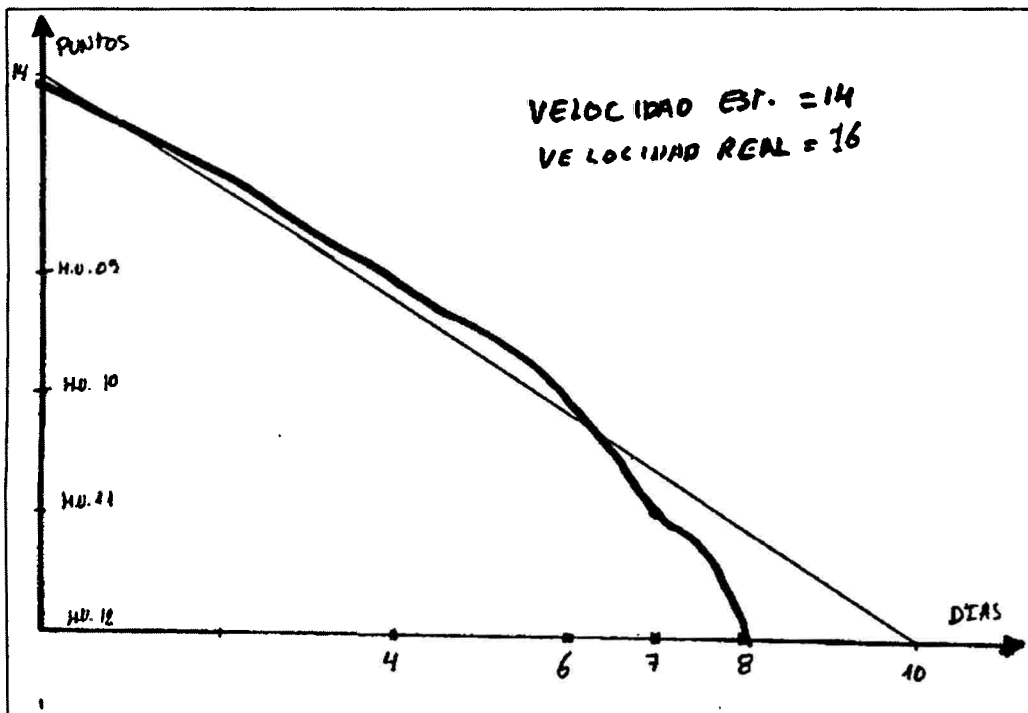
2. Imagen del taskboard en progreso del Sprint 3

HISTORIA DE USUARIO	ESPERANDO	EN DESARROLLO	TERMINADO	EN PRÓXIMA VERSIÓN
<p>H.U. 01 Desarrollar el módulo de inicio de sesión.</p> <p>H.U. 02 Implementar la validación de credenciales.</p> <p>H.U. 03 Crear el flujo de registro de usuarios.</p> <p>H.U. 04 Diseñar la interfaz de usuario para el registro.</p> <p>H.U. 05 Desarrollar la lógica de negocio para el registro.</p>	<p>01 Implementar la validación de credenciales.</p> <p>02 Crear el flujo de registro de usuarios.</p> <p>03 Diseñar la interfaz de usuario para el registro.</p> <p>04 Desarrollar la lógica de negocio para el registro.</p>	<p>01 Implementar la validación de credenciales.</p> <p>02 Crear el flujo de registro de usuarios.</p> <p>03 Diseñar la interfaz de usuario para el registro.</p> <p>04 Desarrollar la lógica de negocio para el registro.</p>	<p>01 Implementar la validación de credenciales.</p> <p>02 Crear el flujo de registro de usuarios.</p> <p>03 Diseñar la interfaz de usuario para el registro.</p> <p>04 Desarrollar la lógica de negocio para el registro.</p>	

3. Imagen del taskboard en terminado del Sprint 3



4. Born down chart del Sprint 3



ANEXOG

1. PLANTILLA DE HISTORIA DE USUARIO:

Esta plantilla tiene dos partes: la narración y el criterio de aceptación que fue tomada de Mike Cohn y de Kent Beck.

HISTORIA DE USUARIO	
Título:	Prioridad
	Estimación
Descripción: As a <user role>, I want <goal> [so that <reason>].	
Criterio de aceptación: "When (scenari0 X), then (scenari0 Y)." 	

2. PLANTILLA DE ESCRITURA DE LAS TAREAS:

Título:

Etiqueta	Propietario	Estado (esperando, en progreso y terminado)

ANEXO H

1. Historia de usuario 01.- "definir la arquitectura", para realizar esta historia de usuario optaremos la técnica de arquitectura incremental.

H.U. DEFINIR ARQUITECTURA	PRIORIDAD 01
DESCRIPCIÓN: DEFINIR LA CONFIGURACIÓN DE RINDES CON LAS CAPAS MODEM - WTA - CONTROLADOR. DEFINIR LA CONFIGURACIÓN DE RMM LA CAPA DE PERSISTENCIA CON ORM EMBEBIDO. DEFINIR LA CONFIGURACIÓN DE RHOSSync PARA SINCRONIZAR LA BASE DE DATOS CLIENTE - SERVIDOR.	ESTIMACIÓN 08

2. Historia de usuario 02.- "obtener localización".

H. U. OBTENER LOCALIZACIÓN	PRIORIDAD 02
DESCRIPCIÓN: COMO PRE DIADOR PUEDO UTILIZAR EL GPS CON LA FUNCIÓN DE OBTENER LA LATITUD Y LONGITUD Y CONVERTIR EN COORDENADAS UTM	ESTIMACIÓN 13

(CRITERIO DE ACEPTACIÓN): ②. Cuando seleccione Ubicar GPS, entonces debería poder obtener la latitud y la longitud. ②. Cuando convierta a coordenadas UTM, entonces deberían calcular las coordenadas con la fórmula UTM (Latitud, longitud)

3. Historia de usuario 03.- "Localizar predio".

H.U.	LOCALIZAR PREDIO	ESTIMACION
		03
DESCRIPCION:		ESTIMACION
COMO PROPIETARIO		20
PUEDO LOCALIZAR EL PREDIO		
CON LA FINALIDAD DE OBTENER DATOS DEL PREDIO		

CRITERIO DE ACEPTACION:

- ① Cuando Compran UTM, entonces debería comprar las coordenadas UTM de catastro con las coordenadas UTM calculado.
- ② Cuando localice el predio, entonces debería ver: provincia, distrito, Sector, Ruta, manzana y local.

4. Historia de usuario 04.- "identifica cliente"

H.U.	IDENTIFICAR CLIENTE	ESTIMACION
		04
DESCRIPCION:		ESTIMACION
COMO PROPIETARIO		20
PUEDO VERIFICAR LOS		
DATOS DEL PUEBLO DEL PREDIO		

CRITERIO DE ACEPTACION:

- ① Cuando ubique el predio, entonces debería poder consultar los datos (nombres y apellidos, DNI, RUC, telefono, direccion, No Dpto, Piso y poder modificar.

5. Historia de usuario 05.- "registrar datos predio"

<p>H.U. REGISTRA DATOS PREDIO</p> <p>DESCRIPCIÓN:</p> <p>Como PREDIO DE PUEDEN REGISTRAR LOS DATOS DEL PREDIO</p>	<p>05</p> <p>05</p>
---	---------------------

CRITERIO DE ACEPTACIÓN:

- ① Cuando ingrese el tipo de propiedad, entonces debería seleccionar entre: 1. particular, 2. mixto, 3. público municipal, 4. público nacional.
- ② Cuando ingrese el tipo de construcción, entonces debería poder seleccionar: 1. lote sin construcción, 2. casa, 3. edificio, 4. multifamiliar, 5. abandonado.

6. Historia de usuario 06.- "registrar estado de conexión"

<p>H.U. REG. ESTADO DE CONEXIÓN</p> <p>DESCRIPCIÓN:</p> <p>Como PUEDE REGISTRAR EL ESTADO DE CONEXIÓN</p>	<p>06</p> <p>05</p>
---	---------------------

CRITERIO DE ACEPTACIÓN:

- ① Cuando ingrese distribución de facturación, entonces debería poder seleccionar entre:
 - 1) Dirección inaneble
 - 2) Dirección de correspondencia.
 - 3) Banco / agencia
- ② Cuando ingrese estado de correspondencia, entonces debería poder seleccionar entre 1. Agua en uso, 2. Agua cortado, 3. factible, 4. potencial, 5. clandestino.

7. Historia de usuario 07.- "registrar datos medidor"

H.U.	REGISTRAR DATOS MEDIDOR	PROBLEMAS
DESCRIPCIÓN:		07
COMO PREVIADOR DEBERIA REGISTRAR LOS DATOS DEL MEDIDOR		ESTIMACION
		05

CRITERIO DE ACEPTACIÓN:

② Cuando ingrese el estado medidor, entonces debería poder seleccionar entre: 1. operativo, 2. inoperativo, 3. alterado, 4. otros.

① Cuando Defina medidor instalado, debería poder seleccionar "SI" o "NO".

8. Historia de usuario 08.- "registrar conexión agua"

H.U.	REGISTRAR CONEXION AGUA	PROBLEMAS
DESCRIPCIÓN:		08
COMO PREVIADOR DEBERIA REGISTRAR LOS DETALLES DE LA CONEXION DE AGUA		ESTIMACION
		05

CRITERIO DE ACEPTACIÓN:

③ Cuando ingrese localización de caja medidor, entonces debería poder seleccionar entre: 1. en vereda, 2. en pared, 3. interno, 4. pista, 5. no ubicado.

② Cuando ingrese material caja, entonces debería poder seleccionar entre: 1. conueta, 2. ladrillo, 3. sin caja, 4. otros.

9. Historia de usuario 09.- "registrar unidad de uso"

H.U. REGISTRAR UNIDAD DE USO	09
DESCRIPCIÓN: COMO ADMINISTRADOR PUEDE REGISTRAR LOS DATOS DE CADA UNIDAD DE USO	05

CRITERIO DE ACEPTACIÓN:

- ① Cuando ingrese el ambito, entonces debería poder seleccionar entre: 1. Huamanga 2. Huanta.
- ② Cuando ingrese categoría, entonces debería poder seleccionar entre: 1. domestico, 2. comunal, 3. industrial, 4. estatal, 5. social, 6. mixto.
- ③ Cuando ingrese cantidad de instalaciones, entonces debería validar que sea numero y menor de 10.

10. Historia de usuario 10.- "iniciar sesión"

H.U. INICIAR SESION	10
DESCRIPCIÓN: COMO USUARIO PUEDE INICIAR CON SU CUENTA Y PASSWORD	03

CRITERIOS DE ACEPTACIÓN:

- ①.- Cuando ingrese mi cuenta y password correcto, entonces debería mostrar la pagina principal.
- ②.- Cuando ingrese mi cuenta y mi password incorrecto, entonces debería mostrar mensaje de error.

11. Historia de usuario 11.- "registrar usuario"

H.U. REGISTRAR USUARIO.	PRIORIDAD 11
DESCRIPCIÓN: COMO ADMINISTRADOR PUEDE CREAR UN NUEVO USUARIO	ESTIMACIÓN 03

CRITERIO DE ACEPTACIÓN:

- ① Cuando ingrese el nombre entonces debe validar que sean caracteres.
- ② cuando ingrese el apellido, entonces debería validar que sean caracteres.
- ③ Cuando ingrese el DNI entonces debería validar que sean números y solo de 8 dígitos.

12. Historia de usuario 12.- "Registrar Rol"

H.U. REGISTRAR ROL	PRIORIDAD 12
DESCRIPCIÓN: COMO ADMINISTRADOR PUEDE ASIGNAR A CADA USUARIO SU ROL	ESTIMACIÓN 03

CRITERIOS DE ACEPTACIÓN:

- ① Cuando defina el rol del empleado, entonces debería poder seleccionar: Activo, precluido, supervisor.
- ②- Cuando ingrese la denominación de usuario, entonces debería verificar que es único y mostrar error si ya existe la denominación del usuario.