

UNIVERSIDAD NACIONAL DE SAN CRISTÓBAL DE HUAMANGA

FACULTAD DE INGENIERÍA DE MINAS, GEOLOGÍA Y CIVIL

ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS



TESIS:

**SISTEMA WEB PARA LA GESTIÓN DE RECURSOS EN LOS JUZGADOS
DE PAZ DEL DISTRITO JUDICIAL DE AYACUCHO, 2023**

Para optar el título profesional de:

INGENIERO DE SISTEMAS

PRESENTADA POR:

Bach. Max AUQUI BAUTISTA

ASESOR:

Dr. Ing. Efraín Elías PORRAS FLORES

AYACUCHO - PERÚ

2024

DEDICATORIA

A Dios, a mi madre y padre, Marina y Teodoro, por su incondicional apoyo y fuerza para seguir adelante. A mis hermanas Paola y Danna, por su constante apoyo. A mis hermanos Renzo y Roque, por su apoyo desinteresado.

AGRADECIMIENTO

A mis padres, por su apoyo incondicional y su confianza en mis sueños.

A mi familia, por su amor y apoyo constante.

A la Universidad San Cristóbal, por la oportunidad de formarme profesionalmente.

A la Corte Superior de Justicia de Ayacucho, por la oportunidad de desarrollarme profesionalmente.

A los docentes y mentores, por su guía y apoyo en mi aprendizaje.

Al Dr. Ing. Efraín Elías Porras Flores, por su amistad, colaboración y apoyo profesional.

A la Ing. Eliane Sammy Jjoyo Meneses por su apoyo constante durante el desarrollo de la investigación.

CAPÍTULO I		
INTRODUCCIÓN		
1.1.	PLANTEAMIENTO DEL PROBLEMA	13
1.2.	FORMULACIÓN DEL PROBLEMA	14
1.2.1.	PROBLEMA GENERAL	14
1.2.2.	PROBLEMAS ESPECÍFICOS	14
1.3.	JUSTIFICACIÓN DE LA INVESTIGACIÓN	14
1.4.	LIMITACIONES DE LA INVESTIGACIÓN	14
1.5.	OBJETIVOS	14
1.5.1.	OBJETIVO GENERAL	14
1.5.2.	OBJETIVOS ESPECÍFICOS	14
CAPÍTULO II		
MARCO TEÓRICO		
2.1.	ANTECEDENTES DE LA INVESTIGACIÓN	15
2.2.	MARCO CONCEPTUAL	15
2.2.1.	JUZGADO DE PAZ	15
2.2.2.	JUEZ DE PAZ	15
2.2.3.	RECURSO HUMANO	16
2.2.4.	RECURSOS FINANCIEROS	16
2.2.5.	RECURSO MATERIAL	17
2.2.6.	RECURSO TECNOLÓGICO	17
2.3.	MARCO REFERENCIAL	17
2.3.1.	SISTEMA WEB	17
2.3.2.	PROGRAMACIÓN EXTREMA (XP)	19
2.3.3.	ACTIVIDADES BÁSICAS DE XP	20
2.3.4.	VARIABLES DE CONTROL DE RECURSOS DE XP	20
2.3.5.	VALORES DE LA PROGRAMACIÓN EXTREMA	20
2.3.6.	PRINCIPIOS BÁSICOS DE XP	20
2.3.7.	PRÁCTICAS DE LA PROGRAMACIÓN EXTREMA	21
2.3.8.	ROLES EN EL PROCESO DE DESARROLLO DE XP	23
2.3.9.	CICLO DE VIDA IDEAL DE UN PROYECTO XP	24
2.3.10.	ARTEFACTOS DE LA PROGRAMACIÓN EXTREMA	30
2.3.11.	SISTEMA GESTOR DE BASE DE DATOS	34
2.3.12.	METODOLOGÍA ÁGIL	35
2.3.13.	LENGUAJE DE PROGRAMACIÓN MULTIPARADIGMA	35
2.3.14.	ARQUITECTURA DE SOFTWARE	36
CAPÍTULO III		
MATERIALES Y MÉTODOS		
3.1.	TIPO Y NIVEL DE LA INVESTIGACIÓN	37
3.1.1.	TIPO DE INVESTIGACIÓN	37
3.1.2.	NIVEL DE INVESTIGACIÓN	37
3.2.	DISEÑO DE LA INVESTIGACIÓN	37
3.3.	POBLACIÓN Y MUESTRA	38
3.3.1.	POBLACIÓN	38
3.3.2.	MUESTRA	38
3.4.	VARIABLES	38
3.4.1.	DEFINICIÓN CONCEPTUAL DE VARIABLES	38
3.4.2.	DEFINICIÓN OPERACIONAL DE VARIABLES	39
3.5.	OPERACIONALIZACIÓN DE VARIABLES	39

3.6.	TÉCNICAS E INSTRUMENTOS DE INVESTIGACIÓN	41
3.6.1.	TÉCNICAS	41
3.6.2.	INSTRUMENTOS	41
3.6.3.	VALIDEZ DEL INSTRUMENTO	41
3.6.4.	CONFIABILIDAD DEL INSTRUMENTO	41
3.7.	PROCEDIMIENTO	41
3.7.1.	TECNICA DE PROCESAMIENTO DE DATOS	41
3.7.2.	HERRAMIENTAS PARA EL PROCESAMIENTO DE DATOS	43
3.7.3.	TÉCNICAS DE ANÁLISIS DE DATOS	45
CAPÍTULO IV		
ANALISIS Y RESULTADOS DE LA INVESTIGACION		
4.1.	RESULTADOS	46
4.1.1.	ARTEFACTOS DEL SOFTWARE APLICANDO EL PROCESO XP	46
4.1.2.	INTERFAZ DE USUARIO Y CODIFICACIÓN	82
4.1.3.	TARJETAS CLASE RESPONSABILIDAD COLABORACIÓN (CRC)	93
4.2.	DISCUSIÓN	135
CAPÍTULO V		
CONCLUSIONES		
5.1.	CONCLUSIONES	136
CAPÍTULO VI		
RECOMENDACIONES		
6.1.	RECOMENDACIONES	137
6.2	REFERENCIAS BIBLIOGRÁFICAS	138
	LISTA DE ABREVIATURAS	140
	GLOSARIO	141
	ANEXOS	142

Lista de tablas

Tabla 1 <i>Prácticas de programación extrema</i>	22
Tabla 2 <i>Fase de exploración</i>	25
Tabla 3 <i>Fase de planeación</i>	26
Tabla 4 <i>Fase de iteración a la primera versión</i>	26
Tabla 5 <i>Fase de puesta en producción</i>	30
Tabla 6 <i>Plantilla historia de usuario</i>	31
Tabla 7 <i>Plantilla tareas de ingeniería</i>	32
Tabla 8 <i>Plantilla tarjeta CRC</i>	32
Tabla 9 <i>Plantilla pruebas de aceptación</i>	33
Tabla 10 <i>Operacionalización de variables</i>	39
Tabla 11 <i>Técnicas de procesamiento de datos</i>	41
Tabla 12 <i>Herramientas para el procesamiento de datos</i>	44
Tabla 13 <i>Historias de Usuario</i>	46
Tabla 14 <i>Plan de Alto Nivel</i>	48
Tabla 15 <i>Historia usuario: administrar usuarios del sistema.</i>	49
Tabla 16 <i>Historia usuario. Administrar Juez de Paz</i>	50
Tabla 17 <i>Historia usuario: Administrar Juzgado de Paz.</i>	51
Tabla 18 <i>Historia usuario. Administrar Recursos Materiales</i>	51
Tabla 19 <i>Historia usuario. Emitir Constancia de Transferencia Posesoria.</i>	52
Tabla 20 <i>Historia usuario. Asignar terna de jueces a juzgado.</i>	53
Tabla 21 <i>Historia usuario. Consultar Juzgados con ternas próximos a vencer</i>	53
Tabla 22 <i>Historia usuario. Iniciar sesión de jueces</i>	54
Tabla 23 <i>Historia usuario. Iniciar sesión de administrador.</i>	55
Tabla 24 <i>Historia usuario. Administrar listado de aranceles.</i>	55
Tabla 25 <i>Historia usuario. Consultar usuarios del sistema.</i>	56
Tabla 26 <i>Historia usuario. Consultar usuarios del sistema.</i>	56
Tabla 27 <i>Historia usuario. Consultar juzgados de paz.</i>	57
Tabla 28 <i>Historia usuario. Consultar recursos materiales</i>	58
Tabla 29 <i>Historia usuario. Consultar juzgados con y sin ternas asignadas.</i>	58
Tabla 30 <i>Historia usuario. Consultar cuadro de aranceles.</i>	59
Tabla 31 <i>Historia usuario. Administrar Recursos Tecnológicos</i>	59
Tabla 32 <i>Plan de versión</i>	60
Tabla 33 <i>Tarea de Ingeniería. Administrar usuarios del sistema.</i>	63

Tabla 34 <i>Tarea de ingeniería. Administrar juez de paz.</i>	64
Tabla 35 <i>Tarea de ingeniería. Administrar juzgado de paz.</i>	65
Tabla 36 <i>Tarea de ingeniería. Administrar recursos materiales y tecnológicos.</i>	66
Tabla 37 <i>Tarea de Ingeniería. Emitir constancias de transferencia posesoria.</i>	68
Tabla 38 <i>Tarea de ingeniería. Asignación de terna al juzgado.</i>	68
Tabla 39 <i>Tarea de ingeniería. Consultar Juzgados con ternas próximas a vencer.</i>	69
Tabla 40 <i>Tarea de Ingeniería. Iniciar sesión de jueces.</i>	70
Tabla 41 <i>Tarea de ingeniería. Iniciar sesión de administrador.</i>	71
Tabla 42 <i>Tarea de ingeniería. Administrar cuadro de aranceles.</i>	71
Tabla 43 <i>Tarea de ingeniería. Consultar usuario del sistema.</i>	73
Tabla 44 <i>Tarea de ingeniería. Consulta de jueces de paz.</i>	73
Tabla 45 <i>Tarea de ingeniería. Consultar juzgados de paz.</i>	74
Tabla 46 <i>Tarea de ingeniería. Consultar recursos materiales y tecnológicos.</i>	75
Tabla 47 <i>Tarea de ingeniería. Consultar juzgados con y sin ternas asignadas.</i>	75
Tabla 48 <i>Tarea de ingeniería. Consultar el cuadro de aranceles.</i>	76
Tabla 49 <i>Plan de iteración (primera)</i>	77
Tabla 50 <i>Plan de iteración (segunda)</i>	78
Tabla 51 <i>Plan de interacciones clasificadas en fechas de desarrollo</i>	79
Tabla 52 <i>Casos de prueba de aceptación</i>	80
Tabla 53 <i>Clase de Usuarios.</i>	93
Tabla 54 <i>Clase Juez de paz</i>	94
Tabla 55 <i>Clase Juzgado de paz</i>	94
Tabla 56 <i>Clase Recurso material y tecnológico.</i>	94
Tabla 57 <i>Clase Asignación de ternas</i>	95
Tabla 58 <i>Clase Cuadro de aranceles</i>	95
Tabla 59 <i>Clase Emisión de certificado</i>	95
Tabla 60 <i>Reporte de Pruebas Unitarias. Primera Iteración.</i>	129
Tabla 61 <i>Reporte de pruebas de integración. Primera iteración.</i>	130
Tabla 62 <i>Reporte de pruebas de aceptación. Administrar Usuarios del Sistema.</i>	131
Tabla 63 <i>Reporte de Pruebas de Aceptación: Administrar Juez de Paz.</i>	131
Tabla 64 <i>Reporte de pruebas de aceptación: Administrar Juzgado de Paz.</i>	131
Tabla 65 <i>Reporte de pruebas de aceptación. Administrar Recursos Materiales y Tecnológicos.</i>	132

Tabla 66 <i>Reporte de pruebas de aceptación. Emitir constancias de transferencia.</i>	131
Tabla 67 <i>Reporte de pruebas de aceptación. Asignar Terna al Juzgado.</i>	133
Tabla 68 <i>Reporte de pruebas de aceptación. Consultar Juzgados con Ternas Próximos a Vencer.</i>	133
Tabla 69 <i>Reporte de pruebas de aceptación: Administrar Cuadro de Aranceles.</i>	134

LISTA DE FIGURAS

Figura 1	<i>Esquema básico de una aplicación web</i>	18
Figura 2	<i>Ciclo de entrega en la programación extrema</i>	19
Figura 3	<i>Principios XP</i>	21
Figura 4	<i>Los roles en el proceso de desarrollo de XP</i>	23
Figura 5	<i>Etapas del proceso de desarrollo para un proyecto XP</i>	24
Figura 6	<i>Arquitectura Técnica inicial</i>	48
Figura 7	<i>Arquitectura técnica final. Diagrama de despliegue.</i>	62
Figura 8	<i>Arquitectura técnica final. Diagrama de componentes.</i>	62
Figura 9	<i>Interfaz, Sistema de administración de juzgados de paz</i>	82
Figura 10	<i>Interfaz. Menú principal del sistema</i>	82
Figura 11	<i>Interfaz. Administración de juzgado de paz</i>	83
Figura 12	<i>Interfaz. Registro de juzgado de paz</i>	83
Figura 13	<i>Interfaz. Administración de juez de paz.</i>	84
Figura 14	<i>Interfaz. Registro de juez de paz</i>	84
Figura 15	<i>Interfaz. Administración de cuadro de aranceles</i>	85
Figura 16	<i>Interfaz. Registro de cuadro de aranceles</i>	85
Figura 17	<i>Interfaz. Administración de recursos materiales</i>	86
Figura 18	<i>Interfaz. Registro de recurso material</i>	86
Figura 19	<i>Interfaz. Registro de asignación de jueces de paz</i>	87
Figura 20	<i>Interfaz. Registro asignación de terna</i>	87
Figura 21	<i>Interfaz. Administración de usuario</i>	88
Figura 22	<i>Interfaz. Registro cuenta de usuario.</i>	88
Figura 23	<i>Interfaz. Menú principal del sistema juez de paz</i>	89
Figura 24	<i>Interfaz. Administración de emisión de certificados</i>	89
Figura 25	<i>Interfaz. Modulo emisión de certificados</i>	90
Figura 26	<i>Interfaz. Certificado: Escritura pública de transferencia posesoria de bienes inmuebles</i>	90
Figura 27	<i>Interfaz. Certificado: Escritura pública de bienes muebles no registrables.</i>	91
Figura 28	<i>Interfaz. Administración de usuarios.</i>	91
Figura 29	<i>Interfaz. Actualización de la cuenta de usuario.</i>	92
Figura 30	<i>Interfaz. Administración de recursos tecnologicos</i>	92
Figura 31	<i>Interfaz. Registro de recurso tecnologico</i>	93

Figura 32 <i>Código fuente. Conexión a la base de datos (PostgreSQL)</i>	96
Figura 33 <i>Código fuente: usuario controller.</i>	96
Figura 34 <i>Código fuente: usuario model.</i>	99
Figura 35 <i>Código fuente: juez de paz model.</i>	100
Figura 36 <i>Código fuente: juez de paz vista.</i>	101
Figura 37 <i>Código fuente: juzgado de paz controller.</i>	103
Figura 38 <i>Código fuente: juez de paz model.</i>	105
Figura 39 <i>Código Fuente: juez de paz vista.</i>	106
Figura 40 <i>Código Fuente: Recurso Material y Tecnológico Controller.</i>	109
Figura 41 <i>Código Fuente: Recurso Material y Tecnológico Model.</i>	110
Figura 42 <i>Código Fuente. Recurso Material y Tecnológico Vista.</i>	111
Figura 43 <i>Código Fuente: Certificados Emitidos Controller.</i>	113
Figura 44 <i>Código Fuente: Certificados Emitidos Model.</i>	115
Figura 45 <i>Código Fuente: Certificados Emitidos Vista.</i>	116
Figura 46 <i>Código Fuente: Asignar Juez Controller.</i>	118
Figura 47 <i>Código Fuente. Asignar Juez Model.</i>	121
Figura 48 <i>Código Fuente: Asignar Juez Vista</i>	123

RESUMEN

La oficina distrital de apoyo a la justicia de paz de Ayacucho (ODAJUP - Ayacucho) tiene 384 juzgados de paz, pero su sistema de información es deficiente. Los datos se registran en hojas de cálculo y libros de actas que no se actualizan de manera oportuna, lo que genera problemas en la gestión de los juzgados de paz.

El propósito del estudio es desarrollar una aplicación web para realizar la gestión de recursos en los Juzgados de Paz del Distrito Judicial de Ayacucho, con información del año 2023. El tipo de investigación es; observacional, prospectivo, transversal y descriptiva.

Se ha desarrollado la aplicación web mediante la programación extrema, un administrador de base de datos relacional y las pruebas unitarias con Jest, para realizar el registro de; asignación de la terna de jueces a los juzgados de paz, jueces de paz, juzgados de paz, y certificados emitidos.

Se ha logrado desarrollar la gestión de recursos humanos mediante la aplicación web como se observa en las tablas 15, 16, 20, 21, 25, 26, 29 y las figuras 13, 14, 19, 20, 21, 22; la gestión de recursos financieros según las tablas 19, 24, 30 y las figuras 15, 16, 23, 24, 25, 26, 27; la gestión de recursos materiales de acuerdo a las tablas 18, 26 y las figuras 17, 18; la gestión de recursos tecnológicos se observa en las tablas 18, 26 y las figuras 30, 31. La base de datos relacional y el código fuente que permiten almacenar datos de forma organizada, modificarlos según sea necesario y consultarlos de forma eficiente.

Palabras claves: Programación extrema, aplicación web, juzgado de paz, jueces de paz.

ABSTRACT

The Ayacucho District Office of Support for Justice of Peace (ODAJUP - Ayacucho) has 384 Justice of Peace courts, but its information system is deficient. Data is recorded in spreadsheets and minutes books that are not updated in a timely manner, which creates problems in the management of the Justice of Peace courts.

The purpose of the study is to develop a web application to manage resources in the Justice of Peace Courts of the Judicial District of Ayacucho, with information from the year 2023. The type of research is: observational, prospective, cross-sectional, and descriptive.

The web application has been developed using extreme programming, a relational database manager, and unit tests with Jest, to register: assignment of the panel of judges to the Justice of Peace courts, Justice of Peace judges, Justice of Peace courts, and certificates issued.

The management of human resources has been achieved through the web application as shown in tables 15, 16, 20, 21, 25, 26, 29 and figures 13, 14, 19, 20, 21, 22; financial resource management according to tables 19, 24, 30 and figures 15, 16, 23, 24, 25, 26, 27; the management of material resources according to tables 18, 26 and figures 17, 18; the management of technological resources is observed in tables 18, 26 and figures 30, 31. Relational database and the source code that allow data to be stored in an organized manner, modified as needed, and queried efficiently.

Keywords: Extreme programming, web application, peace court, peace judges.

CAPÍTULO I INTRODUCCIÓN

1.1. PLANTEAMIENTO DEL PROBLEMA

La oficina distrital de apoyo a la justicia de paz de Ayacucho (ODAJUP - AYACUCHO) tiene 384 juzgados de paz dentro de la jurisdicción de Ayacucho, a) esta oficina cuenta con un formato de archivo en Excel, en la cual se registra la información de los juzgados de paz, así como, la terna de jueces de paz y el número de resolución administrativa, fecha de inicio, dirección, observación, fecha de finalización de terna; que al ser hojas de cálculo, solo se registra la información y no generan reportes actualizados, originado que los juzgados de paz, se queden sin juez de paz, b) se cuenta con libro de actas, donde registra los bienes asignados, como mobiliarios y equipo de cómputo, a los juzgados de paz que muchas veces son poco legibles o modificados y no coinciden con la realidad en cada juzgado de paz c) los jueces de paz tienen como una de sus actividades la función notarial por lo que emiten diferentes tipos de certificados que al ser manuales muchas veces no son registrados por la cual genera conflictos cuando asume la nueva terna; además el costo de los certificados está estandarizado y validado por la oficina nacional de justicia de paz (ONAJUP) quienes publican el cuadro de aranceles judiciales anualmente y muchas veces sin embargo los jueces de paz no siempre están al tanto de estas actualizaciones lo que perjudica una de las fuentes de ingreso del juzgado de paz.

Si los juzgados de paz vienen funcionando como actualmente se encuentran, a) Se incrementa la cantidad de juzgados de paz que se quedan sin terna de jueces y por tanto el centro poblado o comunidad campesina no tienen dónde acudir para que se imparta justicia, lo que puede generar un incremento de casos en los Juzgados de Paz Letrados y demora en la resolución de sus casos; además b) Se tendrían juzgados de paz que se encuentran en situaciones precarias para impartir justicia y cumplir las funciones notariales sin bienes inmuebles ni los recursos tecnológicos debido al desconocimiento del estado situacional actual del juzgado de paz; también c) Se incurre en las especulaciones de los precios de emisión de certificados notariales que afectan a las dos partes al jueces de paz y a los usuarios debido a que no se tiene el cuadro de aranceles actualizados el juez de paz realizará un cobro inferior al precio establecido por la ONAJUP o también un precio superior, lo que generaría que cada vez menos personas quieran asumir ser jueces de paz o que los usuarios realizar quejas por los precios excesivos.

El desarrollo de una aplicación web apoyará , a) Que exista una alerta que de aviso con un plazo mínimo de 2 meses que las ternas están próximas de vencer para que la ODAJUP pueda enviar los oficios a los juzgados de paz pidiendo nueva terna de jueces de paz; b) Que se pueda listar los bienes muebles y recursos tecnológicos de cada juzgado de paz para que la ODAJUP realice los trámites de asignación de estos recursos; además c) A que se realicen el registro de los aranceles judiciales la cual tiene que ser actualizado por la responsable de la ODAJUP.

1.2. FORMULACIÓN DEL PROBLEMA

1.2.1. Problema General

¿Cómo realizar la gestión de recursos en los juzgados de paz del distrito judicial de Ayacucho, 2023?

1.2.2. Problemas Específicos

¿Cómo realizar la gestión de los recursos: humanos, financieros, materiales y tecnológicos?

1.3. JUSTIFICACIÓN DE LA INVESTIGACIÓN

Esta investigación surge de un interés personal de entender el funcionamiento de los juzgados de paz del distrito judicial de Ayacucho y, su relación con la oficina distrital de apoyo a la justicia de paz de Ayacucho, para crear un modelo que pueda ser escalable y/o replicable a nivel nacional de aplicación web.

Así mismo, el interés de desarrollar una aplicación web, mediante la exploración, planificación e interacción, para automatizar la gestión de recursos humanos, financieros, materiales y tecnológicos utilizando la metodología ágil XP.

Principalmente por apoyar la gestión mediante la automatización de; a) El proceso de asignación de jueces de paz, b) El proceso de emisión de certificados notariales con su respectivo costo según cuadro de aranceles y c) La asignación de recursos materiales y tecnológicos; y así mismo ayudar a la oficina de ODAJUP a la toma de decisiones.

1.4. LIMITACIONES DE LA INVESTIGACIÓN

Una limitación importante del sistema es que la mayoría de los juzgados de paz no cuentan con conexión a internet, lo que imposibilita su implementación en todo el distrito judicial, No se realizará la entrevista a todos los jueces de paz para la obtención de información de los certificados notariales debido a la falta de recursos financieros y tiempo, sin embargo, se usará 14 formatos las cuales son estandarizadas por la ONAJUP.

1.5. OBJETIVOS

1.5.1. Objetivo General

Desarrollar una aplicación web para la gestión de recursos en los juzgados de paz del distrito judicial de Ayacucho, 2023.

1.5.2. Objetivos Específicos

- a. Explorar, planificar e iterar la gestión del recurso humano
- b. Explorar, planificar e iterar la gestión del recurso financiero
- c. Explorar, planificar e iterar la gestión del recurso material
- d. Explorar, planificar e iterar la gestión del recurso tecnológico

CAPÍTULO II

MARCO TEÓRICO

2.1. ANTECEDENTES DE LA INVESTIGACIÓN

Flores, A., & Roque, M. (2015), Desarrollo de un sistema de registro único y gestión documentaria de los juzgados de paz del distrito judicial de Puno [Título Profesional], Universidad Nacional del Altiplano en su trabajo de investigación concluye que la implementación del sistema de registro único y gestión documentaria acelerará el tiempo de búsqueda y emisión de documentos en los juzgados de paz del distrito judicial de Puno.

Purificación. J. &, Valverde, J (2018), Desarrollo un sistema de información con tecnología web para modernizar la verificación de autenticidad de documentos expedidos por jueces de paz en la zona registral n.º VII sede – Huaraz [Título Profesional], Universidad Nacional Santiago Antúnez de Mayolo facultad de ciencias en su trabajo de investigación concluye que la implementación del sistema de información con tecnología web moderniza la verificación de la formalidad en la verificación de autenticidad de los documentos expedidos por jueces de paz en la zona registral n.º vii sede – Huaraz; evitando así el uso de oficios en papel y la pérdida de los mismos, ello hacía más dificultoso la precalificación de títulos, pero actualmente eso ha mejorado significativamente.

2.2. MARCO CONCEPTUAL

2.2.1. Juzgado de Paz

Es un órgano jurisdiccional del poder judicial cuya ubicación jerárquica se encuentra establecida por el artículo 26 de la presente ley orgánica. La elección, atribuciones, deberes, derechos y demás aspectos vinculados a esta institución, son regulados por la ley especial de la materia, (Ley Orgánica del Poder Judicial, 1985, artículo 61).

Baltodano, (2021) Afirma que los Juzgados de Paz (anteriormente llamado “Juzgado de Paz no Letrado”) constituyen el menor nivel jerárquico en que se encuentra organizado el Poder Judicial. Interviene como conciliador y contribuye a que las personas resuelvan sus conflictos de modo directo. De no llegar a un acuerdo, el Juez de Paz podría dictar sentencia judicial. Por lo general, el Juez de Paz no requiere ser abogado para ejercer el cargo porque, preferentemente, resuelve de acuerdo al sentido común y en equidad, a fin de restablecer los vínculos vecinales y mantener la paz social. Los jueces de paz son importantes en la comunidad, porque se mantienen como una gran opción para resolver conflictos y para que se sancionen o corrijan diversos actos. (p.7)

2.2.2. Juez de Paz

Según la Ley de Justicia de Paz, Ley N° 29824, es un ciudadano reconocido y respetado por su comunidad, es elegido por ésta democráticamente mediante un proceso de elecciones para el que son convocados los pobladores que radican o viven en el área física en la que ejercerá jurisdicción. Por lo general, resuelve los conflictos que son sometidos a su conocimiento mediante la conciliación y también mediante decisiones de carácter

jurisdiccional o sentencias, (Ley de Justicia de Paz, Ley N° 29824, artículo 3).

El juez de paz es un ciudadano honorable de la localidad. Observa una buena conducta pública, basada en los valores y principios establecidos en la constitución política del Perú. Su recta conducta lo legitima socialmente para exigir el cumplimiento de sus decisiones, sean de carácter dirimente o sancionador, (Ley de Justicia de Paz, 2001, art. VI; Reglamento de la Ley de Justicia de Paz, 2002, art. 24)

2.2.3. Recurso Humano

Barquero (2005, p. 12) Afirma que los empleados son el activo más valioso de una empresa, y su gestión es fundamental para su éxito. El elemento humano está presente en todos los aspectos de una empresa, por lo que es esencial que los gerentes y directores tengan en cuenta las necesidades de sus empleados para obtener resultados positivos.

Montoya y Boyero (2016, p. 108) afirman que la dirección estratégica de los recursos humanos es fundamental para el éxito de la organización. El propósito fue enfatizar algunas anotaciones relevantes que giran en torno a la representación del valor de los recursos humanos en la organización y su base como elemento básico de dirección estratégica. La metodología fue de revisión documental dónde se abordaron los temas de direccionamiento y trabajo en equipo de los Recursos Humanos como aspectos determinantes en el éxito organizacional. Como resultado, los autores mencionaron que hay posibilidad futura de que una organización dependa no solo de sus capacidades financieras, sino también de su capacidad para producir cambios en la estructura y los procesos. Por lo tanto, concluyen que la organización debe orientarse hacia una visión más ambiciosa de los Recursos Humanos vinculados con la Gestión por Competencias para el éxito organizacional.

Dos (2) meses antes de que expire el mandato del juez de paz, la presidencia de la corte superior de justicia oficiará al alcalde distrital, al presidente de la comunidad o al agente municipal del centro poblado menor para que convoque a los vecinos de la localidad a una elección directa y democrática. En las comunidades campesinas y nativas que cuenten con juzgados de paz, las elecciones se llevarán conforme a sus usos y costumbres. Posteriormente cada corte superior de justicia revisará que las personas elegidas cumplan los requisitos establecidos en la presente ley, (Ley de Justicia de Paz, 2001, art. 14).

2.2.4. Recursos Financiero

La justicia de paz es gratuita en general, pero el Consejo Ejecutivo del Poder Judicial puede establecer aranceles por diligencias y actividades especiales. Los procesos jurisdiccionales son gratuitos, pero el juez de paz puede cobrar un arancel por actuaciones fuera de su despacho o por funciones notariales. El monto máximo de los aranceles será fijado por el Consejo Ejecutivo, (Ley de Justicia de Paz, 2001, artículo VI).

Según el Reglamento de la Ley de Justicia de Paz (2002, art. 24.3), los documentos estandarizados que pueden ser expedidos por el juez de paz son: (a) escritura de transferencia posesoria del bien inmueble; (b) escritura de transferencia de bien mueble no registrable; (c) certificación de firmas; (d) constancia de posesión de bien inmueble; (e) declaración jurada sobre posesión de bien inmueble; (f) constancia de posesión de bien mueble no registrable; (g) declaración jurada sobre posesión de bien mueble no registrable; (h) certificación de apertura de libros; (i) certificación de transcripción de documentos; (j) constancia de actos y decisiones de organizaciones sociales; (k) constancia domiciliaria; (l)

constancia de viudez; (m) constancia de abandono de hogar; (n) constancia de separación de cuerpos; (o) constancia de separación de hechos; (p) modelo general de constancia.

En este sentido, según Schiff y Schiff (2010), afirma que el rol de la gestión financiera en una entidad – destacando a los encargados – se orienta a enfrentar los desafíos actuales, y los retos consisten en manejar adecuadamente cada elemento de las finanzas, los cuales apoyan el correcto funcionamiento de las operaciones financieras y que procuren la calidad de los servicios.

Según La Rosa (2007), el financiamiento del juzgado de paz, el juez de paz trabaja sin recibir sueldo del estado. Tampoco puede cobrar por administrar justicia. Sólo puede realizar cobros por sus funciones notariales y por los gastos que ocasionen las diligencias fuera del juzgado.

De acuerdo a las normas el financiamiento debería ser compartido entre los concejos municipales y la comunidad, para brindar los locales de funcionamiento, y el poder judicial que debe proporcionar los útiles como papel, mobiliario, máquina de escribir entre otros. Sin embargo esto sucede con poca frecuencia (ley orgánica del poder judicial, 2012, artículo 63).

2.2.5. Recurso Material

Según la Ley N° 29824 (2012, artículo 41), las cortes superiores de justicia están obligadas a proporcionar a los juzgados de paz, identificación o insignias propias del cargo. Las comunidades y las municipalidades de la circunscripción correspondiente deben apoyar la implementación logística del juzgado de paz.

Según La Rosa (2007, p. 45), de acuerdo a las normas el financiamiento debería ser compartido entre los concejos municipales y la comunidad, para brindar los locales de funcionamiento, y el poder judicial que debe proporcionar los útiles como papel, mobiliario, máquina de escribir entre otros. Sin embargo esto sucede con poca frecuencia.

2.2.6. Recurso Tecnológico

Por otro lado, Quevedo (2009) señaló que el desarrollo tecnológico en las organizaciones está incorporado en la cadena de valor y se realiza de manera sistemática mediante procesos, con el objetivo de cumplir las operaciones que están en la gestión tecnológica.

Morin (1992) amalgama las nociones de recursos y tecnología, y propone una definición de recursos tecnológicos como el conjunto de medios materiales e inmateriales que la empresa dispone y/o a los que pudiera acceder para el diseño, fabricación, comercialización de sus productos y/o servicios, uso de la información y la gestión de todas las funciones que contribuyen a la concreción de sus actividades. (p. 12)

2.3. MARCO REFERENCIAL

2.3.1. Sistema Web

Pressman, Roger (2014) En su libro "Ingeniería de Software", Pressman define un sistema web como "una aplicación de software que se accede y se usa a través de una red, como Internet o una intranet". Enfatiza en la naturaleza distribuida de los sistemas web, donde los componentes se encuentran alojados en diferentes servidores y se comunican a través de protocolos de red.

Larman, Craig (2004) En su libro "Advancing Agile", Larman ofrece una definición más amplia de los sistemas web, considerándolos como "sistemas software que se ejecutan en servidores y se entregan a usuarios finales a través de una variedad de dispositivos, incluyendo navegadores web, teléfonos móviles y dispositivos portátiles". Esta definición abarca la naturaleza multiplataforma de los sistemas web y su capacidad para adaptarse a diferentes interfaces.

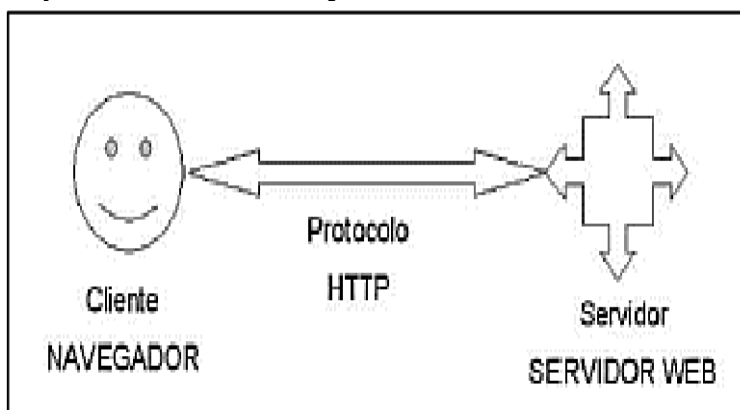
a. **Aplicaciones Web**

Seoane (2005) Las aplicaciones web son programas que se ejecutan en un navegador web y se dividen en tres capas: cliente, servidor de lógica del negocio y servidor de base de datos. El cliente se encarga de la interfaz de usuario, el servidor de lógica del negocio procesa los datos y el servidor de base de datos almacena los datos.

Luján (2001), Las aplicaciones web son aplicaciones cliente-servidor que se ejecutan en un navegador web. El navegador web y el servidor web utilizan un protocolo estandarizado, el HTTP, para comunicarse.

Figura 1

Esquema básico de una aplicación web



Fuente: (Mora, M 2002).

b. **Protocolo HTTP**

El protocolo HTTP es un protocolo cliente-servidor que se utiliza para transferir datos entre un navegador web y un servidor web. El navegador web envía una solicitud al servidor web, que responde con los datos solicitados. (Salavert, 2003, p. 142).

El protocolo HTTP es un protocolo de comunicación que se utiliza para transferir datos entre sistemas heterogéneos en Internet. Forma parte de la familia de protocolos TCP/IP, que permite la comunicación entre ordenadores de diferentes tipos y sistemas operativos. (Luján, 2001, p.8).

c. **Protocolo TCP/IP**

Acón, Trujillo y Guido (2011) Define el protocolo TCP/IP es un conjunto de protocolos que permite la comunicación entre ordenadores en Internet. Está compuesto por dos protocolos principales: TCP, que garantiza la entrega fiable de datos, e IP, que identifica a los ordenadores y establece las rutas de comunicación.

El protocolo TCP/IP es un conjunto de protocolos que permite la comunicación entre

ordenadores en Internet. El protocolo IP especifica el formato de los paquetes que se envían y reciben entre los routers y los sistemas terminales. El protocolo TCP garantiza la entrega fiable de los datos. (Mora, 2002).

“IP y TCP son protocolos de red que permiten la comunicación entre dos puntos. IP se encarga de la entrega de los datos a la red adecuada, mientras que TCP garantiza que los datos se transmitan de forma fiable, en el orden correcto y sin errores.” (Barceló, 2004, 71).

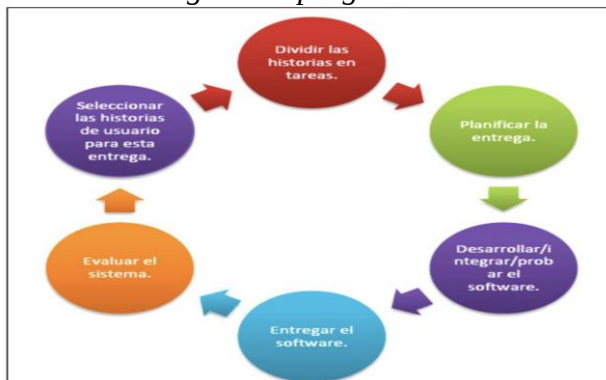
2.3.2. Programación Extrema (XP)

Según Pressman (2010) define es un procedimiento sistemático para el desarrollo de software. Centra en las relaciones entre los empleados y los clientes. Es especialmente útil para empresas que están comenzando o que están pasando por cambios, ya que ayuda a mejorar la comunicación y la colaboración. Se basa en cuatro principios clave trabajo en equipo: enfatiza el trabajo en equipo y la colaboración entre todos los miembros del equipo de desarrollo. Revisión continua: requiere que el equipo de desarrollo reciba comentarios del cliente de manera regular. Refinamiento incremental: divide el desarrollo del software en iteraciones cortas e incrementales. Simplicidad: promueve la simplicidad y la claridad en el diseño del software.

La programación extrema, también conocida como XP, es un enfoque de desarrollo de software que se centra en la colaboración, la simplicidad y la calidad. XP se basa en ciclos de retroalimentación rápidos e iteraciones cortas para garantizar que el software sea lo que los clientes necesitan y que se pueda entregar rápidamente. (Kendall & Kendall, 2005, p. 165).

Para Sommerville (2005), es una metodología ágil de desarrollo de software que se centra en la comunicación, la colaboración y la retroalimentación continua. Se basa en los siguientes principios: el desarrollo se divide en iteraciones cortas, cada una de las cuales produce una versión funcional del software. El cliente participa activamente en el desarrollo del software, proporcionando retroalimentación regular. Utiliza las siguientes prácticas. Escenarios de usuario: Los requisitos se expresan como escenarios de usuario, que describen lo que el software debe hacer parejas de programación: Los programadores trabajan en parejas para mejorar la comunicación y la colaboración. Pruebas unitarias: Las pruebas se desarrollan antes de escribir el código para garantizar que el software funcione correctamente. Es una metodología adecuada para proyectos de software con requisitos cambiantes o con clientes que tienen un papel activo en el desarrollo.

Figura 2
Ciclo de entrega en la programación extrema



Fuente: Sommerville (2005).

2.3.3. Actividades Básicas de XP

Según Kendall, K. E., & Kendall, J. E. (2005), La programación extrema se basa en cuatro actividades básicas: codificar, probar, escuchar y diseñar. El codificar es el proceso de escribir el código fuente del software. Probar es el proceso de verificar que el software cumpla con sus requisitos. Escuchar es el proceso de recopilar información del cliente y otros interesados. Diseñar es el proceso de crear un plan para el software. Estas actividades se realizan en ciclos cortos e iterativos, lo que permite al equipo de desarrollo obtener retroalimentación rápida y ajustar el software según sea necesario.

2.3.4. Variables De Control De Recursos De XP

La programación extrema requiere de recursos para realizar sus actividades básicas. Estos recursos pueden ser ajustados para completar el proyecto antes de una fecha límite. Las cuatro variables de control que se pueden ajustar son tiempo, costo, calidad y alcance. Cuando estas variables se incluyen adecuadamente en la planeación, se logra un equilibrio entre los recursos y las actividades necesarias para completar el proyecto. (Kendall, K. E., & Kendall, J. E, p. 170).

2.3.5. Valores de la Programación Extrema

Según Kendall, K. E., & Kendall, J. E. (2005), La programación extrema (XP) se basa en cuatro valores: comunicación, simplicidad, retroalimentación y valentía. La comunicación es fundamental, ya que los miembros del equipo deben trabajar juntos para comprender los requisitos del cliente y solucionar problemas. Las prácticas de XP, como la programación en parejas y las pruebas del software, están diseñadas para apoyar estos valores. Un instructor de XP puede ayudar a los equipos si hay problemas de comunicación y para ayudar a resolverlos.

a. Comunicación

La comunicación es esencial para el éxito de cualquier proyecto de software. Los equipos de XP se comunican de manera efectiva entre sí, con los clientes y con las partes interesadas.

b. Simpleza

El software debe ser lo más simple posible. Esto facilita su comprensión, mantenimiento y evolución.

c. Retroalimentación

La retroalimentación debe ser rápida y constante. Esto permite a los equipos de XP identificar y corregir los problemas rápidamente.

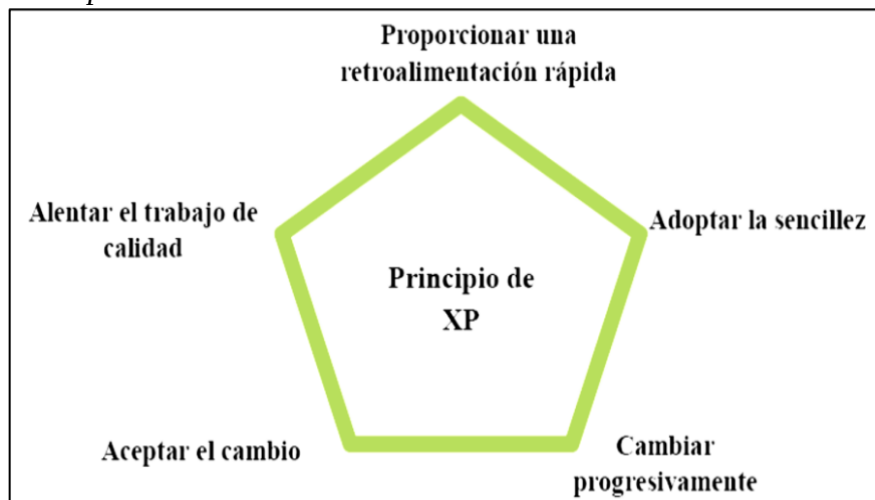
d. Valentía

Los equipos de XP deben ser valientes para tomar riesgos y probar cosas nuevas. Esto les permite aprender y mejorar continuamente.

2.3.6. Principios Básicos De XP

De acuerdo con Kendall y Kendall (2005), Los cuatro valores de XP (comunicación, simplicidad, retroalimentación y valentía) se traducen en cinco principios básicos: para guiar al analista de sistemas a través de un proyecto de XP exitoso (Retroalimentación rápida, sencillez, cambios progresivos, aceptación del cambio, trabajo de calidad).

Figura 3
Principios XP



Fuente: Kendall, K., Kendall, J., 2005

a. La Retroalimentación Rápida

Los equipos de XP reciben retroalimentación del cliente de forma regular, lo que les permite ajustar el software según sea necesario.

b. Adoptar la Sencillez

La programación extrema se basa en la premisa de que la simplicidad es la clave del éxito. Los programadores deben centrarse en soluciones simples que se puedan ampliar más adelante.

c. Cambiar Progresivamente

La metodología ágil XP se basa en el cambio incremental. Los desarrolladores deben realizar cambios pequeños y graduales al código, el equipo y los requisitos del negocio. Estos cambios se realizan de forma progresiva, incluso después de que el producto se ha liberado.

d. Aceptar el Cambio

En XP, se acepta el cambio como una oportunidad para mejorar el producto. Para aprovechar el cambio, los equipos de XP deben ser flexibles y adaptables. Esto significa estar dispuestos a cambiar los planes y los procesos según sea necesario. Los equipos de XP también deben ser capaces de seguir adelante. Esto significa mantener el enfoque en el objetivo final, incluso cuando se producen cambios.

e. Trabajo de Calidad

Los equipos de XP se centran en crear software de alta calidad, que satisfaga las necesidades del cliente y sea fiable.

2.3.7. Prácticas de la Programación Extrema

La programación extrema implica varias prácticas, resumidas en la siguiente tabla 1 que se ajustan a los principios de los métodos ágiles:

Tabla 1*Prácticas de programación extrema*

PRINCIPIO O PRÁCTICA	DESCRIPCIÓN
Planificación incremental	Los requisitos del software se documentan en tarjetas de historias, que se priorizan según su importancia y viabilidad. Las historias que se incluirán en una entrega se determinan en función del tiempo disponible y de la prioridad relativa. Los desarrolladores dividen estas historias en tareas de desarrollo más pequeñas y manejables.
Entregas pequeñas	El desarrollo de software se centra en crear el conjunto mínimo de funcionalidad que proporcione valor al cliente lo antes posible. Las entregas del sistema se realizan con frecuencia, añadiendo funcionalidad a cada entrega.
Diseño sencillo	Se diseña sólo lo que es necesario para satisfacer las necesidades actuales.
Desarrollo previamente probado	Se utiliza un sistema de pruebas de unidad automatizado para escribir pruebas para nuevas funcionalidades antes de que éstas se implementen.
Refactorización	Los desarrolladores deben refactorizar el código de forma continua para mejorar su legibilidad, mantenibilidad y flexibilidad.
Programación en parejas	Los desarrolladores trabajan en colaboración, revisando y ayudándose mutuamente para garantizar la calidad del código.
Propiedad colectiva	Los desarrolladores trabajan en todas las áreas del sistema para evitar la especialización y fomentar la propiedad compartida del código. Esto permite que cualquier desarrollador pueda realizar cambios en cualquier parte del sistema.
Integración continua	El código nuevo se integra en el sistema de forma frecuente para detectar errores lo antes posible. Las pruebas de unidad se ejecutan después de cada integración para garantizar la calidad del código.
Ritmo sostenible	Las horas extras excesivas no son recomendables, ya que pueden tener un impacto negativo en la calidad del código y la productividad a largo plazo

Cliente presente	En la programación extrema, el cliente es un miembro activo del equipo de desarrollo. Debe estar disponible a tiempo completo para proporcionar al equipo información sobre los requisitos del sistema y para proporcionar comentarios sobre el progreso del desarrollo.
------------------	--

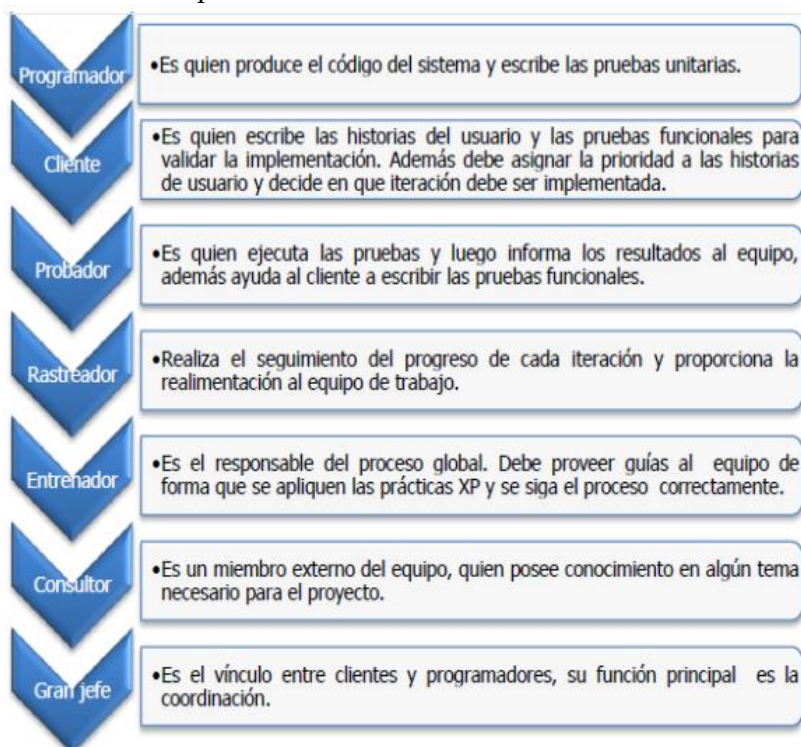
Fuente: Sommerville (2005).

2.3.8. Roles en el Proceso de Desarrollo de XP

Según Kendall, K. y Kendall, J. (2005), La programación extrema (XP) es un enfoque flexible para el desarrollo de software que se centra en la colaboración entre los miembros del equipo, la comunicación fluida y la simplicidad. Los proyectos de xp se dividen en ciclos cortos de desarrollo, llamados iteraciones. Al final de cada iteración, se entrega un producto funcional al cliente. Hay siete roles principales: el programador es responsable de escribir el código. El cliente es la persona o el equipo que necesita el software. El probador es responsable de probar el software para asegurarse de que funciona correctamente. El rastreador es responsable de mantener el seguimiento del progreso del proyecto. El entrenador es responsable de ayudar al equipo a aprender y aplicar las prácticas de xp. Consultor: el consultor es un experto en xp que puede proporcionar orientación al equipo. En algunos proyectos, una persona puede desempeñar varios roles. Por ejemplo, un programador también puede ser probador. En xp, el equipo de desarrollo es autosuficiente y se encarga de todo el trabajo necesario para desarrollar y mantener el software. Trabajan juntos en un entorno colaborativo para crear software de alta calidad que satisfaga las necesidades del cliente.

Figura 4

Los roles en el proceso de desarrollo de XP



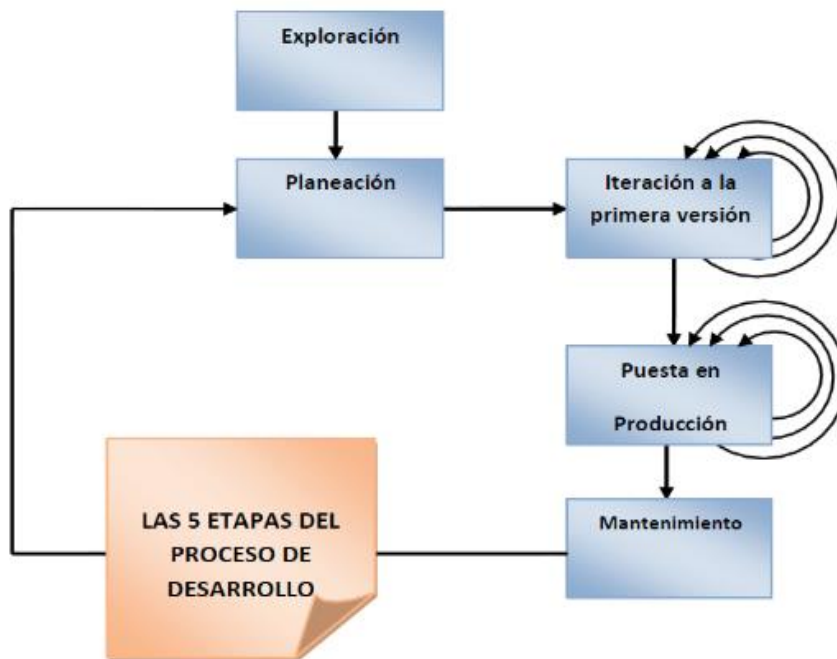
Fuente: Kendall, K., Kendall, J., 2005.

2.3.9. Ciclo de Vida Ideal de un Proyecto XP

Según Kendall, K. y Kendall, J. (2005), Los proyectos de XP se basan en dos conceptos fundamentales: la interactividad y la incrementalidad. Los desarrolladores trabajan en estrecha colaboración con los clientes para comprender sus necesidades y entregarles software que cumpla con sus expectativas. El desarrollo se lleva a cabo en iteraciones cortas, lo que permite a los clientes ver el progreso del proyecto de forma regular y proporcionar comentarios. El proceso de XP se divide en cinco etapas: Exploración, Planificación, Iteraciones a la primera versión, Puesta en producción, Mantenimiento.

Figura 5

Etapas del proceso de desarrollo para un proyecto XP



Fuente: Kendall, K., Kendall, J., 2005.

a. Exploración

En la programación extrema es un período de investigación y planificación. Durante esta etapa, el equipo de desarrollo se familiariza con el entorno, el problema a resolver y las tecnologías disponibles. También comienza a construir relaciones con el cliente y a desarrollar una comprensión de las necesidades del cliente. Las tareas principales de la etapa de exploración son:

- Investigar el entorno:** El equipo de desarrollo debe comprender el entorno en el que se implementará el software. Esto incluye comprender los requisitos del negocio, los usuarios del software y las tecnologías disponibles.
- Construir relaciones con el cliente:** El equipo de desarrollo debe establecer una relación sólida con el cliente. Esto ayudará a garantizar que el software satisfaga las necesidades del cliente.
- Desarrollar un plan de desarrollo:** El equipo de desarrollo debe desarrollar un plan de desarrollo que defina el alcance del software, el cronograma y el presupuesto.

Tabla 2*Fase de exploración*

TAREA	ARTEFACTO	TÉCNICA	RESPONSABLES
Escribir historias de usuario	Historia de usuario	Identificar los requisitos funcionales y no funcionales del software. Establecer un orden de prioridad para las partes de la historia.	Cliente
Probar las tecnologías a utilizar	Lista de tecnologías a utilizar	Identificar tecnologías viables, evaluar su rendimiento y seleccionar las mejores.	Programador Entrenador
Establecer arquitectura técnica inicial	Arquitectura técnica inicial	Definir la estructura y los componentes de la arquitectura técnica.	Programador Entrenador
Estimar esfuerzo para historia de usuario	Plan de alto nivel	Analizar la historia de usuario y estimar su esfuerzo de desarrollo.	Programador

Fuente: Creado por autor

b. Planeación

(Parker, 2020). En esta etapa, el equipo de desarrollo y el cliente se reúnen para establecer un cronograma para el desarrollo del software. El cronograma se basa en la urgencia de los problemas del cliente y en la información recopilada durante la etapa de exploración. Si las actividades de Si la etapa de exploración proporcionó la información necesaria, esta etapa puede ser corta. Sin embargo, si se necesitan más información o si el cliente tiene requisitos cambiantes, la planeación puede ser más prolongada.

A partir de la información citada y de otras fuentes, se presenta un resumen de la siguiente tabla, que incluye las tareas, los artefactos, la técnica y los responsables del desarrollo.

Tabla 3*Fase de planeación*

TAREA	ARTEFACTO	TÉCNICA	RESPONSABLES
Establecer prioridad de historias de usuario	Historias de Usuario por prioridad	Clasificar las historias de usuario según su contribución al éxito del negocio.	Cliente
Estimar esfuerzo para las historias de usuario	Estimación de esfuerzo	Evaluar el trabajo necesario para completar cada historia de usuario, incluyendo la planificación, el diseño, la implementación y las pruebas.	Programador
Elaborar el plan de entrega	Plan de versión	Determinar las fechas de entrega de cada tarea o hito del plan de entrega.	Programador

Fuente: Creado por autor

c. Iteraciones a la Primera Versión

Beck, K., Beedle, M. (2001) En la tercera etapa del proceso de desarrollo de XP, el equipo de desarrollo comienza a implementar las historias de usuario. Estas iteraciones, que duran aproximadamente tres semanas, consisten en un proceso iterativo de desarrollo, pruebas y retroalimentación. Durante esta etapa, el equipo de desarrollo también debe desarrollar la arquitectura del sistema y realizar pruebas funcionales del software. Además, es importante revisar las fechas programadas y el trabajo pendiente para asegurarse de que el proyecto está progresando según lo previsto. Al final de cada iteración, el equipo de desarrollo debe celebrar una pequeña ceremonia con el cliente y los desarrolladores para reconocer sus logros. Esta celebración es importante para motivar al equipo y mantener su entusiasmo.

A partir de la información citada y de otras fuentes, se presenta un resumen de la siguiente tabla, que incluye las tareas, los artefactos, la técnica y los responsables del desarrollo.

Tabla 4*Fase de iteración a la primera versión*

TAREA	ARTEFACTO	TÉCNICA	RESPONSABLES
Definir la arquitectura técnica	Arquitectura técnica	Alinear la arquitectura técnica con los objetivos del negocio. Considerar las necesidades del negocio al diseñar el sistema.	Cliente Programador Entrenador

		Arquitectura por capas y frameworks	
Escribir tareas de ingeniería	Tarea de ingeniería	Convertir las historias de usuario en tareas de ingeniería que cumplan con los requisitos del negocio.	Cliente programador
Formular el plan de iteraciones	Plan de iteración	Estimar el tiempo y los recursos necesarios para completar una tarea de ingeniería y asignarlos a un período de desarrollo definido.	Programador
		Asignar a un programador para realizar una tarea de ingeniería. Utilizar el plan de iteraciones en el plan de versión. Incorporar las tareas de ingeniería de la siguiente iteración al plan. Actualizar los criterios de aceptación de las pruebas Actualizar las tareas no concluidas al plan de iteraciones. Actualizar las tareas de ingeniería según el plan de iteraciones.	Entrenador Programador
Crear pruebas de aceptación	Caso de prueba de aceptación	En cada iteración, el equipo de desarrollo debe crear pruebas que verifiquen que cada historia de usuario cumple con los requisitos del cliente.	Cliente Encargado de pruebas

Implementar las interfaces	GUI	Crear una interfaz gráfica de usuario (GUI) que satisfaga las necesidades de cada historia de usuario, utilizando una herramienta de desarrollo de software.	Cliente Programador
Escribir tarjetas CRC para cada tarea de ingeniería	Tarjeta CRC	Elaborar un plan sencillo para la implementación de una actividad que requiere conocimientos y habilidades de ingeniería. Modificar una tarea debido a que la prueba de aceptación no fue exitosa. Delinear las funciones y los roles. Comprender el trabajo en equipo. Determinar las características.	Cliente Programador
Implementar la base de datos física	Base de datos física	Desarrollar código usando tarjeta (CRC) y Implementarlo en un sistema de gestión de bases de datos. (DBMS).	Programador
Implementar código para clases entidad	Código fuente	Crear el código fuente de forma manual o automática mediante una herramienta basada en tarjetas. CRC.	Programador
Crear Pruebas unitarias para las clases control	Prueba unitaria	Se escribe el código de la prueba, se genera un archivo de prueba y se ejecuta la prueba.	Programador
Implementar código fuente	Código fuente	Elaborar y llevar a cabo un proyecto de ingeniería. Hacer que el código sea más fácil de entender y mantener.	Programador Supervisor

		Asignar recursos humanos a tareas específicas.	
Ejecutar pruebas unitarias	Reporte de prueba unitaria	Ejecutar cada prueba unitaria y realizar los cambios necesarios en el código fuente si la prueba falla.	Programador
Realizar integración continua	Código fuente	Agrupar tareas de una historia de usuario puede ayudar a mantener el sistema unido.	Programador
Ejecutar pruebas de integración para una historia de usuario	Reporte pruebas de integración	El código de las historias de usuario se integra y prueba de forma continúa para detectar errores lo antes posible.	Programador
Ejecutar pruebas de aceptación	Reporte de pruebas de aceptación	Verificar que la última versión de una iteración cumple los requisitos del cliente mediante pruebas de aceptación.	Cliente Encargado de pruebas

Fuente: Creado por autor

d. Puesta en Producción

Beck, K., Beedle, M. (2001) La etapa de liberación es la última fase del proceso de desarrollo de XP. En esta etapa, el equipo se centra en completar el producto y ponerlo en producción. La retroalimentación se acelera para garantizar la calidad. Las revisiones semanales y las reuniones diarias ayudan a detectar errores. El producto se libera, pero se puede mejorar con nuevas características. La puesta en producción de un sistema es un hito importante que debe celebrarse para reconocer el esfuerzo del equipo. XP promueve el desarrollo de software divertido y gratificante. Esto ayuda a mantener la motivación del equipo y a garantizar que el producto sea de alta calidad.

Tabla 5*Fase de puesta en producción*

TAREA	ARTEFACTO	TÉCNICA	RESPONSABLES
Realizar pruebas adicionales	Informe sobre pruebas adicionales y de rendimiento	Verificar que la última versión de la aplicación funciona correctamente.	Cliente

Fuente: Creado por autor.

e. Mantenimiento

(Beck, 2000, p. 135). El mantenimiento es la etapa final del ciclo de vida del software en la programación extrema. Se trata de un proceso continuo que implica la corrección de errores, la incorporación de nuevas características y la adaptación del sistema a los cambios en los requisitos del cliente. La actitud que se debe tomar en esta etapa es más conservadora que en las etapas anteriores. El objetivo es mantener el sistema funcionando correctamente, no experimentar con nuevas ideas.

2.3.10. Artefactos de la Programación Extrema

a. Historias de Usuario

Según Beck, K. (2000), Las historias de usuario son una técnica utilizada en XP para especificar los requisitos del software. Se describen en tarjetas de papel y pueden ser requisitos funcionales o no funcionales. Pueden ser modificadas o reemplazadas en cualquier momento. Las historias de usuario deben ser de tamaño manejable y fácil de entender para que los programadores puedan implementarlas en un plazo breve.

Según Wake (2000), Las historias de usuario son una forma de describir los requisitos de un sistema de software desde la perspectiva usuario. Son similares a los casos de uso, pero se centran en lo que Usuario registrado quiere hacer, en lugar de cómo lo quiere hacer. Las historias de usuario se utilizan para estimar el tiempo y los esfuerzos necesarios para desarrollar el sistema. También se utilizan para crear pruebas de aceptación, que verifican que el sistema satisfaga las necesidades usuario. Las historias de usuario se escriben en un formato sencillo y fácil de entender, y suelen consistir en tres frases. El cliente debe escribir las historias de usuario, ya que son las personas que mejor conocen sus necesidades. Los cambios en las historias de usuario son bienvenidos, ya que reflejan la evolución de las necesidades usuario.

En el desarrollo de software ágil, las historias de usuario son una forma de describir los requisitos del sistema desde la perspectiva usuario. Cada historia de usuario es una descripción breve de una función o característica del sistema. El análisis de las historias de usuario es un proceso continuo que se lleva a cabo a lo largo del desarrollo del software. Este análisis ayuda a garantizar que las historias de usuario sean claras y comprensibles tanto para el cliente como para los desarrolladores. Ron, R., Ann, A., & Chet, C. (2000).

Tabla 6*Plantilla historia de usuario*

HISTORIA DE USUARIO	
Número:	Nombre:
Usuario:	
Modificación de historia N°:	Iteración asignada:
Prioridad en negocio: (Alta/Media/Baja)	Riesgo en Desarrollo: (Alto/Medio/Bajo)
Descripción:	
Observaciones:	

Fuente: Priolo, A. (2007)

Según Cohn (2004). La descripción de una historia de usuario es un resumen claro y conciso de lo que un usuario final quiere lograr., desde su perspectiva. Se compone de tres partes: El rol usuario ¿Quién es la persona que necesita esta función? La función del sistema: ¿Qué puede hacer el sistema para ayudar al usuario? El valor de negocio: ¿Qué beneficio obtiene Usuario registrado al utilizar la función? una historia de usuario útil debe ser:

- **Independiente:** Una historia de usuario debe ser independiente de otras historias de usuario.
- **Negociable:** Las historias de usuario deben ser flexibles y abiertas a la negociación. Esto permite que el equipo de desarrollo y Usuario registrado trabajen juntos para garantizar que las historias de usuario satisfagan las necesidades de todos.
- **Valiosa:** Las historias de usuario deben proporcionar un valor claro para Usuario registrado y el negocio. Esto ayuda a garantizar que el desarrollo del software sea eficiente y eficaz.
- **Estimable:** La funcionalidad descrita debe ser lo suficientemente pequeña como para que un desarrollador pueda estimar su tiempo y esfuerzo de implementación.
- **Pequeña:** Las historias de usuario deben ser pequeñas y fáciles de entender. Esto facilita la comunicación entre el equipo de desarrollo y Usuario registrado.
- **Testeable:** Las funcionalidades deben ser probadas para garantizar que estén completas y funcionen correctamente.

b. Tareas de Ingeniería

Según Ron, A., Ann, A., & Chet, C. (2000). Son una herramienta utilizada en el desarrollo de software para describir el trabajo que se debe realizar para completar una historia de usuario. Estas tareas se especifican con una fecha de inicio y fin, el programador

responsable de realizarlas y una descripción de lo que se debe hacer. Las tareas de ingeniería son fundamentales para el desarrollo de software ágil, ya que ayudan a garantizar que el trabajo se complete de manera eficiente y eficaz. Ya que permiten planificar y estimar el trabajo de manera precisa. También ayudan a garantizar que el trabajo se complete a tiempo y dentro del presupuesto.

Las tareas de ingeniería pueden clasificarse según su objetivo, como diseñar, desarrollar, corregir o mejorar el software.

Tabla 7

Plantilla tareas de ingeniería

TAREA DE INGENIERÍA	
Número de tarea:	Número de historia de usuario:
Nombre de tarea:	
Tipo de tarea:	Puntos estimados:
Fecha de inicio:	Fecha fin:
Programador responsable:	
Descripción:	

Fuente: Priolo, 2007.

c. Tarjetas CRC

Según Castillo, A. (2010). Las tarjetas CRC se utilizan para representar objetos y sus responsabilidades. La clase a la que pertenece el objeto se escribe en la parte superior de la tarjeta. Las responsabilidades u objetivos del objeto se colocan en la columna de la izquierda, y las clases colaboradoras se colocan en la columna de la derecha. Las tarjetas CRC ayudan a los programadores a comprender y comunicar el comportamiento de los objetos, así como a identificar sus dependencias.

Tabla 8

Plantilla tarjeta CRC

TARJETA CRC	
Número:	Escenario:
Nombre CRC:	

Responsabilidades	Colaboradores	Atributos
Observaciones		

Fuente: Priolo, A. (2007)

d. Pruebas en XP

Según Marchesi et al. (2002) La programación extrema (XP) es una metodología ágil de desarrollo de software que se basa en los valores de comunicación, simplicidad, feedback, coraje y respeto. Estos valores se reflejan en los cuatro tipos de pruebas que se utilizan en los proyectos XP: Pruebas unitarias, Pruebas de integración, Pruebas de sistema, Pruebas de aceptación.

Pruebas de aceptación

Según Marchesi et al. (2002) son pruebas que realizan los clientes para verificar que un sistema cumple con sus requisitos. En el desarrollo ágil, las pruebas de aceptación se basan en las historias de usuario, que describen los requisitos del cliente desde su perspectiva. El cliente especifica los escenarios de prueba que se utilizarán para verificar que cada historia de usuario se ha implementado correctamente. Las pruebas de aceptación se consideran pruebas de caja negra, ya que no requieren conocimiento del código interno del sistema. Se realizan interactuando con el sistema a través de su interfaz de usuario o API.

El objetivo de las pruebas de aceptación es garantizar que el sistema sea funcional y cumpla con las expectativas del cliente. Marchesi et al. (2002) señalan que se realizan en cada iteración del desarrollo para verificar la funcionalidad que se implementa en esa iteración. Las pruebas de aceptación se consideran un éxito cuando todas las historias de usuario pasan las pruebas. Los resultados de las pruebas se deben publicar para que todo el equipo esté al tanto de su progreso.

Tabla 9

Plantilla pruebas de aceptación

PRUEBA DE ACEPTACIÓN	
Caso de prueba:	
Número de prueba:	Número historia de usuario:
Nombre de caso de prueba:	
Descripción:	
Condiciones de ejecución:	

Entradas:
Resultados esperados:
Evaluación:

Fuente: Priolo, A. (2007)

Pruebas Unitarias

Marchesi et al. (2002) encontraron que las pruebas unitarias son una técnica de prueba de software que se utiliza para garantizar que los módulos individuales de un programa funcionen correctamente. Las pruebas unitarias se centran en el código interno de un programa, lo que significa que tienen acceso al código fuente del módulo que se está probando. Las pruebas unitarias se centran en la validación de pequeñas unidades de funcionalidad, aislando el módulo de las demás partes del sistema. Las pruebas unitarias se suelen escribir utilizando un marco de pruebas, como Jest. Los marcos de pruebas pueden ser útiles para crear pruebas unitarias simples, pero no pueden generar pruebas para todos los casos de uso. La regla 80-20 se aplica a las pruebas unitarias, lo que significa que las pruebas que toman el 80% del tiempo de desarrollo suelen ser las más difíciles de generar. En general, las pruebas unitarias son una herramienta valiosa para garantizar la calidad del software.

Pruebas de Rendimiento

Las pruebas de rendimiento evalúan el comportamiento de un sistema bajo carga. Pueden ser realizadas por desarrolladores o por el equipo de calidad de software. Los objetivos de las pruebas de rendimiento incluyen: Medir la velocidad y la capacidad de respuesta del sistema. Evaluar cómo el sistema maneja una carga pesada. Identificar los cuellos de botella de rendimiento. Las pruebas de rendimiento proporcionan información valiosa sobre el desempeño del sistema, lo que ayuda a garantizar que el sistema cumpla con los requisitos usuario. (Marchesi et al., 2002).

Pruebas Manuales

Las pruebas manuales son pruebas de software que se realizan sin la ayuda de herramientas automatizadas que se utiliza para probar la interfaz gráfica de usuario (GUI) de un sistema. Las pruebas manuales suelen realizarse haciendo clic en los elementos de la GUI y verificando que el sistema responde correctamente. Las pruebas manuales pueden ser realizadas por profesionales de la calidad de software o por los usuarios finales. (Marchesi et al., 2002).

2.3.11. Sistema Gestor de Base de Datos

Un DBMS es una herramienta que ayuda a los usuarios a organizar y gestionar sus datos. Los DBMS proporcionan una serie de funciones para ayudar a los usuarios a crear, almacenar, acceder y actualizar sus datos. (Mannino, 2007).

Es una herramienta que ayuda a las personas a organizar y gestionar grandes cantidades de datos. Una base de datos es una colección de datos relacionados entre sí. El objetivo principal de un SGBD es proporcionar una forma de almacenar y recuperar datos de una base de datos de forma rápida y segura. (Silberschatz, Galvin, & Gagne, 2006).

Un sistema de administración de bases de datos (DBMS) es un software que ayuda a las organizaciones a organizar, gestionar y acceder a los datos de manera eficiente. El DBMS proporciona las siguientes funciones: almacenamiento de datos, acceso a datos, administración de datos. (Silberschatz, Korth, & Sudarshan, 2012, p. 2).

2.3.12. Metodología Ágil

Las metodologías de software son guías para crear software de forma efectiva. Las metodologías de software se basan en el desarrollo iterativo e incremental, lo que significa que el software se desarrolla en ciclos cortos y repetitivos. En cada ciclo, se recopilan requisitos, se diseña y se implementa una nueva funcionalidad, y se prueba el software. Los requisitos y las soluciones pueden evolucionar durante el proceso de desarrollo, según las necesidades del proyecto. Las metodologías de software pueden ayudar a los equipos de desarrollo a entregar software de alta calidad en tiempo y dentro del presupuesto. (Arreola, 2018).

Somerville (2020) afirma que los métodos ágiles de desarrollo de software se basan en la entrega de software de forma incremental e iterativa. En lugar de desarrollar todo el software de una sola vez, los métodos ágiles dividen el desarrollo en pequeñas fases, llamadas iteraciones. Cada iteración se centra en la implementación de una pequeña cantidad de características del producto. El desarrollo incremental permite a los equipos de desarrollo entregar software de forma rápida y frecuente al cliente. Esto permite al cliente obtener comentarios tempranos sobre el software y realizar cambios según sea necesario. Los métodos ágiles son adecuados para el desarrollo de software porque los productos de software suelen ser sistemas independientes. Esto significa que las características del producto pueden desarrollarse de forma independiente y en cualquier orden. Además, los equipos de desarrollo de software suelen estar co-ubicados. Esto facilita la comunicación informal entre los miembros del equipo y la colaboración en el desarrollo de las características del producto.

2.3.13. Lenguaje de Programación Multiparadigma

Permiten a los programadores combinar diferentes enfoques para resolver problemas. Cada paradigma tiene su propia visión del problema y su propio estilo de programación. Los problemas del mundo real suelen ser complejos y requieren un enfoque multidisciplinario. Por ello, los lenguajes de programación multiparadigma son cada vez más populares. (Hofstedt, 2011).

Dice Suehring, S. (2013), JavaScript y Java son dos lenguajes de programación diferentes. JavaScript es un lenguaje de scripting que se ejecuta en el navegador web, mientras que Java es un lenguaje de programación compilado que se puede ejecutar en cualquier plataforma. JavaScript es una implementación de la especificación ECMAScript. ECMAScript es una especificación técnica que define el lenguaje JavaScript.

Dice Freeman & Robson (2014) JavaScript es un lenguaje de programación único en el mundo de la programación. A diferencia de otros lenguajes de programación, JavaScript no requiere compilación ni enlace. En lugar de ello, el código JavaScript se escribe directamente en el navegador y se ejecuta de inmediato. Esto hace que JavaScript sea un lenguaje de programación muy flexible y fácil de usar.

2.3.14. Arquitectura de Software

Según Garlan y Shaw (1994), la arquitectura de software es el diseño global de un sistema de software. La arquitectura de software va más allá de los algoritmos y estructuras de datos, y se ocupa de cuestiones como: La organización de los componentes del sistema. Las relaciones entre los componentes. Los patrones de comunicación entre los componentes. La arquitectura de software es un nuevo tipo de problema porque requiere un enfoque holístico que considere el sistema en su conjunto.

La arquitectura de software es la estructura de un sistema, que define los componentes y las relaciones entre ellos. Los componentes son las partes individuales del sistema, y las relaciones son las interacciones entre ellos. La arquitectura de software proporciona una visión general del sistema, y ayuda a los desarrolladores a comprender cómo funciona y cómo se puede modificar. (Bass, Clemets y Kazman, 2003, p. 11).

CAPÍTULO III MATERIALES Y MÉTODOS

3.1. TIPO Y NIVEL DE LA INVESTIGACIÓN

3.1.1. Tipo de Investigación

La investigación observacional implica la observación sistemática y detallada de eventos, comportamientos o fenómenos en su entorno natural, sin intervención ni manipulación activa por parte del investigador (Fraenkel & Wallen, 2006). La investigación es observacional porque no se modifica la gestión de recursos ni las variables descriptivas. Solo se observa cómo se comportan estas variables en su estado natural, para levantar los datos que definan los requerimientos de la aplicación web.

Según Bastidas (2009), la investigación prospectiva "representa una habilidad para tomar decisiones relevantes que permiten determinar futuros posibles". Se ha recopilado información para definir los requisitos de las historias de usuario relacionadas con la gestión de recurso humano, financiero, material y tecnológico, mediante entrevistas.

La investigación transversal se realiza en un solo momento en el tiempo. En este tipo de investigación, los investigadores recopilan datos sobre las variables de interés de una muestra de población, sin manipular el entorno de estudio. (Hernández Sampieri, Fernández Collado, & Baptista Lucio, 2020, p. 265). En el estudio, se realizó la entrevista al responsable de ODAJUP y jueces de paz en un momento único, en relación con el recurso humano, financiero, material y tecnológico, mediante entrevistas en un solo momento.

Dankhe, N. (1986), Los estudios descriptivos buscan caracterizar un fenómeno en su estado actual. Para ello, recopilan datos sobre las propiedades o características del fenómeno, como su frecuencia, distribución, intensidad o relación con otras variables. El estudio es descriptivo porque se ha sometido a análisis los requerimientos de información para los jueces de paz.

3.1.2. Nivel de Investigación

En el nivel de investigación descriptivo se centra en la descripción de las características de un objeto o fenómeno. Para ello, selecciona las características fundamentales del objeto de estudio y las describe de forma detallada, identificando sus partes, categorías o clases. Así como, desarrollar prototipos (Bernal, 2010).

El estudio es descriptivo, ya que se desarrolló un prototipo de aplicación web para comprender mejor el proceso de gestión de recursos, mediante la automatización de las historias de usuario de los Juzgados de Paz del Distrito Judicial de Ayacucho.

3.2. DISEÑO DE LA INVESTIGACIÓN

Los diseños de investigación no experimental son aquellos en los que los investigadores no manipulan las variables de interés. En estos diseños, los investigadores observan las variables tal como se presentan en la realidad. (Hernández Sampieri, Fernández Collado, &

Baptista Lucio, 2020, p. 265). No se ha manipulado la variable de interés gestión de recursos, entonces la investigación es de nivel descriptivo.

El diseño de investigación prospectivo es un tipo de diseño de investigación observacional en el que los investigadores recopilan datos sobre una variable de interés antes de que se produzca un evento o condición. (Hernández Sampieri, Fernández Collado, & Baptista Lucio, 2020, p. 267). El estudio es de diseño prospectivo, porque se ha recopilado datos para las variables descriptivas del presente estudio.

El diseño de la investigación transversal es un tipo de diseño de investigación observacional en el que los investigadores recopilan datos sobre una variable de interés en un solo momento en el tiempo. (Hernández Sampieri, Fernández Collado, & Baptista Lucio, 2020, p. 266). El estudio es de diseño transversal porque se ha recopilado datos en un solo momento, mediante entrevistas.

3.3. POBLACIÓN Y MUESTRA

3.3.1. Población

"Es el conjunto de todos los individuos o elementos de un universo que tienen una característica común o características comunes" (Hernández, Fernández, & Baptista, 2020, p. 163). La población estuvo compuesta por cinco procesos de recursos humanos, cinco procesos de recursos financieros, tres procesos de recursos materiales, tres procesos de recurso tecnológicos, de los Juzgados de Paz del Distrito Judicial de Ayacucho el año 2023.

3.3.2. Muestra

La muestra es un subconjunto de la población que se utiliza para representar a toda la población en el estudio. El muestreo es un proceso estadístico que permite seleccionar un grupo de personas o cosas que representen a una población más grande (Hernández, Fernández, & Baptista, 2020, p. 155).

El censo es un procedimiento estadístico que consiste en recopilar información sobre todas las unidades de una población. Es el método más completo para obtener información sobre una población, ya que permite conocer sus características de manera detallada y precisa (Hernández, Fernández, & Baptista, 2020, p. 156).

No existe muestra, fue censo, porque se estudió; cinco procesos de recursos humanos, cinco procesos de recursos financieros, tres procesos de recursos materiales, tres procesos de recurso tecnológicos, de los Juzgados de Paz del Distrito Judicial de Ayacucho el año 2023.

3.4. VARIABLES

3.4.1. Definición Conceptual de Variables

Variables de Interés

Gestión de Recursos: se refiere al proceso de administración, planificación y uso de recursos disponibles en los juzgados de paz.

Variables descriptivas

Recurso Humano: se refiere a la terna de jueces de paz que se asigna a cada juzgado de paz.

Recurso Financiero: se refiere a las tasas y aranceles que se pueden recibir al emitir

certificados notariales en los juzgados de paz.

Recurso Material: se refiere al mobiliario asignado al juzgado de paz.

Recurso Tecnológico: se refiere a los equipos de cómputo asignados al juzgado de paz.

3.4.2. Definición Operacional de Variables

Variables descriptivas

Recurso humano: Son las ternas de jueces de paz.

Recurso financiero: Son documentos aranceles y certificados que tienen costos.

Recurso material: Son mobiliarios asignados a los juzgados de paz.

Recurso tecnológico: Son equipos de procesamiento de datos asignados a los juzgados de paz.

3.5. OPERACIONALIZACIÓN DE VARIABLES

Tabla 10

Operacionalización de variables

VARIABLE	DIMENSIÓN	INDICADOR	PREGUNTA
Gestión de recursos	Recurso humano	Terna de jueces de paz	¿Qué documento formaliza la designación de jueces de paz?
			¿Cómo identifica usted que un juzgado de paz tiene terna de jueces asignadas?
			¿Cómo identifica usted que un juzgado de paz no tiene terna de jueces asignadas?
			¿Cómo identifica usted que juzgados de paz tienen terna de jueces asignadas próximas a vencer?
			¿Con cuánto tiempo de anticipación se debe realizar la convocatoria de una nueva terna jueces de paz?
	Recurso material	Mobiliario de Juzgado de paz	¿Cómo sabe usted con qué mobiliarios cuenta el juzgado de paz?
			¿Qué tipo de mobiliario son útiles para el buen funcionamiento del juzgado de paz?

			¿Qué características es necesario saber para identificar el bien patrimonial del juzgado de paz?
	Recurso tecnológico	Equipo de procesamiento de datos	¿Cómo sabe usted con qué equipo de procesamiento de datos cuenta el juzgado de paz?
			¿Qué tipo de equipo de procesamiento de datos son útiles para el buen funcionamiento del juzgado de paz?
			¿Qué características es necesario saber para identificar el equipo de procesamiento de datos del juzgado de paz?
	Recurso financiero	Aranceles y certificados	¿Cómo obtiene el juez de paz los aranceles judiciales?
			¿Cómo será el formato predefinido para la emisión de certificados?
			¿Cómo verifica qué certificados se emitieron en el juzgado de paz?
			¿Qué elementos visuales ayudan al juez de paz a prevenir errores en la emisión de certificados?
			¿Cuáles son los formatos más usados para la emisión de certificados?
			¿Cómo obtiene los aranceles judiciales para la emisión de certificados?

Fuente: Creado por el autor.

3.6. TÉCNICAS E INSTRUMENTOS DE INVESTIGACIÓN

3.6.1. Técnicas

"La entrevista es una técnica de recolección de datos en la que el investigador establece un diálogo con el entrevistado para obtener información sobre un tema determinado" (Hernández Sampieri et al., 2020, p. 200). Se aplicó la técnica de entrevista al responsable de la Oficina de apoyo a la justicia de paz y personal administrativo usuario de la aplicación web, para obtener información sobre las historias de usuario para automatizarla.

3.6.2. Instrumentos

"La entrevista guía para su planificación es un método de investigación que consiste en una conversación entre el entrevistador y el entrevistado, con el fin de obtener información sobre un tema determinado" (Cruz, 2012, p. 11). Se aplicó el instrumento, guía de entrevista, al responsable de la Oficina de apoyo a la justicia de paz y, otra guía de entrevista para el personal administrativo usuario de la aplicación web, para recolectar información necesaria sobre las historias de usuario, para la gestión de los recursos de los Juzgados de paz del distrito judicial de Ayacucho; el instrumento se muestra en el anexo 2.

3.6.3. Validez del Instrumento

La validez de un instrumento es la capacidad de medir lo que pretende medir. Se evalúa mediante la suma de la validez de contenido y la validez de criterio. La validez de contenido se refiere a la adecuación de los ítems del instrumento a las variables que se pretenden medir. Se puede evaluar mediante juicio de expertos, que valorarán si los ítems son relevantes, representativos y comprensibles. Hernández, R., Fernández, C., & Baptista, L. (2020)

La validez de contenido de un instrumento se evalúa mediante la opinión de expertos en la materia, quienes analizan si las preguntas formuladas abarcan todas las dimensiones de las variables descriptivas recurso humano, recurso financiero, recurso material, recurso tecnológico, se representa los instrumentos en Anexo 3.

3.6.4. Confiabilidad del Instrumento

No se determina la confiabilidad del instrumento porque es una guía de entrevista abierta.

3.7. PROCEDIMIENTOS

3.7.1. Técnicas de Procesamiento de Datos

El procesamiento de datos es el proceso de convertir los datos brutos en información útil. Se utiliza una variedad de técnicas, como la programación extrema, para organizar, analizar y comprender los datos. (Hernández Sampieri et al. 2020).

Tabla 11

Técnicas de procesamiento de datos

FASE	ARTEFACTO	TÉCNICA
Fase de exploración	historial de usuario	Como [Usuario], quiero [Acción], [Resultado] (Regla) - Divide historias grandes en reglas separadas.
	Arquitectura técnica inicial	Configurar la arquitectura técnica desde cero.

	Plan general	Estimar esfuerzo de historia de usuario
Planeación	Historias de Usuario por prioridad	Seleccionar historias de usuario que aporten mayor valor al negocio.
	Estimación de esfuerzo	Estimar esfuerzo (semanas) por historia de usuario, considerando planificación, diseño, implementación y pruebas.
	Estrategia de lanzamiento	Planificar el cronograma del plan de entrega.
Fase de iteración	la estructura física y lógica de un sistema informático, Actividades de diseño, construcción y mantenimiento de sistemas y estructuras, Plan de trabajo para un ciclo de desarrollo, Pruebas de funcionamiento, GUI, tarjetas CRC, base de datos física, Código que representa clase entidades, pruebas unitarias de código, código para tarea de ingeniería, Resultados de la verificación del comportamiento del código,	Adaptar arquitectura a negocio Considerar negocio en diseño Organizar sistema por capas Integrar frameworks dividir historias en tareas Especificar tareas con reglas del negocio Calcular esfuerzo por tarea Asignar tareas a programadores Utilizar el plan de versión Actualizar el plan con tareas de ingeniería de la siguiente iteración Corregir errores de pruebas Agregar tareas pendientes Revisar y actualizar la información de las tarjetas de tarea de ingeniería Crear interfaz personalizada para cada usuario Generar código de interfaz con herramienta Simplificar diseño de tareas Rediseñar una tarea debido a que no

	Pruebas de integración y de aceptación registradas.	cumple con los requisitos Determinar quién es responsable de qué Desarrollar trabajo conjunto Describir características de clases Generar script con tarjeta CRC Ejecutar script con base de datos Generar código fuente automáticamente a partir de tarjetas CRC Generar prueba unitaria Desarrollar tareas de ingeniería Mejorar código Asignar tareas a programadores Probar código Corregir errores de código Unir tareas de historia de usuario Actualizar el sistema para obtener las últimas correcciones y mejoras. Integrar a medida que se completan las tareas Comprobar que los componentes funcionan juntos Probar la última versión de un desarrollo Probar la funcionalidad del software
--	---	--

Fuente: Creado por el autor.

3.7.2. Herramientas para el Procesamiento de Datos

"Las herramientas de procesamiento de datos son software o programas informáticos que automatizan las tareas de procesamiento de datos. Estas tareas pueden incluir la recopilación, el almacenamiento, la limpieza, el análisis y la visualización de datos." (Hernández Sampieri et al., 2020, p. 267). Instrumentos y software para transformar datos en información.

Tabla 12*Herramientas para el procesamiento de datos*

SOFTWARE	FABRICANTE	SERVICIO
CSS 3	Hakon Wium Lie	Es un lenguaje de diseño gráfico que permite definir y crear la presentación de un documento estructurado escrito en un lenguaje de marcado.
DOCKER	Apache License 2.0	Docker es un software que permite empaquetar y ejecutar aplicaciones de forma aislada en contenedores, lo que facilita su despliegue y administración en diferentes sistemas operativos
ESLINT	Nicholas C. Zakas	Un depurador de JavaScript es un programa que ayuda a los desarrolladores a rastrear y solucionar problemas en su código JavaScript.
GIT	Software Libre	Control de versiones diseñado para aplicaciones con gran cantidad de archivos fuente.
GITHUB	Ruby on Rails	El servicio se utiliza principalmente para desarrollar software, pero también se puede utilizar para alojar documentos, sitios web y obras de arte digital.
HTML 5	Tim Berners Lee	HTML5 es la última versión del lenguaje de marcado básico de la web. Establece dos variantes de sintaxis para HTML.
JAVASCRIPT	Brendan Eich	JavaScript es un lenguaje de programación que se traduce a código máquina en el momento de la ejecución. Esto significa que los programas escritos en JavaScript no necesitan ser compilados para ejecutarse. En cambio, se pueden ejecutar directamente por un intérprete.
NODEJS	Ryan Dahl	Node.js es un entorno de desarrollo para crear aplicaciones web escalables, basado en el lenguaje de programación JavaScript.
PRETTIER	Open Source	Formatea el código de forma estricta y coherente, siguiendo las reglas de estilo de una variedad de lenguajes y formatos.
REACT	Facebook, Inc. y Jordan Walke	Es un conjunto de herramientas y funciones que se pueden utilizar para crear interfaces de usuario en Javascript.
VISUAL	MIT License	Un editor de código fuente es una

STUDIO CODE		herramienta que permite a los desarrolladores escribir, editar y depurar código informático. Visual Studio Code es un editor de código fuente desarrollado por Microsoft para Windows, Linux y macOS.
NPM	Isaac Z. Schlueter.	NPM es un sistema de gestión de paquetes para Node.js. Permite a los desarrolladores instalar y administrar paquetes de código JavaScript.
PostgreSQL	Open Source	Es un sistema de gestión de bases de datos relacional (SGBDR) de código abierto y gratuito.

Fuente: Creado por el autor.

3.7.3. Técnicas de Análisis de Datos

Las técnicas de análisis de datos son las herramientas que permiten transformar los datos brutos en información útil. Estas técnicas se utilizan para extraer significado de los datos, identificar patrones y tendencias, y generar conclusiones. (Hernández Sampieri et al., 2020, p. 268). Los datos obtenidos mediante la técnica de entrevista se analizaron y se procesaron mediante la programación extrema, para desarrollar la aplicación web, que apoye la gestión de recursos en los Juzgados de Paz del Distrito Judicial de Ayacucho.

CAPÍTULO IV ANÁLISIS Y RESULTADOS DE LA INVESTIGACIÓN

4.1. RESULTADOS

4.1.1. Artefactos del Software Aplicando el Proceso XP

a. Fase de Exploración

Siguiendo las instrucciones de la tabla 2 En la fase de exploración del proceso XP, descrito en el capítulo II, se recopilan las historias de usuario, se diseña la arquitectura técnica inicial y se crea un plan de alto nivel.

Tabla 13

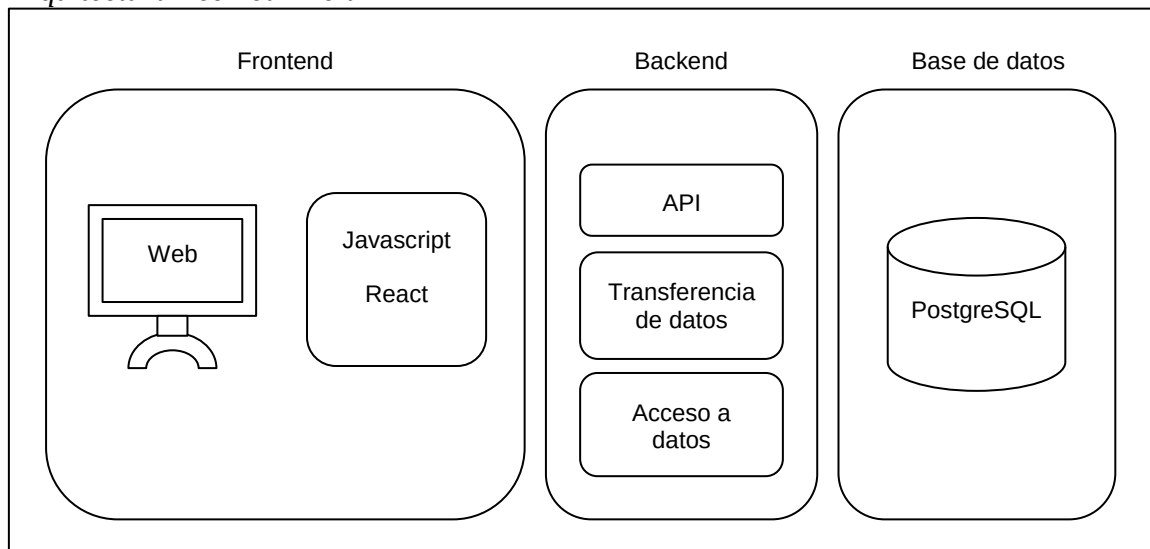
Historias de Usuario

Nº	HISTORIA DE USUARIO	DESCRIPCIÓN
1	Administrar Usuarios del Sistema	Usuario registrado Responsable de ODAJUP , registra, lista, actualiza y da de baja a los Usuarios del Sistema.
2	Administrar Juez de Paz	Usuario registrado Responsable de ODAJUP , registra, busca, actualizar y habilita o deshabilita al Juez de Paz
3	Administrar Juzgado de Paz	Usuario registrado Responsable de ODAJUP , registra, busca, actualiza el Juzgado de Paz.
4	Administrar Recurso Material y Tecnológico	Usuario registrado Responsable de ODAJUP registra, lista, actualiza los Recurso Material y Tecnológico
5	Emitir Certificado Escritura pública de Transferencia de bienes inmuebles no registrables.	Usuario registrado Juez de Paz lista, selecciona ingresa los datos en el Certificado Escritura pública de Transferencia de bienes inmuebles no registrables.
6	Asignar terna de jueces a juzgado	Usuario registrado Responsable de ODAJUP Busca, registra la asigna a los Jueces de paz a los Juzgados.
7	Consultar Juzgados con ternas próximos a vencer	Usuario registrado Responsable de ODAJUP puede visualizar los Juzgados de Paz que tienen terna asignada, así como las que no están asignadas. Además puede listar los juzgados que tienen ternas próximas a vencer en un determinado número de días que podrá asignar.

8	Iniciar sesión de jueces	Usuario registrado Juez de paz inicia sesión con su nombre de usuario y clave con la cual accederá al sistema.
9	Iniciar sesión de administrador	Usuario registrado Responsable de ODAJUP inicia sesión con su nombre de usuario y clave con la cual acceder al sistema.
10	Administrar cuadro de aranceles	Usuario registrado Responsable de ODAJUP registra, lista, actualiza los aranceles judiciales.
11	Consultar Usuarios del Sistema	Usuario registrado Responsable de ODAJUP busca por nombre y/o apellido y visualiza la lista de usuarios del sistema.
12	Consultar Jueces de Paz	Usuario registrado Responsable de ODAJUP busca por nombre y/o apellido y visualiza la lista de los Jueces de Paz.
13	Consultar Juzgados de Paz	Usuario registrado Responsable de ODAJUP busca por nombre y visualiza la lista de los Juzgados de Paz.
14	Consultar Recursos Materiales y Tecnológicos	Usuario registrado Responsable de ODAJUP busca por nombre y/o código patrimonial y visualiza la lista de los Materiales y Tecnológicos.
15	Consultar Juzgados con y sin ternas Asignadas	Usuario registrado Responsable de ODAJUP busca por nombre de Juzgado y visualiza la lista de los Juzgados de Paz con o sin ternas asignadas.
16	Consultar Cuadro de Aranceles	Usuario registrado Responsable de ODAJUP busca por nombre por concepto y visualiza la lista de aranceles judiciales.

Fuente: Creado por el autor.

Figura 6
Arquitectura Técnica inicial



Fuente: Creado por el autor.

Tabla 14
Plan de Alto Nivel

No	HISTORIA DE USUARIO	ESFUERZO (DÍAS)
1	Administrar Usuarios del Sistema	4
2	Administrar Juez de Paz	4
3	Administrar Juzgado de Paz	4
4	Administrar Recursos Materiales y tecnológicos	4
5	Emitir constancias de transferencia Posesoria	5
6	Asignar terna de jueces a juzgado	5
7	consultar Juzgados con ternas próximos a vencer	3
8	Iniciar sesión de jueces	3
9	Iniciar sesión de administrador	3

10	Administrar Cuadro de Aranceles	3
11	Consultar Usuario del Sistema	2
12	Consultar Jueces de Paz	2
13	Consultar Juzgados de Paz	2
14	Consultar Recursos Materiales y Tecnológicos	2
15	Consultar Juzgados con y sin ternas Asignadas	3
16	Consultar Cuadro de Aranceles	2

Fuente: Creado por el autor.

b. Fase de Planificación

Implementamos la técnica en la fase de planificación, para el proceso XP, según la teoría del capítulo II, sección 2.3.9, tabla 3, obteniendo los artefactos; historias de usuario según su importancia, estimar el esfuerzo necesario para completarlas y establecer un plan para revisarlas (primera iteración).

Tabla 15

Historia usuario: administrar usuarios del sistema.

Historia de usuario	
Número: 1	Usuario: Responsable de ODAJUP
Nombre de historia: Administrar Usuarios de Sistema	
Prioridad en negocio: Medio	Riesgo en desarrollo: Medio
Puntos estimados: 1	iteración asignada: 2
Programador responsable: Max Auqui Bautista	
Descripción: Usuario registrado Responsable de ODAJUP pulsa un clic en el “modulo de usuario” el sistema automáticamente le muestra la interfaz “administración de usuarios”, y pulsa un clic en la opción “nuevo” el sistema presenta la interfaz de “registrar cuenta de usuario” donde se agrega información como son datos personales y los guarda con el botón “guardar”, el sistema revisa que los campos cumplan con los requisitos y los registra en la base de datos de nuevo registro de usuario acompañando un	

mensaje de confirmación de registro. Usuario registrado ingresa a la interfaz “administración de usuarios” e ingresa la inicial del nombre o apellido en la barra de búsqueda y el sistema listará los usuarios registrados con esa inicial ingresada, Usuario registrado selecciona un usuario del listado y pulsa un clic en “editar” el sistema le exhibe la interfaz “actualizar cuenta de usuario” con los datos extraídos desde la base de datos donde se podrá modificar todo los campos, actualiza y se puede cambiar el estado del usuario de activado a desactivado.

Observaciones: No hay ninguno

Fuente: Creado por el autor.

Tabla 16

Historia usuario. Administrar Juez de Paz

Historia de usuario	
Número: 2	Usuario: Responsable de ODAJUP
Nombre de historia: Administrar Juez de Paz	
Prioridad en negocio: Alto	Riesgo en desarrollo: Alto
Puntos estimados: 1	iteración asignada: 2
Programador responsable: Max Auqui Bautista	
Descripción: Usuario registrado Responsable de ODAJUP pulsa un clic en “módulo juez de paz” usuario registrado pulsa un clic en la opción “nuevo” el sistema presenta la interfaz de “registrar juez de paz” usuario registrado agrega información como son datos personales y los guarda con el botón “guardar”, el sistema revisa que los campos cumplan con los requisitos y los registra en la base de datos de nuevo registro del juez de paz acompañando un mensaje de confirmación de registro. el sistema automáticamente le mostrará la interfaz “administración de juez de paz” en la barra de búsqueda ingresa la inicial del nombre o apellido y el sistema listará los usuarios registrados con esa inicial ingresada, usuario registrado selecciona un usuario del listado y pulsa un clic en “editar” el sistema le exhibe la interfaz “actualizar juez de paz” con los datos extraídos desde la base de datos donde podrá modificar todo los campos, actualiza los datos necesarios, usuario registrado para deshabilitar o habilitar a un juez de paz.	
Observaciones: No hay ninguno	

Fuente: Creado por el autor.

Tabla 17*Historia usuario: Administrar Juzgado de Paz.*

Historia de usuario	
Número: 3	Usuario: Responsable de ODAJUP
Nombre de historia: Administrar Juzgado de Paz	
Prioridad en negocio: Alto	Riesgo en desarrollo: Alto
Puntos estimados: 1	iteración asignada: 2
Programador responsable: Max Auqui Bautista	
Descripción: Usuario registrado Responsable de ODAJUP pulsa un clic en “módulo juzgado de paz” usuario registrado pulsa un clic en la opción “nuevo” el sistema presenta la interfaz de “registrar juzgado de paz” usuario registrado agrega información como son datos del juzgado y los guarda con el botón “guardar”, el sistema revisa que los campos cumplan con los requisitos y los registra en la base de datos de nuevo registro del juzgado de paz acompañando un mensaje de confirmación de registro. El sistema le muestra automáticamente la interfaz de “administración de juzgado de paz” en la barra de búsqueda ingresa la inicial del nombre del juzgado y el sistema listará los juzgados registrados con esa inicial ingresada, usuario registrado selecciona un juzgado del listado y pulsa un clic en “editar” el sistema le exhibe la interfaz “actualizar juzgado de paz” con los datos extraídos desde la base de datos donde podrá modificar todo los campos, actualiza los datos necesarios.	
Observaciones: Ninguna	

Fuente: Creado por el autor.

Tabla 18*Historia usuario. Administrar Recursos Materiales*

Historia de usuario	
Número: 4	Usuario: Responsable de ODAJUP
Nombre de historia: Administrar Recursos Materiales	
Prioridad en negocio: Medio	Riesgo en desarrollo: Alto
Puntos estimados: 1	iteración asignada: 2
Programador responsable: Max Auqui Bautista	

Descripción: Usuario registrado **Responsable de ODAJUP** pulsa un clic en “módulo gestión de recursos materiales” y pulsa un clic en la opción “nuevo” el sistema presenta la interfaz de “administración de recursos materiales y tecnológicos” usuario registrado agrega información del juzgado como son datos del bien material y los guarda con el botón “guardar”, el sistema revisa que los campos cumplan con los requisitos y los registra en la base de datos de nuevo registro del recurso material y tecnológico acompañando un mensaje de confirmación de registro. usuario registrado ingresa al “administración de recursos materiales y tecnológicos” en la barra de búsqueda ingresa la inicial del nombre del recurso material y el sistema listará los recursos materiales y tecnológicos que tenga el juzgado registrados con esa inicial ingresada, usuario registrado selecciona un juzgado del listado y pulsa un clic en “editar” el sistema le exhibe la interfaz “actualizar recurso material y tecnológico” con los datos extraídos desde la base de datos donde podrá modificar todo los campos, actualiza los datos necesarios.

Observaciones: Ninguna

Fuente: Creado por el autor.

Tabla 19

Historia usuario. Emitir Constancia de Transferencia Posesoria.

Historia de usuario	
Número: 5	Usuario: Juez de Paz
Nombre de historia: Emitir Constancia de Transferencia Posesoria.	
Prioridad en negocio: Medio	Riesgo en desarrollo: Medio
Puntos estimados: 1	iteración asignada: 2
Programador responsable: Max Auqui Bautista	
Descripción: Usuario registrado Juez de Paz ingresa al “módulo emisión de certificados” y pulsa un clic en la el botón “nuevo” el sistema presenta la interfaz de “módulo emisión de certificado”, donde tendrá que elegir el tipo de certificado a generar, donde tendrá que llenar todo los datos requeridos, una vez concluido usuario registrado pulsa un clic en el botón de “generar certificado” el sistema automáticamente genera el pdf y guarda el archivo en la base de datos, el sistema presenta automáticamente la interfaz de “administración emisión de certificados”.	
Observaciones: Ninguna	

Fuente: Creado por el autor.

Tabla 20*Historia usuario. Asignar terna de jueces a juzgado.*

Historia de usuario	
Número: 6	Usuario: Responsable de ODAJUP
Nombre de historia: Asignar terna de jueces a juzgado	
Prioridad en negocio: Alto	Riesgo en desarrollo: Alto
Puntos estimados: 1	iteración asignada: 3
Programador responsable: Max Auqui Bautista	
Descripción: Usuario registrado Responsable de ODAJUP pulsa un clic en “módulo asignación de terna” y pulsa un clic en el botón “nuevo” el sistema presenta la interfaz de “módulo asignación de jueces de paz” usuario registrado agrega información y pulsa un clic en el botón “guardar”, el sistema revisa que los campos cumplan con los requisitos y los registra en la base de datos de nuevo registro de la asignación de la terna acompañando un mensaje de confirmación de registro. Usuario registrado ingresa al “asignación de jueces de paz” en la barra de búsqueda ingresa la inicial del nombre del juzgado y el sistema listará los juzgados de paz y la terna asignada que tenga el juzgado registrado.	
Observaciones: Ninguna	

Fuente: Creado por el autor.

Tabla 21*Historia usuario. Consultar Juzgados con ternas próximos a vencer*

Historia de usuario	
Número: 7	Usuario: Responsable de ODAJUP
Nombre de historia: Consultar Juzgados con Ternas próximos a vencer	
Prioridad en negocio: Alto	Riesgo en desarrollo: Alto
Puntos estimados: 1	iteración asignada: 2
Programador responsable: Max Auqui Bautista	

Descripción: Usuario registrado Responsable de ODAJUP pulsa un clic en el “módulo asignación de terna” usuario registrado digita la cantidad de días a vencer la terna y pulsa un clic en el botón “buscar por días por vencer” el sistema te listará los juzgados con sus respectivas ternas, fechas a finalizar, también se puede buscar los juzgados por ternas asignadas y no asignadas.
Observaciones: No hay ninguno

Fuente: Creado por el autor.

Tabla 22

Historia usuario. Iniciar sesión de jueces

Historia de usuario	
Número: 8	Usuario: Juez de Paz
Nombre de historia: Iniciar sesión de jueces	
Prioridad en negocio: Bajo	Riesgo en desarrollo: Bajo
Puntos estimados: 1	iteración asignada: 1
Programador responsable: Max Auqui Bautista	
Descripción: Usuario registrado Juez de paz abre el programa: el sistema presenta la página de “inicio de sesión”, usuario registrado accede con su nombre de usuario y clave, el sistema presenta la página principal cuando los datos ingresados son autenticos, si no se proporcionan los datos correctos, el sistema volverá a solicitarlos y mostrará los mensajes de error pertinentes.	
Observaciones: No hay ninguno	

Fuente: Creado por el autor.

Tabla 23*Historia usuario. Iniciar sesión de administrador.*

Historia de usuario	
Número: 9	Usuario: Responsable de ODAJUP
Nombre de historia: Iniciar Administrador de Sistema	
Prioridad en negocio: Bajo	Riesgo en desarrollo: Bajo
Puntos estimados: 1	iteración asignada: 1
Programador responsable: Max Auqui Bautista	
Descripción: Usuario registrado Responsable de ODAJUP abre el programa: el sistema presenta la página de “inicia sesión”, usuario registrado accede con su nombre de usuario y clave, el sistema presenta la página principal cuando los datos ingresados son auténticos, si no se proporcionan los datos correctos, el sistema volverá a solicitarlos y mostrará los mensajes de error pertinentes.	
Observaciones: No hay ninguno	

Fuente: Creado por el autor.

Tabla 24*Historia usuario. Administrar cuadro de aranceles.*

Historia de usuario	
Número: 10	Usuario: Responsable de ODAJUP
Nombre de historia: Gestión de aranceles	
Prioridad en negocio: Alto	Riesgo en desarrollo: Alto
Puntos estimados: 1	iteración asignada: 2
Programador responsable: Max Auqui Bautista	
Descripción: Usuario registrado Responsable de ODAJUP pulsa un clic en “módulo cuadro de aranceles” y pulsa un clic en la opción “nuevo” el sistema presenta la interfaz de “registro de aranceles” usuario registrado agrega información del concepto y costo del arancel y los guarda con el botón “guardar”, el sistema revisa que los campos cumplan con los requisitos y los registra en la base de datos de nuevo registro de usuario acompañando un mensaje de confirmación de registro. Usuario registrado ingresa a la interfaz inicial “administración de cuadro de aranceles” e ingresa la inicial del “concepto” en la barra de búsqueda y el sistema listará los aranceles registrados con esa inicial, usuario registrado selecciona un arancel del listado y pulsa un clic en “editar” el	

sistema le exhibe la interfaz “actualizar aranceles” con los datos extraídos desde la base de datos donde se podrá modificar todo los campos.
Observaciones: No hay ninguno

Fuente: Creado por el autor.

Tabla 25

Historia usuario. Consultar usuarios del sistema.

Historia de usuario	
Número: 11	Usuario: Responsable de ODAJUP
Nombre de historia: Consultar Usuarios del Sistema	
Prioridad en negocio: Medio	Riesgo en desarrollo: Alto
Puntos estimados: 1	iteración asignada: 1
Programador responsable: Max Auqui Bautista	
Descripción: Usuario registrado Responsable de ODAJUP pulsa un clic en “modulo usuarios” e ingresa a la interfaz “administración de usuarios” en la barra de búsqueda ingresa inicial de nombre, apellido y el sistema automáticamente le muestra el listado de personas quienes tienen esa letra en su nombre.	
Observaciones: No hay ninguno	

Fuente: Creado por el autor.

Tabla 26

Historia usuario. Consultar usuarios del sistema.

Historia de usuario	
Número: 12	Usuario: Responsable de ODAJUP
Nombre de historia: Consultar Jueces de Paz	
Prioridad en negocio: Medio	Riesgo en desarrollo: Alto
Puntos estimados: 1	iteración asignada: 1

Programador responsable: Max Auqui Bautista
Descripción: Usuario registrado Responsable de ODAJUP pulsa un clic en el “módulo juez de paz” el sistema le exhibe la interfazde “administración de juez de paz” en la barra de búsqueda ingresa inicial de nombre, apellido y el sistema automáticamente le muestra el listado de personas quienes tienen esa letra en su nombre.
Observaciones: No hay ninguno

Fuente: Creado por el autor.

Tabla 27

Historia usuario. Consultar juzgados de paz.

Historia de usuario	
Número: 13	Usuario: Responsable de ODAJUP
Nombre de historia: Consultar Juzgados de Paz	
Prioridad en negocio: Medio	Riesgo en desarrollo: Alto
Puntos estimados: 1	iteración asignada: 1
Programador responsable: Max Auqui Bautista	
Descripción: Usuario registrado Responsable de ODAJUP pulsa un clic en el “módulo juzgado de paz” el sistema le muestra el “administración de juzgado de paz” en la barra de búsqueda ingresa inicial de nombre del juzgado y el sistema automáticamente le muestra el listado de juzgados que tienen esa letra en su denominación.	
Observaciones: No hay ninguno	

Fuente: Creado por el autor.

Tabla 28*Historia usuario. Consultar recursos materiales*

Historia de usuario	
Número: 14	Usuario: Responsable de ODAJUP
Nombre de historia: Consultar recursos materiales	
Prioridad en negocio: Medio	Riesgo en desarrollo: Alto
Puntos estimados: 1	iteración asignada: 1
Programador responsable: Max Auqui Bautista	
Descripción: Usuario registrado Responsable de ODAJUP pulsa un clic en el “módulo recurso material y tecnológico” el sistema le muestra el “administración de recursos materiales y tecnológicos” en la barra de búsqueda ingresa inicial de nombre de recurso material y tecnológico y el sistema automáticamente le muestra el listado de bienes que tienen esa letra en su denominación.	
Observaciones: No hay ninguno	

Fuente: Creado por el autor.

Tabla 29*Historia usuario. Consultar juzgados con y sin ternas asignadas.*

Historia de usuario	
Número: 15	Usuario: Responsable de ODAJUP
Nombre de historia: Consultar Juzgados con y sin ternas Asignadas	
Prioridad en negocio: Medio	Riesgo en desarrollo: Alto
Puntos estimados: 1	iteración asignada: 1
Programador responsable: Max Auqui Bautista	
Descripción: Usuario registrado Responsable de ODAJUP ingresa al “módulo asignación de terna” El sistema presenta la interfaz de “asignación de jueces de paz” selecciona los botones “asignados o no asignados” ingresa una letras que contiene la denominación del juzgado y presiona la opción del botón “buscar”, el sistema automáticamente le muestra la lista de juzgados que tienen ternas “asignadas o no asignadas” que tienen esa letra en su denominación, según lo seleccionado.	

Observaciones: No hay ninguno

Fuente: Creado por el autor.

Tabla 30

Historia usuario. Consultar cuadro de aranceles.

Historia de usuario	
Número: 16	Usuario: Responsable de ODAJUP
Nombre de historia: Consultar Cuadro de Aranceles	
Prioridad en negocio: Medio	Riesgo en desarrollo: Alto
Puntos estimados: 1	iteración asignada: 1
Programador responsable: Max Auqui Bautista	
Descripción: Usuario registrado Responsable de ODAJUP pulsa un clic en “módulo cuadro aranceles” y el sistema le muestra “administración de cuadro de aranceles” en la barra de búsqueda ingresar la inicial de nombre de concepto del arancel y el sistema automáticamente le muestra la lista de aranceles que tienen esa letra en su denominación.	
Observaciones: No hay ninguno	

Fuente: Creado por el autor.

Tabla 31

Historia usuario. Administrar Recursos Tecnológicos

Historia de usuario	
Número: 17	Usuario: Responsable de ODAJUP
Nombre de historia: Administrar Recursos Tecnológicos	
Prioridad en negocio: Medio	Riesgo en desarrollo: Alto
Puntos estimados: 1	iteración asignada: 2
Programador responsable: Max Auqui Bautista	
Descripción: Usuario registrado Responsable de ODAJUP pulsa un clic en “módulo gestión de recursos materiales” y pulsa un clic en la opción “nuevo” el sistema presenta la interfaz de “administración de recursos materiales y tecnológicos” usuario registrado	

agrega información del juzgado como son datos del bien material y los guarda con el botón “guardar”, el sistema revisa que los campos cumplan con los requisitos y los registra en la base de datos de nuevo registro del recurso material y tecnológico acompañando un mensaje de confirmación de registro.

usuario registrado ingresa al “administración de recursos materiales y tecnológicos” en la barra de búsqueda ingresa la inicial del nombre del recurso material y el sistema listará los recursos materiales y tecnológicos que tenga el juzgado registrados con esa inicial ingresada, usuario registrado selecciona un juzgado del listado y pulsa un clic en “editar” el sistema le exhibe la interfaz “actualizar recurso material y tecnológico” con los datos extraídos desde la base de datos donde podrá modificar todo los campos, actualiza los datos necesarios.

Observaciones: Ninguna

Fuente: Creado por el autor.

Tabla 32

Plan de versión

Nº	HISTORIA DE USUARIO	PRIORIDAD	RIESGO	ESFUERZO (DÍAS)	ITERACIÓN
1	Administrar Usuario del Sistema	Medio	Medio	4	2
2	Administrar Juez de Paz	Alto	Alto	4	2
3	Administrar Juzgado de Paz	Alto	Alto	4	2
4	Administrar Recursos Materiales y Tecnológicos	Medio	Alto	4	2
5	Emitir constancias de transferencia Posesoria	Medio	Medio	5	2
6	Asignar terna de jueces a juzgado	Alto	Alto	5	2
7	Consultar Juzgados con ternas próximos a vencer	Alto	Alto	3	2

8	Iniciar sesión de jueces	Bajo	Bajo	3	1
9	Iniciar sesión de administrador	Bajo	Bajo	3	1
10	Administrar Cuadro de Aranceles	Alto	Alto	3	2
11	Consultar Usuario del Sistema	Medio	Alto	2	1
12	Consultar Jueces de Paz	Medio	Alto	2	1
13	Consultar Juzgados de Paz	Medio	Alto	2	1
14	Consultar Recursos Materiales y Tecnológicos	Medio	Alto	2	1
15	Consultar Juzgados con y sin ternas Asignadas	Medio	Alto	3	1
16	Consultar Cuadro de Aranceles	Medio	Alto	2	1
17	Historia usuario. Administrar Recursos Tecnológicos	Medio	Alto	2	1
18	Emisión de Certificado Escritura Pública de Transferencia de Bienes Inmuebles no registrables	Alto	Alto	3	1

Fuente: Creado por el autor.

Tabla 33*Tarea de Ingeniería. Administrar usuarios del sistema.*

TAREA DE INGENIERÍA	
Número tarea de ingeniería: 1	Usuario historia de usuario: 1
Nombre de tarea: Administrar Usuarios del Sistema	
Tipo de tarea: Desarrollo	Puntos Estimados: 2
Fecha de Inicio: 01/08/2023	Fecha Fin: 04/08/2023
Programador Responsable: Max Auqui Bautista	
Descripción: Usuario registrado responsable de ODAJUP (administrador) accede al sistema y pulsa un clic en “modulo usuarios” el sistema automáticamente le mostrará la interfaz “administración de usuarios”, para crear un nuevo usuario el administrador pulsa un clic en el botón “nuevo”. El sistema presenta la interfaz de “registrar usuario”. El administrador agrega los datos requeridos que son: nombres, apellidos, dni, nombre de usuario, clave de usuario, estado de usuario, tipo de usuario. Si el administrador pulsa un clic en el botón “guardar”, el sistema verifica que los datos requeridos se encuentren con información: si es válido el sistema emite un mensaje de confirmación “guardado satisfactoriamente” y el botón “cerrar” al pulsa un clic te dirige al interfaz “administración de usuarios”. Si el administrador pulsa un clic en el botón “cancelar” el sistema emite un mensaje de confirmación “¿está seguro que desea cancelar?” y los botones “sí” y “no”, si el administrador pulsa un clic en “sí” el sistema redirige a la interfaz “registrar usuarios”, en cambio si pulsa un clic en “no” el sistema permanecerá en la interfaz “registrar usuarios”. Para actualizar la información de un usuario se dirige a la barra de búsqueda e ingresa la inicial del nombre o apellido y el sistema mostrará una lista de usuarios. El administrador selecciona el nombre del usuario registrado de quien se realizará la actualización de los datos y pulsa un clic en el botón de “editar”. El sistema presenta la interfaz “actualizar cuenta de usuario” con los datos extraídos desde la base de datos, el administrador puede modificar donde se podrá modificar todo los campos como son nombres, apellidos, dni, nombre de usuario, clave de usuario, estado de usuario, tipo de usuario, si el administrador pulsa un clic en el botón “editar”, el sistema verifica que los datos requeridos se encuentren con información: si es válido el sistema emite un mensaje de confirmación “modificado satisfactoriamente” y el botón “cerrar” al pulsa un clic te dirige al interfaz “registro de usuarios”. Si el administrador pulsa un clic en el botón “cancelar” el sistema emite un mensaje de confirmación “¿está seguro que desea cancelar?” y los botones “sí” y “no”, si el administrador pulsa un clic en “sí” el sistema redirige a la interfaz “registro de usuarios”, en cambio si pulsa un clic en “no” el sistema permanecerá en la interfaz “actualizar cuenta de usuario”.	
Observaciones: No hay ninguno	

Fuente: Creado por el autor.

Tabla 34*Tarea de ingeniería. Administrar juez de paz.*

TAREA DE INGENIERÍA	
Número tarea de ingeniería: 1	Usuario historia de usuario: 1
Nombre de tarea: Registrar Juez de Paz	
Tipo de tarea: Desarrollo	Puntos Estimados: 2
Fecha de Inicio: 05/08/2023	Fecha Fin: 09/08/2023
Programador Responsable: Max Auqui Bautista	
Descripción: Usuario registrado responsable de ODAJUP (administrador) accede al sistema y pulsa un clic en “módulo juez de paz” el sistema automáticamente le mostrará la interfaz “administración de juez de paz”. Para crear un nuevo usuario el administrador pulsa un clic en el botón “nuevo”. El sistema presenta la interfaz de “registrar juez de paz”. El administrador agrega los datos requeridos que son: nombres, apellidos, dni, teléfono, observacion. Si el administrador pulsa un clic en el botón “guardar”, el sistema verifica que los datos requeridos se encuentren con información: si es válido el sistema emite un mensaje de confirmación “guardado satisfactoriamente” y el botón “cerrar” al pulsa un clic te dirige al interfaz “registrar juez de paz”. Para actualizar la información de un usuario se dirige a la barra de búsqueda e ingresa la inicial del nombre o apellido y el sistema mostrará una lista de usuarios. El administrador selecciona el nombre usuario de quien se realizará la actualización de los datos y pulsa un clic en el botón de “editar”. El sistema presenta la interfaz “actualizar juez de paz” con los datos extraídos desde la base de datos, el administrador puede modificar donde se podrá modificar todo los campos como son nombres, apellidos, dni, teléfono, observación, si el administrador pulsa un clic en el botón “modificar”, el sistema verifica que los datos requeridos se encuentren con información: si es válido el sistema emite un mensaje de confirmación “modificado satisfactoriamente” y el botón “cerrar” al pulsa un clic te dirige al interfaz “registro de usuarios”. Si el administrador pulsa un clic en el botón “cancelar” el sistema emite un mensaje de confirmación “¿está seguro que desea cancelar?” y los botones “sí” y “no”, si el administrador pulsa un clic en “sí” el sistema redirige a la interfaz “registro de usuarios”, en cambio si pulsa un clic en “no” el sistema permanecerá en la interfaz “actualizar juez de paz”. Usuario registrado responsable de odajup accede al sistema y pulsa un clic en “módulo de juez de paz” el sistema automáticamente le mostrará la interfaz “administración de juez de paz”, pulsa un clic en la opción “nuevo” el sistema presenta la interfaz de “registrar de juez de paz”. Usuario registrado agrega información como son datos personales (nombres,	

apellidos,dni, teléfono, observación) y los graba con el botón “guardar”, el sistema revisa que los campos cumplan con los requisitos y los registra en la base de datos de nuevo registro de juez de paz acompañando un mensaje de confirmación de registro.

Usuario registrado estando en el “administración de juez de paz” se dirige a la barra de búsqueda e ingresa la inicial del nombre o apellido y el sistema listará los jueces de paz registrados con la inicial ingresada.

usuario registrado selecciona el nombre del juez de paz de quien se realizará la edición y pulsa un clic en el botón “editar” el sistema le exhibe la interfaz “actualizar juez de paz” con los datos extraídos desde la base de datos donde se podrá modificar todo los campos como son (nombres, apellidos,dni, teléfono, observación), realizado la modificación se dirige a la opción de “guardar”, el sistema verifica la validación de los campos y guarda en la base de datos de la edición y le muestra un mensaje “actualizado satisfactoriamente”.

Observaciones: No hay ninguno

Fuente: Creado por el autor.

Tabla 35

Tarea de ingeniería. Administrar juzgado de paz.

TAREA DE INGENIERÍA	
Número tarea de ingeniería: 1	Usuario historia de usuario: 1
Nombre de tarea: Administrar Juzgado de Paz	
Tipo de tarea: Desarrollo	Puntos Estimados: 2
Fecha de Inicio: 10/08/2023	Fecha Fin: 14/08/2023
Programador Responsable: Max Auqui Bautista	
Descripción: Usuario registrado responsable de ODAJUP (administrador) accede al sistema y pulsa un clic en “módulo juzgado de paz” el sistema automáticamente le mostrará la interfaz “administración de juzgado de paz”. Para crear un nuevo juzgado de paz el administrador pulsa un clic en el botón “nuevo”. El sistema presenta la interfaz de “registrar juzgado de paz”. El administrador agrega los datos requeridos que son: nombre juzgado, departamento, provincia, distrito, dirección, categoría, teléfono, observaciones. Si el administrador pulsa un clic en el botón “guardar”, el sistema verifica que los datos requeridos se encuentren con información: si es válido el sistema emite un mensaje de confirmación “guardado satisfactoriamente” y el botón “cerrar” al pulsa un clic te dirige al interfaz “registrar juzgado de paz”.	

Para actualizar la información de un usuario se dirige a la barra de búsqueda e ingresa la inicial del nombre y el sistema mostrará una lista de juzgados de paz.

El administrador selecciona el nombre del juzgado de paz de quien se realizará la actualización de los datos y pulsa un clic en el botón de “editar”.

El sistema presenta la interfaz “actualizar juzgado de paz” con los datos extraídos desde la base de datos, el administrador puede modificar todos los campos como son nombre juzgado, departamento, provincia, distrito, dirección, categoría, teléfono, observaciones.

Si el administrador pulsa un clic en el botón “modificar”, el sistema verifica que los datos requeridos se encuentren con información: si es válido el sistema emite un mensaje de confirmación “modificado satisfactoriamente” y el botón “cerrar” al pulsa un clic te dirige al interfaz “administración de juzgado de paz”. Si el administrador pulsa un clic en el botón “cancelar” el sistema emite un mensaje de confirmación “¿está seguro que desea cancelar?” y los botones “sí” y “no”, si el administrador pulsa un clic en “sí” el sistema muestra a la interfaz “administración de juzgado de paz”, en cambio si pulsa un clic en “no” el sistema permanecerá en la interfaz “actualizar juzgado de paz”.

Observaciones: No hay ninguno

Fuente: Creado por el autor.

Tabla 36

Tarea de ingeniería. Administrar recursos materiales y tecnológicos.

TAREA DE INGENIERÍA	
Número tarea de ingeniería: 1	Usuario historia de usuario: 1
Nombre de tarea: Administrar Recursos Material y tecnológicos	
Tipo de tarea: Desarrollo	Puntos Estimados: 2
Fecha de Inicio: 15/08/2023	Fecha Fin: 19/08/2023
Programador Responsable: Max Auqui Bautista	
Descripción: Usuario registrado responsable de ODAJUP (administrador) accede al sistema y pulsa un clic en “módulo recurso material y tecnológico” el sistema automáticamente le mostrará la interfaz “administración de recursos materiales y tecnológicos”. Para crear un nuevo bien el administrador pulsa un clic en el botón “nuevo”. El sistema presenta la interfaz de “registrar recurso material y tecnológicos”.	

El administrador agrega los datos requeridos que son: nombre, tipo, serie, código patrimonial, color, descripción, observación. si el administrador pulsa un clic en el botón “guardar”, el sistema verifica que los datos requeridos se encuentren con información: si es válido el sistema emite un mensaje de confirmación “guardado satisfactoriamente” y el botón “cerrar” al pulsa un clic te dirige al interfaz “administración de recursos materiales y tecnológicos”.

Para actualizar la información de un usuario se dirige a la barra de búsqueda e ingresa la inicial del nombre y el sistema mostrará una lista de bienes.

El administrador selecciona el nombre del bien de quien se realizará la actualización de los datos y pulsa un clic en el botón de “editar”.

El sistema presenta la interfaz “actualizar recurso material y tecnológico” con los datos extraídos desde la base de datos, el administrador puede modificar todo los campos como son nombre, tipo, serie, código patrimonial, color, descripción, observación, si el administrador pulsa un clic en el botón “modificar”, el sistema verifica que los datos requeridos se encuentren con información: si es válido el sistema emite un mensaje de confirmación “modificado satisfactoriamente” y el botón “cerrar” al pulsa un clic te dirige al interfaz “administración de recursos materiales y tecnológicos”.

Si el administrador pulsa un clic en el botón “cancelar” el sistema emite un mensaje de confirmación “¿está seguro que desea cancelar?” y los botones “si” y “no”, si el administrador pulsa un clic en “si” el sistema redirige a la interfaz “módulo de recursos materiales y tecnológicos”, en cambio si pulsa un clic en “no” el sistema permanecerá en la interfaz “actualizar recurso material”.

Observaciones: No hay ninguno

Fuente: Creado por el autor.

Tabla 37*Tarea de Ingeniería. Emitir constancias de transferencia posesoria.*

TAREA DE INGENIERÍA	
Número tarea de ingeniería: 1	Usuario historia de usuario: 1
Nombre de tarea: Emitir constancias de transferencia Posesoria	
Tipo de tarea: Desarrollo	Puntos Estimados: 2
Fecha de Inicio: 20/08/2023	Fecha Fin: 25/08/2023
Programador Responsable: Max Auqui Bautista	
Descripción: Usuario registrado Juez de Paz accede al sistema y pulsa un clic en “módulo emisión de certificados” el sistema automáticamente le mostrará la interfaz “administración de certificado”. El juez de paz pulsa un clic en el botón “nuevo” el sistema le mostrará “módulo emisión de certificado” usuario registrado despliega el listado de certificados y elige “escritura pública de transferencia posesoria de bienes inmuebles”. El sistema presenta una ventana donde muestra se tiene que llenar datos según lo requerido como son (nombres y apellidos, dni, estado civil, domicilio, ocupación, estado civil, medida, lugar, distrito, provincia, departamento, límites, contexto de adquisición, modo de pago y folios) y pulsa un clic en el botón “generar certificado” el sistema verifica la validación de los campos y aparecerá un botón “guardar y descargar pdf”, acompañando un mensaje de confirmación de registro.	
Observaciones: No hay ninguno	

Fuente: Creado por el autor.

Tabla 38*Tarea de ingeniería. Asignación de terna al juzgado.*

TAREA DE INGENIERÍA	
Número tarea de ingeniería: 1	Usuario historia de usuario: 1
Nombre de tarea: Asignar Terna al Juzgado.	
Tipo de tarea: Desarrollo	Puntos Estimados: 2
Fecha de Inicio: 26/08/2023	Fecha Fin: 31/08/2023
Programador Responsable: Max Auqui Bautista	

Descripción:

Usuario registrado **responsable de ODAJUP** accede al sistema y pulsa un clic en “módulo asignación de terna” el sistema automáticamente le mostrará la interfaz “asignación de jueces de paz”, usuario registrado pulsa un clic en la opción “nuevo” el sistema presenta la interfaz de “registrar asignación de terna”.

Usuario registrado agrega información las cuales son: número de resolución, fecha de inicio, fecha fin, fecha de resolución, subir archivo pdf, seleccionar nombre del juzgado de paz, juez titular, accesitario 1, accesitario 2, el sistema verifica la información ingresada: si es válido el sistema emite un mensaje de confirmación “guardado satisfactoriamente” al realizar clic en el botón “cerrar” el sistema automáticamente muestra la interfaz de “asignación de jueces de paz”.

Si el usuario registrado pulsa un clic en el botón de cerrar el sistema automáticamente le muestra la interfaz de “asignación de jueces de paz”.

Observaciones: No hay ninguno

Fuente: Creado por el autor.

Tabla 39

Tarea de ingeniería. Consultar Juzgados con ternas próximas a vencer.

TAREA DE INGENIERÍA	
Número tarea de ingeniería: 1	Usuario historia de usuario: 1
Nombre de tarea: Consultar Juzgados con Ternas Próximos a Vencer.	
Tipo de tarea: Desarrollo	Puntos Estimados: 2
Fecha de Inicio: 01/09/2023	Fecha Fin: 03/09/2023
Programador Responsable: Max Auqui Bautista	
Descripción: Usuario registrado Responsable de ODAJUP accede al sistema y pulsa un clic en "módulo asignación de ternas" el sistema automáticamente le mostrará la interfaz “asignación de jueces de paz”. Usuario registrado ingresa la cantidad de días en el campo de ‘buscar por días por vencer’ el sistema listará los juzgados, el número de resolución, ternas, fecha de inicio, fecha final, vigencia plasmado en los días a vencer.	
Observaciones: No hay ninguno	

Fuente: Creado por el autor.

Tabla 40*Tarea de Ingeniería. Iniciar sesión de jueces.*

TAREA DE INGENIERÍA	
Número tarea de ingeniería: 1	Usuario historia de usuario: 1
Nombre de tarea: Iniciar Sesión de Usuario	
Tipo de tarea: Desarrollo	Puntos Estimados: 2
Fecha de Inicio: 04/09/2023	Fecha Fin: 07/09/2023
Programador Responsable: Max Auqui Bautista	
Descripción: El Juez de Paz accede al sistema; el sistema presenta la página “iniciar sesión”. El usuario ingresa el nombre de usuario y contraseña; selecciona el botón “ingresar”, el sistema muestra la página juez de paz con los módulos: modulo personal, módulo de emisión de certificado, si usuario registrado pulsa un clic en el botón menú el sistema le mostrará los módulos: módulo personal, módulo de emisión de certificado, el sistema valida los datos ingresados y, si son correctos, continúa con la operación. En caso contrario, muestra mensajes de error y solicita los datos nuevamente.	
Observaciones: No hay ninguno	

Fuente: Creado por el autor.

Tabla 41*Tarea de ingeniería. Iniciar sesión de administrador.*

TAREA DE INGENIERÍA	
Número tarea de ingeniería: 1	Usuario historia de usuario: 1
Nombre de tarea: Iniciar Sesión de Administrador	
Tipo de tarea: Desarrollo	Puntos Estimados: 2
Fecha de Inicio: 08/09/2023	Fecha Fin: 10/09/2023
Programador Responsable: Max Auqui Bautista	
Descripción: El administrador del sistema accede al sistema; el sistema presenta la página “iniciar sesión”. el administrador ingresa el nombre de usuario y contraseña; selecciona el botón “ingresar”, el sistema muestra la página principal con los módulos: módulo juzgado de paz, módulo juez de paz, módulo recurso material, módulo cuadro aranceles, módulo asignación de terna y módulo de personal, si usuario registrado pulsa un clic en el botón menú el sistema le mostrará los módulos: módulo juzgado de paz, módulo juez de paz, módulo recurso material, módulo cuadro aranceles, módulo asignación de terna y módulo de personal, el sistema valida los datos ingresados y, si son correctos, continúa con la operación. En caso contrario, muestra mensajes de error y solicita los datos nuevamente.	
Observaciones: No hay ninguno	

Fuente: Creado por el autor.

Tabla 42*Tarea de ingeniería. Administrar cuadro de aranceles.*

TAREA DE INGENIERÍA	
Número tarea de ingeniería: 1	Usuario historia de usuario: 1
Nombre de tarea: Administrar Cuadro de Aranceles	
Tipo de tarea: Desarrollo	Puntos Estimados: 2
Fecha de Inicio: 11/09/2023	Fecha Fin: 13/09/2023
Programador Responsable: Max Auqui Bautista	

Descripción:

Usuario registrado responsable de ODAJUP (administrador) accede al sistema y pulsa un clic en “módulo cuadro aranceles” el sistema automáticamente le mostrará la interfaz “administración de cuadro de aranceles”.

Para crear un nuevo arancel el administrador pulsa un clic en el botón “nuevo”. El sistema presenta la interfaz de “registro de aranceles”.

El administrador agrega los datos requeridos que son: concepto, costo de arancel. Si el administrador pulsa un clic en el botón “guardar”, el sistema verifica que los datos requeridos se encuentren con información: si es válido el sistema emite un mensaje de confirmación “guardado satisfactoriamente” y el botón “cerrar” al pulsa un clic te dirige al interfaz “administración de juzgado de paz”.

Para actualizar la información de un arancel se dirige a la barra de búsqueda e ingresa la inicial del concepto y el sistema mostrará una lista de aranceles.

El administrador selecciona el nombre del concepto del arancel de quien se realizará la actualización de los datos y pulsa un clic en el botón de “editar”.

El sistema presenta la interfaz “actualizar aranceles” con los datos extraídos desde la base de datos, el administrador puede modificar todo los campos como son concepto, costo de arancel.

Si el administrador pulsa un clic en el botón “modificar”, el sistema verifica que los datos requeridos se encuentren con información: si es válido el sistema emite un mensaje de confirmación “modificado satisfactoriamente” y el botón “cerrar” al pulsa un clic te dirige al interfaz “administración de cuadro de aranceles”.

Si el administrador pulsa un clic en el botón “cancelar” el sistema emite un mensaje de confirmación “¿está seguro que desea cancelar?” y los botones “si” y “no”, si el administrador pulsa un clic en “si” el sistema redirige a la interfaz “cuadro de aranceles”, en cambio si pulsa un clic en “no” el sistema permanecerá en la interfaz “actualizar aranceles”.

Observaciones: No hay ninguno

Fuente: Creado por el autor.

Tabla 43*Tarea de ingeniería. Consultar usuario del sistema.*

TAREA DE INGENIERÍA	
Número tarea de ingeniería: 1	Usuario historia de usuario: 1
Nombre de tarea: Consultar Usuario del Sistema	
Tipo de tarea: Desarrollo	Puntos Estimados: 2
Fecha de Inicio: 14/09/2023	Fecha Fin: 16/09/2023
Programador Responsable: Max Auqui Bautista	
Descripción: Usuario registrado Responsable de ODAJUP accede al sistema, pulsa un clic en “módulo usuarios” y el sistema automáticamente le muestra la interfaz “administración de usuarios”. Usuario registrado ingresa una letra en la barra de búsqueda nombre o apellido y el sistema listará los usuarios registrados con la inicial ingresada como son dni, nombres, apellidos, nombre usuario, estado, tipo usuario.	
Observaciones: No hay ninguno	

Fuente: Creado por el autor.

Tabla 44*Tarea de ingeniería. Consulta de jueces de paz.*

TAREA DE INGENIERÍA	
Número tarea de ingeniería: 1	Usuario historia de usuario: 1
Nombre de tarea: Consultar Jueces de Paz	
Tipo de tarea: Desarrollo	Puntos Estimados: 2
Fecha de Inicio: 17/09/2023	Fecha Fin: 19/09/2023
Programador Responsable: Max Auqui Bautista	

Descripción: Usuario registrado Responsable de ODAJUP accede al sistema, pulsa un clic en “módulo juez de paz” y el sistema automáticamente le muestra “administración de juez de paz”. Usuario registrado ingresa una letra en la barra de búsqueda nombre o apellido y el sistema listará los jueces de paz registrados con la inicial ingresada, con los siguientes datos, dni, nombres, apellidos.
Observaciones: No hay ninguno

Fuente: Creado por el autor.

Tabla 45

Tarea de ingeniería. Consultar juzgados de paz.

TAREA DE INGENIERÍA	
Número tarea de ingeniería: 1	Usuario historia de usuario: 1
Nombre de tarea: Consultar Juzgados de Paz.	
Tipo de tarea: Desarrollo	Puntos Estimados: 2
Fecha de Inicio: 20/09/2023	Fecha Fin: 22/09/2023
Programador Responsable: Max Auqui Bautista	
Descripción: Usuario registrado Responsable de ODAJUP accede al sistema, pulsa un clic en “módulo juzgado de paz” y el sistema le exhibe la interfaz de “administración de juzgados de paz”. Usuario registrado ingresa una letra en la barra de búsqueda e ingresa la inicial del nombre del juzgado y el sistema listará los juzgados registrados con la inicial ingresada, con los siguientes datos: nombre, dirección, teléfono.	
Observaciones: No hay ninguno	

Fuente: Creado por el autor.

Tabla 46*Tarea de ingeniería. Consultar recursos materiales y tecnológicos.*

TAREA DE INGENIERÍA	
Número tarea de ingeniería: 1	Usuario historia de usuario: 1
Nombre de tarea: Consultar Recursos Materiales y Tecnológicos	
Tipo de tarea: Desarrollo	Puntos Estimados: 2
Fecha de Inicio: 23/09/2023	Fecha Fin: 25/09/2023
Programador Responsable: Max Auqui Bautista	
Descripción: Usuario registrado Responsable de ODAJUP accede al sistema, pulsa un clic en el módulo “recursos materiales y tecnológicos” el sistema le mostrará la interfaz de “administración recursos materiales y tecnológicos”. Usuario registrado ingresa una letra en la barra de búsqueda e ingresa la inicial del nombre del recurso material y el sistema listará los bienes registrados con la inicial ingresada. El sistema le mostrará nombre, tipo, serie, código patrimonial, color, descripción y observación.	
Observaciones: No hay ninguno	

Fuente: Creado por el autor.

Tabla 47*Tarea de ingeniería. Consultar juzgados con y sin ternas asignadas.*

TAREA DE INGENIERÍA	
Número tarea de ingeniería: 1	Usuario historia de usuario: 1
Nombre de tarea: Consultar Juzgados con terna y sin ternas Asignadas	
Tipo de tarea: Desarrollo	Puntos Estimados: 2
Fecha de Inicio: 26/09/2023	Fecha Fin: 28/09/2023

Programador Responsable: Max Auqui Bautista
Descripción: Usuario registrado Responsable de ODAJUP (administrador) accede al sistema y pulsa un clic en "módulo asignación de terna" el sistema automáticamente le mostrará la interfaz "asignación de jueces de paz". Usuario registrado elige el tipo de búsqueda de juzgados, se tiene 3 opciones: todo, asignados, no asignados, usuario registrado al seleccionar una de las 3 opciones: ingresa la inicial del nombre del juzgado del quien realizará la verificación si se tiene la terna asignados/ no asignados el sistema listará los juzgados de paz con los siguientes datos: juzgado, nro. Resolución, terna, fecha inicio, fecha fin y días de vigencia.
Observaciones: No hay ninguno

Fuente: Creado por el autor.

Tabla 48

Tarea de ingeniería. Consultar el cuadro de aranceles.

TAREA DE INGENIERÍA	
Número tarea de ingeniería: 1	Usuario historia de usuario: 1
Nombre de tarea: Consultar Cuadro de Aranceles	
Tipo de tarea: Desarrollo	Puntos Estimados: 2
Fecha de Inicio: 29/09/2023	Fecha Fin: 30/09/2023
Programador Responsable: Max Auqui Bautista	
Descripción: Usuario registrado Responsable de ODAJUP pulsa un clic en "módulo cuadro aranceles" el sistema automáticamente le muestra la interfaz de "administración de cuadro de aranceles". Usuario registrado digita una letra en la barra de búsqueda e ingresa la inicial del concepto de arancel y el sistema listará los aranceles con los siguientes datos: concepto, costo arancel.	
Observaciones: No hay ninguno	

Fuente: Creado por el autor.

Tabla 49*Plan de iteración (primera)*

N	HISTORIAS DE USUARIO	TAREAS DE INGENIERÍA
1	Administrar Usuarios del Sistema	Registro de datos del usuario registrado del sistema.
		Actualización de los datos del usuario registrado
		Búsqueda de Usuarios de Sistema
2	Administrar Juez de Paz	Registro de Datos del Juez de Paz
		Actualización de datos del Juez de Paz
		Búsqueda de Jueces de Paz
3	Administrar Juzgado de Paz	Registro del Juzgado de Paz
		Actualización de datos del Juzgado de Paz
		Búsqueda de Juzgados de Paz
4	Administrar Recursos Materiales y Tecnológicos	Registro del Recurso Materiales
		Actualización del Recurso Materiales
		Búsqueda de Recurso Material y Tecnológico
5	Emitir constancias de transferencia Posesoria	Registro de la constancia de transferencia Posesoria Generar PDF e Imprimir
6	Asignar Terna al Juzgado.	Registro del Juzgado de Paz con sus respectivas ternas.
		Búsqueda de Juzgados con Terna
7	Consultar Juzgados con Ternas Próximos a Vencer.	Reporte de Juzgados con ternas próximos a vencer
10	Administrar Cuadro de Aranceles	Registro de Cuadro de Aranceles
		Actualización de Cuadro de Aranceles
		Búsqueda de Aranceles

Fuente: Creado por el autor.

Tabla 50

Plan de iteración (segunda)

N	HISTORIAS DE USUARIO	TAREAS DE INGENIERÍA
8	Iniciar Sesión de Jueces	Iniciar Sesión como Juez de Paz
9	Iniciar Sesión de Administrador	Iniciar Sesión como Administrador
11	Consultar Usuario del Sistema	Reporte de Usuarios del Sistema
12	Consultar Jueces de Paz	Reporte de Jueces de paz
13	Consultar Juzgados de Paz.	Reporte de Juzgados de paz
14	Consultar Recursos Materiales y Tecnológicos	Reporte de Recursos Materiales
15	Consultar Juzgados con y sin ternas Asignadas	Reporte de Juzgados con terna Reporte de Juzgados sin terna
16	Consultar Cuadro de Aranceles	Reporte de Aranceles Judiciales

Fuente: Creado por el autor.

Tabla 51*Plan de interacciones clasificadas en fechas de desarrollo*

N	TAREA DE INGENIERÍA	FECHA INICIO	FECHA FIN	RESPONSABLE
1	1	01/08/2023	04/08/2023	MAX AUQUI BAUTISTA
	2	01/08/2023	04/08/2023	MAX AUQUI BAUTISTA
	3	01/08/2023	04/08/2023	MAX AUQUI BAUTISTA
2	1	05/08/2023	08/08/2023	MAX AUQUI BAUTISTA
	2	05/08/2023	08/08/2023	MAX AUQUI BAUTISTA
	3	05/08/2023	08/08/2023	MAX AUQUI BAUTISTA
3	1	09/08/2023	12/08/2023	MAX AUQUI BAUTISTA
	2	09/08/2023	12/08/2023	MAX AUQUI BAUTISTA
	3	09/08/2023	12/08/2023	MAX AUQUI BAUTISTA
4	1	13/08/2023	16/08/2023	MAX AUQUI BAUTISTA
	2	13/08/2023	16/08/2023	MAX AUQUI BAUTISTA
	3	13/08/2023	16/08/2023	MAX AUQUI BAUTISTA
5	1	17/08/2023	21/08/2023	MAX AUQUI BAUTISTA
6	1	22/08/2023	26/08/2023	MAX AUQUI BAUTISTA
	2	22/08/2023	26/08/2023	MAX AUQUI BAUTISTA
7	1	27/08/2023	30/08/2023	MAX AUQUI BAUTISTA
8	1	31/08/2023	02/09/2023	MAX AUQUI BAUTISTA
9	1	03/09/2023	05/09/2023	MAX AUQUI BAUTISTA
10	1	06/09/2023	09/09/2023	MAX AUQUI BAUTISTA
	2	06/09/2023	09/09/2023	MAX AUQUI BAUTISTA
	3	06/09/2023	09/09/2023	MAX AUQUI BAUTISTA
11	1	10/09/2023	11/09/2023	MAX AUQUI BAUTISTA
12	1	12/09/2023	13/09/2023	MAX AUQUI BAUTISTA
13	1	14/09/2023	15/09/2023	MAX AUQUI BAUTISTA
14	1	16/09/2023	17/09/2023	MAX AUQUI BAUTISTA

15	1	18/09/2023	20/09/2023	MAX AUQUI BAUTISTA
16	1	21/09/2023	23/09/2023	MAX AUQUI BAUTISTA

Fuente: Creado por el autor.

Tabla 52

Casos de prueba de aceptación

Nº H.U.	REQUISITO	Nº C.P.	CASO DE PRUEBA (C.P.)
1	Usuario (Administrador) puede crear nuevos usuarios en el sistema.	1	El sistema verifica los atributos en el formulario de registro.
	Usuario (Administrador) puede realizar la Actualización de los datos del usuario registrado	2	El sistema emite un mensaje de confirmación de actualización de registros.
	Usuario (administrador) puede buscar el listado de usuarios registrados en el sistema.	3	El sistema recupera información de la base de datos de usuarios registrados.
2	Usuario (administrador) es capaz de registrar juez de Paz	1	El sistema verifica los atributos en el formulario de registro.
	Usuario (administrador) puede realizar la Actualización de datos de los Juez de Paz	2	El sistema emite un mensaje de confirmación de actualización de registros.
	Usuario (administrador) puede realizar la búsqueda de Jueces de Paz	3	El sistema recupera información de la base de datos de Jueces de paz registrados.
3	Usuario (administrador) puede realizar el registro del Juzgado de Paz	1	El sistema verifica los atributos en el formulario de registro.
	Usuario (administrador) puede realizar la actualización de datos del Juzgado de Paz	2	El sistema emite un mensaje de confirmación de actualización de registros.
	Usuario (administrador) puede realizar la búsqueda de Juzgados de Paz	3	El sistema recupera información de la base de datos de Juzgados de paz registrados.
4	Usuario (administrador) puede realizar el registro del recurso Material y Tecnológico	1	El sistema verifica los atributos en el formulario de registro.
	Usuario (administrador) puede realizar la actualización del	2	El sistema emite un mensaje de confirmación de actualización de

	recurso material y Tecnológico		registros.
	Usuario (administrador) puede realizar la búsqueda de recurso material y tecnológico	3	El sistema recupera información de la base de datos de recursos materiales y tecnológicos registrados.
5	Usuario (juez de paz) puede Emitir Certificado Escritura pública de Transferencia de bienes inmuebles no registrables, generar PDF y Descargar	1	El sistema verifica los atributos en el formulario de registro.
6	Usuario (administrador) puede realizar el registro del Juzgado de Paz y la asignación de su respectiva terna.	1	El sistema verifica los atributos en el formulario de registro.
	Usuario (administrador) puede realizar la búsqueda de Juzgados de paz con Terna	2	El sistema recupera información de la base de datos de juzgados con terna asignada.
7	Usuario (administrador) puede realizar la búsqueda de Juzgados con ternas próximos a vencer	3	El sistema recupera información de la base de datos de los juzgados de paz con ternas próximas a vencer.
10	Usuario (administrador) puede realizar el registro de cuadro de aranceles	1	El sistema verifica los atributos en el formulario de registro.
	Usuario (administrador) puede realizar la actualización de cuadro de aranceles	2	El sistema emite un mensaje de confirmación de actualización de registros.
	Usuario (administrador) puede realizar la búsqueda de Aranceles registrados.	3	El sistema recupera información de la base de datos de aranceles registrados

Fuente: Creado por el autor.

4.1.2. INTERFAZ DE USUARIO Y CODIFICACIÓN

Las interfaces se han diseñado para ser fáciles de usar y amigables, siguiendo los estándares de usabilidad. Los factores que se han tenido en cuenta son los botones, títulos de páginas y menús, fuentes y colores, tamaños de fuentes, etc.

Aunque los objetivos de la tesis no incluyen la evaluación de la usabilidad, estos factores son importantes para garantizar que las interfaces sean fáciles de usar y amigables.

Figura 9

Interfaz, Sistema de administración de juzgados de paz

The screenshot shows the login page of a web application. At the top, there is a dark red header bar with a home icon, the text 'ADMINISTRACIÓN DE JUZGADOS DE PAZ', a user icon, and the text 'Iniciar Sesión'. Below the header, the title 'Iniciar Sesión' is centered. There are two input fields: 'Usuario *' and 'Clave *'. Below these fields is a red button labeled 'INICIAR SESIÓN'. At the bottom, there is a link that says '¿Olvidó la contraseña? Contactese con su administrador'.

Fuente: Creado por el autor.

Figura 10

Interfaz. Menú principal del sistema



Fuente: Creado por el autor.

Figura 11

Interfaz. Administración de juzgado de paz

Header bar with a dark red background. On the left, there is a hamburger menu icon, a home icon, and the text 'ADMINISTRACIÓN DE JUZGADOS DE PAZ'. On the right, there is the text 'carmelo', a user profile icon, and the text 'Cerrar sesión'.

[← REGRESAR](#)

ADMINISTRACIÓN DE JUZGADO DE PAZ

Main content area with a light gray background. It contains a search bar with the placeholder text 'Buscar por nombre de juzgado' and a blue 'BUSCAR' button. Below the search bar, there are two buttons: a green '+ NUEVO' button and a gray 'EDITAR' button with a pencil icon.

Fuente: Creado por el autor.

Figura 12

Interfaz. Registro de juzgado de paz

Header bar with a dark red background. On the left, there is a hamburger menu icon, a home icon, and the text 'ADMINISTRACIÓN DE JUZGADOS DE PAZ'. On the right, there is the text 'max', a user profile icon, and the text 'Cerrar sesión'.

REGISTRAR JUZGADO DE PAZ

Main content area with a light gray background. It contains a form with the following fields: 'NOMBRE JUZGADO *' (text input), 'DEPARTAMENTO *' (dropdown menu), 'PROVINCIA *' (dropdown menu), 'DISTRITO *' (dropdown menu), 'CATEGORIA' (dropdown menu), 'DIRECCION *' (text input), 'TELEFONO' (text input), and 'OBSERVACIONES' (text input). At the bottom, there are two buttons: a blue 'GUARDAR' button and a red 'CANCELAR' button.

Fuente: Creado por el autor.

Figura 13

Interfaz. Administración de juez de paz.



Header bar with a dark red background. On the left, there is a hamburger menu icon, a home icon, and the text 'ADMINISTRACIÓN DE JUZGADOS DE PAZ'. On the right, there is the name 'carmelo', a user profile icon, and the text 'Cerrar sesión'.

[← REGRESAR](#)

ADMINISTRACIÓN DE JUEZ DE PAZ



A light gray rounded rectangle containing a search bar with the placeholder text 'Buscar por nombre, apellido o dni' and a blue 'BUSCAR' button. Below the search bar are three buttons: a green '+ NUEVO' button, a gray 'EDITAR' button with a pencil icon, and a gray 'HABILITAR/INHABILITAR' button with a checkmark icon.

Fuente: Creado por el autor.

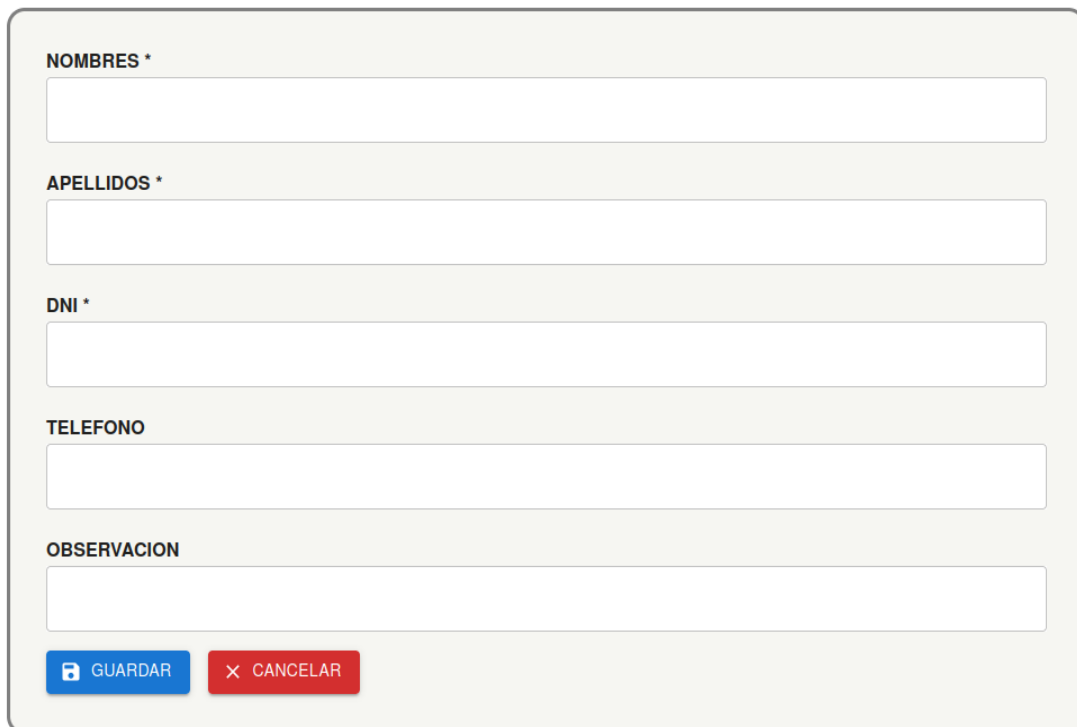
Figura 14

Interfaz. Registro de juez de paz



Header bar with a dark red background. On the left, there is a home icon and the text 'ADMINISTRACIÓN DE JUZGADOS DE PAZ'. On the right, there is a user profile icon and the text 'Iniciar Sesión'.

REGISTRAR JUEZ DE PAZ



A light gray rounded rectangle containing a registration form. The form has five input fields: 'NOMBRES *', 'APELLIDOS *', 'DNI *', 'TELEFONO', and 'OBSERVACION'. At the bottom, there are two buttons: a blue 'GUARDAR' button with a save icon and a red 'CANCELAR' button with a close icon.

Fuente: Creado por el autor.

Figura 15

Interfaz. Administración de cuadro de aranceles

The header of the application is a dark red bar. On the left, there is a hamburger menu icon followed by a home icon and the text 'ADMINISTRACIÓN DE JUZGADOS DE PAZ'. On the right, the username 'carmelo' is displayed next to a user profile icon and a 'Cerrar sesión' button.

[← REGRESAR](#)

ADMINISTRACIÓN DE CUADRO DE ARANCELES

This interface contains a search bar with the placeholder text 'Buscar por concepto de arancel' and a blue 'BUSCAR' button. Below the search bar are two buttons: a green '+ NUEVO' button and a grey 'EDITAR' button with a pencil icon.

Fuente: Creado por el autor.

Figura 16

Interfaz. Registro de cuadro de aranceles

The header is identical to the one in Figure 15, showing the application name and user information.

REGISTRAR ARANCELES

The registration form has two main input fields. The first is labeled 'CONCEPTO *' and is an empty text box. The second is labeled 'COSTO DE ARANCEL *' and contains the value '0'. At the bottom of the form are two buttons: a blue 'GUARDAR' button with a save icon and a red 'CANCELAR' button with an 'X' icon.

Fuente: Creado por el autor.

Figura 17

Interfaz. Administración de recursos materiales.

The screenshot shows a web interface with a dark red header. On the left, there is a hamburger menu icon and a home icon followed by the text 'ADMINISTRACIÓN DE JUZGADOS DE PAZ'. On the right, the user name 'carmelo' is displayed next to a profile icon and a 'Cerrar sesión' button. Below the header, there is a red button with a left arrow and the text 'REGRESAR'. The main content area has a large heading 'ADMINISTRACIÓN DE RECURSOS MATERIALES'. Below this heading is a light gray box containing a search bar with the placeholder text 'Buscar por nombre de recurso' and a blue 'BUSCAR' button. At the bottom of this box are two buttons: a green '+ NUEVO' button and a gray 'EDITAR' button with a pencil icon.

Fuente: Creado por el autor.

Figura 18

Interfaz. Registro de recurso materia.

The screenshot shows a web interface with a dark red header. On the left, there is a hamburger menu icon and a home icon followed by the text 'ADMINISTRACIÓN DE JUZGADOS DE PAZ'. On the right, the user name 'max' is displayed next to a profile icon and a 'Cerrar sesión' button. Below the header, there is a large heading 'REGISTRAR RECURSO MATERIAL'. The main content area is a light gray box containing several form fields. At the top is a dropdown menu labeled 'NOMBRE DE JUZGADO DE PAZ'. Below this are two input fields: 'NOMBRE' and 'CODIGO PATRIMONIAL'. Underneath these are three input fields: 'TIPO', 'SERIE', and 'COLOR'. Below these is a large text area labeled 'DESCRIPCION'. At the bottom is another large text area labeled 'OBSERVACION'. At the very bottom of the form box are two buttons: a blue 'GUARDAR' button with a save icon and a red 'CANCELAR' button with a close icon.

Fuente: Creado por el autor.

Figura 19

Interfaz. Registro de asignación de jueces de paz

ADMINISTRACIÓN DE JUZGADOS DE PAZ max Cerrar sesión

← REGRESAR

ASIGNACIÓN DE JUECES DE PAZ

Buscar por juzgado:

Escribe el nombre del juzgado

Buscar por días por vencer:

30

Filtrar juzgado por:

☒ Todo ☐ Asignados ☐ No asignados

Fuente: Creado por el autor.

Figura 20

Interfaz. Registro asignación de terna

ADMINISTRACIÓN DE JUZGADOS DE PAZ max Cerrar sesión

REGISTRAR ASIGNACIÓN DE TERNA

NUMERO DE RESOLUCION

RESOLUCION PDF

Ningún archivo subido.

FECHA DE RESOLUCION mm / dd / yyyy

FECHA INICIO mm / dd / yyyy

FECHA FIN mm / dd / yyyy

NOMBRE DE JUZGADO DE PAZ

SELECCIONAR JUEZ TITULAR

SELECCIONAR ACCESITARIO 1

SELECCIONAR ACCESITARIO 2

Fuente: Creado por el autor.

Figura 21

Interfaz. Administración de usuario



The header of the user administration interface. It features a dark red navigation bar. On the left, there is a hamburger menu icon followed by a home icon and the text 'ADMINISTRACIÓN DE JUZGADOS DE PAZ'. On the right, the username 'carmelo' is displayed next to a user profile icon and the text 'Cerrar sesion'.

[← REGRESAR](#)

ADMINISTRACION DE USUARIOS



A search and action bar for user management. It includes a text input field with the placeholder 'Buscar por usuario o apellido', a blue 'BUSCAR' button with a magnifying glass icon, a green '+ NUEVO' button, and a grey 'EDITAR' button with a pencil icon.

Fuente: Creado por el autor.

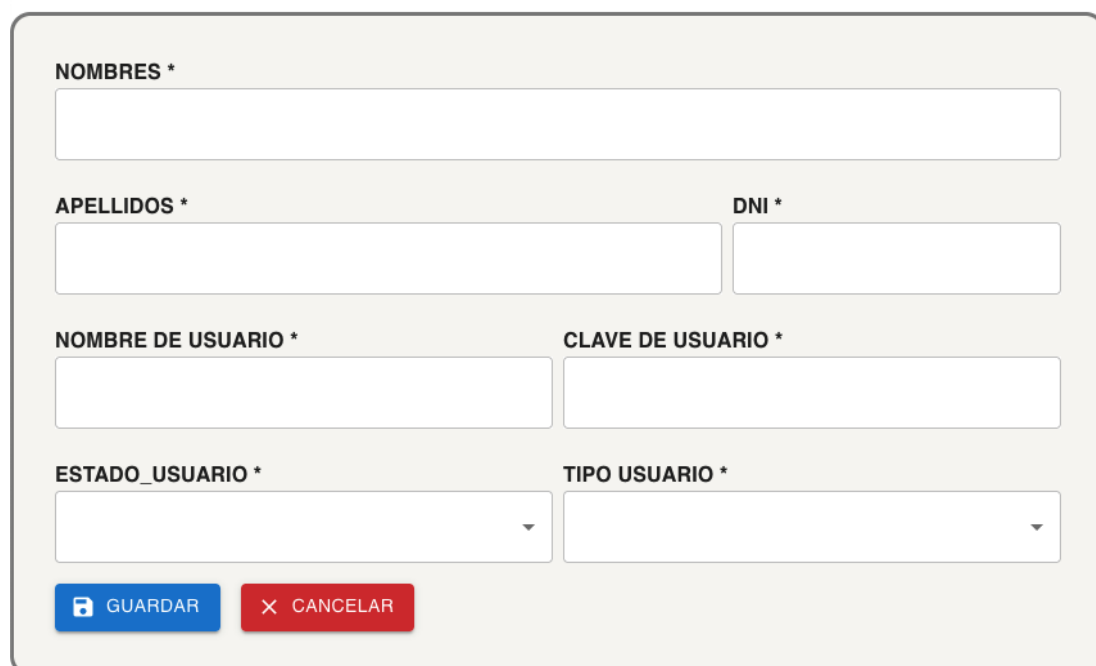
Figura 22

Interfaz. Registro cuenta de usuario.



The header of the user registration interface. It features a dark red navigation bar. On the left, there is a hamburger menu icon followed by a home icon and the text 'ADMINISTRACIÓN DE JUZGADOS DE PAZ'. On the right, the username 'max' is displayed next to a user profile icon and the text 'Cerrar sesion'.

REGISTRAR CUENTA DE USUARIO



A form for registering a new user account. It contains several input fields with asterisks indicating required fields: 'NOMBRES *', 'APELLIDOS *', 'DNI *', 'NOMBRE DE USUARIO *', 'CLAVE DE USUARIO *', 'ESTADO_USUARIO *' (a dropdown menu), and 'TIPO USUARIO *' (a dropdown menu). At the bottom, there are two buttons: a blue 'GUARDAR' button with a save icon and a red 'CANCELAR' button with a close icon.

Fuente: Creado por el autor.

Figura 23

Interfaz. Menú principal del sistema juez de paz



Fuente: Creado por el autor.

Figura 24

Interfaz. Administración de emisión de certificados



[← REGRESAR](#)

ADMINISTRACIÓN EMISION DE CERTIFICADOS

Fuente: Creado por el autor.

Figura 25

Interfaz. Modulo emisión de certificados

The screenshot shows the 'MODULO EMISIÓN DE CERTIFICADOS' interface. At the top, there is a dark red header bar with a menu icon, a home icon, the text 'ADMINISTRACIÓN DE JUZGADOS DE PAZ', the user role 'juez', and a 'Cerrar sesión' button. Below the header, there is a red button labeled '← REGRESAR'. The main content area is titled 'MODULO EMISIÓN DE CERTIFICADOS'. A dropdown menu is open, showing a list of certificate types under the heading 'CERTIFICADO'. The list includes: 'Escritura pública de transferencia posesoria de bienes inmuebles', 'Escritura pública de transferencia de bienes muebles no registrables', 'Certificación de firmas', 'Certificación de copias de documentos presentados por la parte interesada, y otras reproducciones', 'Otorgamiento de copias certificadas de documentos que obran en el archivo del Juzgado de Paz', 'Certificación de apertura de libros', 'Constancia de actos y decisiones que adopten en asamblea las organizaciones sociales o comunales', 'Constancia de posesión', 'Constancia domiciliaria', 'Constancia de supervivencia', 'Constancia de convivencia', 'Constancia de viudez', and 'Otras constancias de hechos que el juez de paz pueda verificar personalmente'.

Fuente: Creado por el autor.

Figura 26

Interfaz. Certificado: Escritura pública de transferencia posesoria de bienes inmuebles

The screenshot shows the 'MODULO EMISIÓN DE CERTIFICADOS' interface for issuing a certificate. The header bar is the same as in Figure 25. The main content area is titled 'MODULO EMISIÓN DE CERTIFICADOS'. A dropdown menu is open, showing the selected certificate type: 'Escritura pública de transferencia posesoria de bienes inmuebles'. Below the dropdown, there is a section titled 'DATOS DEL ARANCEL'. It contains two columns: 'CONCEPTO' and 'COSTO (SOLES)'. The 'CONCEPTO' column has a text input field with the value 'Escritura pública de transferencia posesoria de bienes inmuebles'. The 'COSTO (SOLES)' column has a text input field with the value '18.27'. Below this section, there is a section titled 'DATOS DEL SOLICITANTE'. It contains four input fields: 'NOMBRES Y APELLIDOS' (a large text input field), 'DNI' (a text input field), 'ESTADO CIVIL' (a dropdown menu), and 'OCUPACION' (a text input field). Below these fields, there is a section titled 'DOMICILIO' with a large text input field. At the bottom, there is a section titled 'DATOS DEL RECEPTOR' with a text input field for 'NOMBRES Y APELLIDOS'.

Fuente: Creado por el autor.

Figura 27

Interfaz. Certificado: Escritura pública de bienes muebles no registrables.

ADMINISTRACIÓN DE JUZGADOS DE PAZ

juez  Cerrar sesión

MODULO EMISIÓN DE CERTIFICADOS

CERTIFICADO

Escritura pública de transferencia de bienes muebles no registrables

DATOS DEL ARANCEL

CONCEPTO

Escritura pública de transferencia de bienes muebles no registrables

COSTO (SOLES)

24.91

DATOS DEL SOLICITANTE

NOMBRES Y APELLIDOS

DNI

ESTADO CIVIL

OCUPACION

DOMICILIO

DATOS DEL INMUEBLE

NOMBRE

MEDIDA

Fuente: Creado por el autor.

Figura 28

Interfaz. Administración de usuarios.

ADMINISTRACIÓN DE JUZGADOS DE PAZ

juez  Cerrar sesión

← REGRESAR

ADMINISTRACION DE USUARIOS

EDITAR

LISTA DE PERSONAL

	DNI	NOMBRES	APELLIDOS	NOMBRE USUARIO	ESTADO	TIPO USUARIO
☐	99999998	juez	juez de paz	juez	activo	juez

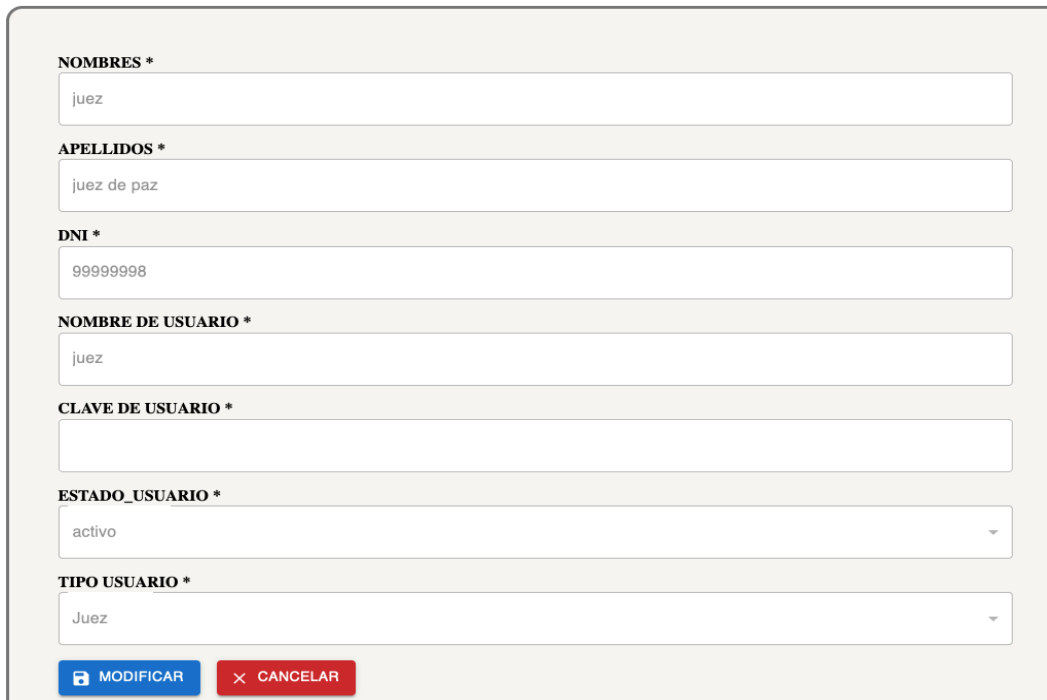
Fuente: Creado por el autor.

91

Figura 29

Interfaz. Actualización de la cuenta de usuario.

ACTUALIZAR CUENTA DE USUARIO



Formulario para actualizar la cuenta de usuario. El formulario contiene los siguientes campos:

- NOMBRES ***: Campo de texto con el valor "juez".
- APELLIDOS ***: Campo de texto con el valor "juez de paz".
- DNI ***: Campo de texto con el valor "99999998".
- NOMBRE DE USUARIO ***: Campo de texto con el valor "juez".
- CLAVE DE USUARIO ***: Campo de texto vacío.
- ESTADO_USUARIO ***: Selector de lista desplegable con el valor "activo".
- TIPO USUARIO ***: Selector de lista desplegable con el valor "Juez".

En la parte inferior del formulario hay dos botones: "MODIFICAR" (azul) y "CANCELAR" (rojo).

Fuente: Creado por el autor.

Figura 30

Interfaz. Administración de Recursos Tecnológicos.



Interfaz de Administración de Recursos Tecnológicos. La interfaz muestra una barra superior roja con el texto "ADMINISTRACIÓN DE JUZGADOS DE PAZ" y el nombre de usuario "carmelo". A la derecha de la barra superior hay un botón "Cerrar sesión".

Debajo de la barra superior hay un botón "← REGRESAR".

El título principal de la interfaz es "ADMINISTRACIÓN DE RECURSOS TECNOLÓGICOS".

Debajo del título principal hay un formulario con los siguientes elementos:

- Campo de búsqueda: "Buscar por nombre de recurso".
- Botón "BUSCAR" (azul).
- Botón "+ NUEVO" (verde).
- Botón "EDITAR" (gris).

Fuente: Creado por el autor.

Figura 31

Interfaz. Registro de Recurso Tecnológico.

REGISTRAR RECURSO TECNOLÓGICO

NOMBRE DE JUZGADO DE PAZ

NOMBRE CODIGO PATRIMONIAL

TIPO SERIE COLOR

DESCRIPCION

OBSERVACION

GUARDAR CANCELAR

Fuente: Creado por el autor.

4.1.3. TARJETAS CLASE RESPONSABILIDAD COLABORACIÓN (CRC)

Las tarjetas CRC son una técnica de modelado de sistemas que se utiliza para identificar las clases, sus responsabilidades y cómo colaboran entre sí. Para un diseño simple, se pueden utilizar tarjetas CRC para comprender mejor las necesidades de los usuarios y las tareas que debe realizar el sistema, según se desarrolla en el capítulo II.

Tabla 53

Clase de Usuarios.

CLASE: USUARIOS	
Responsabilidades: Administrar Usuario del Sistema Iniciar sesión de jueces Iniciar sesión de administrador Consultar Usuarios del Sistema	Colaboradores: Atributos: DNI nombres apellidos nombre de usuario clave de usuario estado de usuario tipo de usuario

Fuente: Creado por el autor.

Tabla 54*Clase Juez de paz*

CLASE: JUEZ DE PAZ	
Responsabilidades: Administrar Juez de Paz Consultar Jueces de Paz	Colaboradores: Atributos: juez_id dni nombres apellidos tipo de juez

Fuente: Creado por el autor.

Tabla 55*Clase Juzgado de paz*

CLASE: JUZGADOS DE PAZ	
Responsabilidades: Administrar Juzgados de Paz Consultar Juzgados de Paz	Colaboradores: Categorías Ubigeos Atributos: nombre de juzgado teléfono dirección observaciones

Fuente: Creado por el autor.

Tabla 56*Clase Recurso material y tecnológico.*

CLASE: RECURSO MATERIAL Y TECNOLÓGICO	
Responsabilidades: Administrar Recursos Materiales y Tecnológicos Consultar Recursos Materiales y Tecnológicos	Colaboradores: Juzgado de Paz Atributos: nombre tipo serie código patrimonial color descripción observación

Fuente: Creado por el autor.

Tabla 57*Clase Asignación de ternas*

CLASE: ASIGNACIÓN DE TERNAS	
Responsabilidades: Asignar terna de jueces a juzgado Consultar Juzgados con y sin ternas Asignadas Consultar Juzgados con ternas próximos a vencer	Colaboradores: juez de paz juzgado de paz resolución administrativa Atributos:

Fuente: Creado por el autor.

Tabla 58*Clase Cuadro de aranceles*

CLASE: CUADRO DE ARANCELES	
Responsabilidades: Administrar Cuadro de Aranceles Consultar Cuadro de Aranceles	Colaboradores: Atributos: concepto de arancel costo de arancel

Fuente: Creado por el autor.

Tabla 59*Clase Emisión de certificado*

CLASE: EMISIÓN DE CERTIFICADOS	
Responsabilidades: Emisión de Certificado Escritura Pública de Transferencia de Bienes Inmuebles no registrables	Colaboradores: cuadro arancel juzgado de paz Atributos: nombre del template nombre del juez nombre de solicitante fecha de emisión costo arancel

Fuente: Creado por el autor.

La Figura 32 muestra el código fuente de la clase que permite conectarse a una base de datos. Las Figuras 33, 34 y 35 el código fuente de las clases entidad Usuario se muestra en la historia de usuario “Administrar Usuarios del Sistema”, para la clase “Actividad Administrar usuarios del sistema”.

Figura 32

Código fuente. Conexión a la base de datos (PostgreSQL)

```
const Pool =require ("pg").Pool;

const pool = new Pool({

  user: process.env.PG_USER ,

  host: process.env.PG_HOST ,

  database: process.env.PG_DATABASE,

  password: process.env.PG_PASSWORD ,

  port: process.env.PG_PORT

});

module.exports=pool;
```

Fuente: Creado por el autor.

Figura 33

Código fuente: usuario controller.

```
require('dotenv').config();

const bcrypt = require('bcrypt');

const jwt = require('jsonwebtoken');

const { respuestaExitosa, respuestaFallida } = require("./utils")

const pool = require('../db');

const queries = require('../models/usuarioModel');

const getUsuario = (req, res) => {
  pool.query(queries.getUsuario, (error, results) => {
    const respuesta = error ? respuestaFallida(error) : respuestaExitosa(results.rows);
    const codigoError = error ? 500 : 200;
    res.status(codigoError).json(respuesta)
  });
};
```

```

};
const getCheckUsuarioIdExist = (req, res) => {
  const usuario_id = parseInt(req.params.usuario_id);
  pool.query(queries.getCheckUsuarioIdExist, [usuario_id], (error, results) => {
    const respuesta = error ? respuestaFallida(error) : respuestaExitosa(results.rows);
    const codigoError = error ? 500 : 200;
    res.status(codigoError).json(respuesta)
  });
};
const addUsuario = async (req, res) => {
  const { dni, nombres, apellidos, nombre_usuario, clave_usuario, estado_usuario,
  tipo_usuario } = req.body;

  pool.query(queries.checkDniExists, [dni], (error, results) => {
    if (results.rows.length) {
      const nuevoError = { message: "not ok - Usuario already exists." }
      res.send(respuestaFallida(nuevoError))
      return;
    }
  })

  const nuevaClave = await bcrypt.hash(clave_usuario, 10);

  pool.query(queries.addUsuario,
    [dni, nombres, apellidos, nombre_usuario, nuevaClave, estado_usuario,
    tipo_usuario],
    (error, results) => {
      const respuesta = error ? respuestaFallida(error) : respuestaExitosa(null, "ok -
      Usuario creado exitosamente");
      const codigoError = error ? 500 : 200;
      res.status(codigoError).json(respuesta)
    } );
};
const removeUsuario = (req, res) => {
  const usuario_id = parseInt(req.params.usuario_id);

  pool.query(queries.getCheckUsuarioIdExist, [usuario_id], (error, results) => {
    const noUsuarioFound = !results.rows.length;
    if (noUsuarioFound) {
      const nuevoError = { message: "not ok - Usuario no Existe en la Base de Datos."
    }

    res.send(respuestaFallida(nuevoError));
    return;
  }

  pool.query(queries.removeUsuario[usuario_id], (error, results) => {
    const respuesta = error ? respuestaFallida(error) : respuestaExitosa(null, "ok -
    Usuario eliminado successfully.");
    const codigoError = error ? 500 : 200;

```

```

        res.status(codigoError).json(respuesta)
    });
});
};
const getUsuarioPorDni = (req, res) => {
    const dni = req.query.dni;
    pool.query(queries.getUsuarioPorDni(dni), (error, results) => {
        const respuesta = error ? respuestaFallida(error) : respuestaExitosa(results.rows);
        const codigoError = error ? 500 : 200;
        res.status(codigoError).json(respuesta)
    });};
const updateUsuario = (req, res) => {
    const usuarioId = parseInt(req.params.usuario_id);
    const { dni } = req.body;

    pool.query(queries.getCheckUsuarioIdExist, [usuarioId], (error, results) => {
        const noJuzgadoPazFound = !results?.rows?.length;
        if (noJuzgadoPazFound) {
            const nuevoError = { message: "not ok - Usuario no Existe en la Base de Datos."
        }

            res.send(respuestaFallida(nuevoError));
            return;
        }
        pool.query(queries.updateUsuario(req.body, usuarioId), Object.values(req.body),
(error, results) => {
            const respuesta = error ? respuestaFallida(error) : respuestaExitosa(null, "ok -
Usuario Update successfully.");
            const codigoError = error ? 500 : 200;
            res.status(codigoError).json(respuesta)
        });
    });
};
const authUsuario = (req, res) => {
    const { nombre_usuario, clave_usuario } = req.body;

    pool.query(queries.authUsuario, [nombre_usuario], async (error, results) => {
        const usuario = results?.rows?.[0];

        let respuesta = error ? respuestaFallida(error) : respuestaExitosa(results.rows);
        let codigoError = error ? 500 : 200;
        if (!usuario) {
            respuesta = 'Usuario registrado no existe';
            codigoError = 400;
        }

        if (usuario) {
            const isValid = await bcrypt.compare(clave_usuario, usuario.clave_usuario);

            if (!isValid) {

```

```

        respuesta = 'La contraseña es incorrecta.';
        codigoError = 400;
    } else {
        delete usuario.clave_usuario;
        respuesta = respuestaExitosa([usuario]);
        // const accessToken = jwt.sign(usuario,
process.env.ACCESS_TOKEN_SECRET);
        // res.json({ accessToken});
        // return;
    }
}
res.status(codigoError).json(respuesta)
});
};
module.exports = {
    getUsuario,
    getCheckUsuarioIdExist,
    addUsuario,
    removeUsuario,
    updateUsuario,
    authUsuario,
    getUsuarioPorDni};

```

Fuente: Creado por el autor.

Figura 34

Código fuente: usuario model.

```

const getUsuario = "SELECT * FROM usuarios";
const getCheckUsuarioIdExist = "SELECT * FROM usuarios WHERE usuario_id=$1";
const checkDniExists = "SELECT j FROM usuarios j WHERE j.dni=$1";
const addUsuario = "INSERT INTO usuarios(dni, nombres, apellidos, nombre_usuario,
clave_usuario, estado_usuario,tipo_usuario) values($1,$2,$3,$4,$5,$6,$7)";
const removeUsuario = "DELETE FROM usuarios WHERE usuario_id=$1";
const authUsuario = "select * from usuarios where nombre_usuario=$1";
const getUsuarioPorDni = (terminoBusqueda) => {
    const query = `SELECT * FROM usuarios WHERE dni ilike
'${terminoBusqueda}%' or nombres ilike '${terminoBusqueda}%' or apellidos ilike
'${terminoBusqueda}%'`;
    return query;}
const updateUsuario = (usuario, usuarioId) => {
    const keys = Object.keys(usuario).join(',');
    const argKeys = Object.keys(usuario).map((obj, index) => `${index + 1}`).join(',');
    const query = `UPDATE usuarios SET (${keys})=(${argKeys}) WHERE
usuario_id=${usuarioId}`;
    return query;}
module.exports = {
    getUsuario,
    getCheckUsuarioIdExist,
    addUsuario,
    checkDniExists,

```

```
removeUsuario,  
updateUsuario,  
authUsuario,  
getUsuarioPorDni  
};
```

Fuente: Creado por el autor.

Figura 35

Código fuente: juez de paz model.

```
const getJuezPaz = "SELECT * FROM jueces_paz";  
const getCheckJuezIdExist = "SELECT * FROM jueces_paz WHERE juez_paz_id=$1";  
const checkDniExists = "SELECT j FROM jueces_paz j WHERE j.dni=$1";  
const addJuezPaz =  
  "INSERT INTO jueces_paz(dni, nombres, apellidos,teléfono,observacion)  
  values($1,$2,$3,$4,$5)";  
const removeJuezPaz = "DELETE FROM jueces_paz WHERE juez_paz_id=$1";  
const getJuezPazPorDni = (terminoBusqueda) => {  
  const query = `SELECT * FROM jueces_paz WHERE dni ilike  
  '%${terminoBusqueda}%' or nombres ilike '%${terminoBusqueda}%' or apellidos ilike  
  '%${terminoBusqueda}%'`;   
  return query;  
};  
  
const updateJuezPaz = (juezPaz, juezPazId) => {  
  const keys = Object.keys(juezPaz).join(",");  
  const argKeys = Object.keys(juezPaz)  
    .map((obj, index) => `${index + 1}`)  
    .join(",");  
  const query = `UPDATE jueces_paz SET (${keys})=(${argKeys}) WHERE  
  juez_paz_id=${juezPazId}`;  
  return query;  
};  
  
module.exports = {  
  
  getJuezPaz,  
  
  addJuezPaz,  
  
  getCheckJuezIdExist,  
  
  checkDniExists,  
  
  removeJuezPaz,  
  
  updateJuezPaz,  
  getJuezPazPorDni,};
```

Fuente: Creado por el autor.

Figura 36

Código fuente: juez de paz vista.

```
import React, { useEffect, useMemo, useRef, useState } from "react";
import Link from "next/link";
import axios from "axios";
import Table from "@components/Table";
import ActualizarJuzgadoPaz from "../actualizar";
import { useDispatch } from "react-redux";
import { agregarJuzgadoPaz, guardarJuzgadoPazSeleccionado,
} from "../juzgadoPazSlice";
import { useParams } from "react-router-dom";
import { Button, IconButton } from "@mui/material";
import { Add, Edit, Check, Search, Clear } from "@mui/icons-material";
import styles from "../../styles/JuezPaz.module.css";
import globals from "../../styles/Global.module.css";
import ArrowBackIcon from "@mui/icons-material/ArrowBack";
import SideBar from "@components/SideBar";
import BackButton from "@components/BackButton";
import SearchSegment from "@components/SearchSegment";
import EmptySearchResult from "@components/EmptySearchResult";
import { buscarJuzgadosPaz } from "@actions/juzgadoPaz";

const AdministrarJuzgadoPaz = () => {
  const [terminoBusqueda, setTerminoBusqueda] = useState("");
  const [juzgadosPaz, setJuzgadosPaz] = useState();
  const [selectedJuzgado, setSelectedJuzgado] = useState(null);

  const dispatch = useDispatch();
  const onChange = (e) => {
    const termino = e.target.value;
    if (!termino) setJuzgadosPaz([]);
    if (terminoBusqueda !== termino) setTerminoBusqueda(e.target.value);
  };

  const guardarJuzgadoPazSeleccionadoHandler = (juzgado) => {
    dispatch(guardarJuzgadoPazSeleccionado(juzgado));
  };

  useEffect(() => {
    if (!terminoBusqueda) setJuzgadosPaz([]);
    else buscarJuzgadosPaz({ termino: terminoBusqueda, setJuzgadosPaz });
  }, [terminoBusqueda]);

  useEffect(() => {
    guardarJuzgadoPazSeleccionadoHandler(selectedJuzgado);
  }, [selectedJuzgado]);

  const columnas = useRef([
    {
```

```

    Header: "departamento",
    accessor: "nombre_departamento",
  },
  {
    Header: "provincia",
    accessor: "nombre_provincia",
  },
  {
    Header: "distrito",
    accessor: "nombre_distrito",
  },

  {
    Header: "nombre",
    accessor: "nombre_juzgado",
  },
  {
    Header: "Direccion",
    accessor: "direccion",
  },
  {
    Header: "Telefono",
    accessor: "telefono",
  },
]);

const onSearch = (e) => {
  if (!terminoBusqueda) setJuzgadosPaz([]);
  else buscarJuzgadosPaz({ termino: terminoBusqueda, setJuzgadosPaz });
};

return (
  <div>
    <SideBar />
    <BackButton />
    <div className={styles.container}>
      <div className={globals.title_general}>
        ADMINISTRACIÓN DE JUZGADO DE PAZ
      </div>

      <div className={globals.form_container}>
        <SearchSegment
          placeholder="Buscar por nombre de juzgado"
          searchTerm={terminoBusqueda}
          setSearchTerm={setTerminoBusqueda}
          onChange={onChange}
          onSearch={onSearch}
          item={selectedJuzgado}
          showEdit={true}

```

```

    newPath="/juzgado-paz/registro"
    editPath="/juzgado-paz/actualizar"
  />
  {!juzgadosPaz?.length && !!terminoBusqueda && <EmptySearchResult />}
  {!!juzgadosPaz?.length && (
    <div>
      <div className={globals.title_subgeneral}>
        LISTA DE JUZGADOS DE PAZ
      </div>
      <Table
        data={juzgadosPaz}
        columns={columnas.current}
        setSelectedItem={setSelectedJuzgado}
      />
    </div>
  )}
</div>
</div>
</div>
);
};
export default AdministrarJuzgadoPaz;

```

Fuente: Creado por el autor.

Figura 37

Código fuente: juzgado de paz controller.

```

const { respuestaExitosa, respuestaFallida } = require("../utils")
const pool = require('.././db');
const queries = require('../models/juzgadoPazModel');

const getJuzgadoPaz = (req, res) => {
  pool.query(queries.getJuzgadoPaz, (error, results) => {
    const respuesta = error ? respuestaFallida(error) : respuestaExitosa(results.rows);
    const codigoError = error ? 500 : 200;
    res.status(codigoError).json(respuesta)
  }); };

const getJuzgadoPazPorNombre = (req, res) => {
  const nombre_juzgado = req.query.nombre_juzgado;
  pool.query(queries.getJuzgadoPazPorNombre(nombre_juzgado), (error, results) => {

    const respuesta = error ? respuestaFallida(error) : respuestaExitosa(results.rows);
    const codigoError = error ? 500 : 200;
    res.status(codigoError).json(respuesta)
  }); };

const getCheckJuzgadoPazIdExist = (req, res) => {
  const juzgado_paz_id = parseInt(req.params.juzgado_paz_id);
  pool.query(queries.getCheckJuzgadoPazIdExist, [juzgado_paz_id], (error, results) => {

```

```

    if (error) throw error;
    const respuesta = error ? respuestaFallida(error) : respuestaExitosa(results.rows);
    const codigoError = error ? 500 : 200;
    res.status(codigoError).json(respuesta) });});

const addJuzgadoPaz = (req, res) => {
  const { telefono, direccion, categoria_id, nombre_juzgado, observaciones, ubigeo_id }
  = req.body;

  pool.query(queries.checkNombreJuzgadoExists, [nombre_juzgado], (error, results) => {
    if (results?.rows?.length) {
      const nuevoError = { message: "not ok - Juzgado already exists." }
      res.send(respuestaFallida(nuevoError))
      return;    } })

  pool.query(queries.addJuzgadoPaz,
    [telefono, direccion, categoria_id, nombre_juzgado, observaciones, ubigeo_id],
    (error, results) => {
      const respuesta = error ? respuestaFallida(error) : respuestaExitosa(null, "ok -
  Juzgado creado exitosamente");
      const codigoError = error ? 500 : 200;
      res.status(codigoError).json(respuesta)
    } ); });

const removeJuzgadoPaz = (req, res) => {
  const juzgado_paz_id = parseInt(req.params.juzgado_paz_id);

  pool.query(queries.getCheckJuzgadoPazIdExist, [juzgado_paz_id], (error, results) => {
    const noJuezPazFound = !results.rows.length;
    if (noJuezPazFound) {
      const nuevoError = { message: "not ok - Juzgado de Paz no Existe en la Base de
  Datos." }
      res.send(respuestaFallida(nuevoError))
      return;    }

    pool.query(queries.removeJuzgadoPaz, [juzgado_paz_id], (error, results) => {
      if (error) throw error;
      const respuesta = error ? respuestaFallida(error) : respuestaExitosa(null, "ok -
  Juzgado de paz eliminado successfully.");
      const codigoError = error ? 500 : 200;
      res.status(codigoError).json(respuesta)    }); }); });

const updateJuzgadoPaz = (req, res) => {
  const juzgadoPazId = parseInt(req.params.juzgado_paz_id);
  const { nombre_juzgado } = req.body;
  pool.query(queries.getCheckJuzgadoPazIdExist, [juzgadoPazId], (error, results) => {
    const noJuzgdoPazFound = !results?.rows?.length;
    if (noJuzgdoPazFound) {
      const nuevoError = { message: "not ok - Juzgado de Paz no Existe en la Base de
  Datos." }
      res.send(respuestaFallida(nuevoError))
    }
  })
}

```

```

        return;    }
        pool.query(queries.updateJuzgadoPaz(req.body,                juzgadoPazId),
Object.values(req.body), (error, results) => {
        const respuesta = error ? respuestaFallida(error) : respuestaExitosa(null, "ok -
Juzgado de paz actualizado correctamente.");
        const codigoError = error ? 500 : 200;
        res.status(codigoError).json(respuesta)
    }); }); };
module.exports = {
    getJuzgadoPaz,
    getCheckJuzgadoPazIdExist,
    addJuzgadoPaz,
    removeJuzgadoPaz,
    updateJuzgadoPaz,
    getJuzgadoPazPorNombre};

```

Fuente: Creado por el autor.

Figura 38

Código fuente: juez de paz model.

```

const getJuzgadoPaz = "SELECT * FROM juzgados_paz";
const getCheckJuzgadoPazIdExist = "SELECT * FROM juzgados_paz WHERE
juzgado_paz_id=$1";
const checkNombreJuzgadoExists = "SELECT j FROM juzgados_paz j WHERE
j.nombre_juzgado=$5";
const addJuzgadoPaz = "INSERT INTO juzgados_paz(telefono, direccion, categoria_id,
nombre_juzgado, observaciones, ubigeo_id) values($1,$2,$3,$4,$5,$6)";
const getJuzgadoPazPorNombre=(terminodebusqueda)=>{
const query=`SELECT u.nombre_departamento,
u.nombre_provincia,
u.nombre_distrito,
jp.nombre_juzgado,
jp.direccion,
jp.telefono,
u.ubigeo_id,
jp.categoria_id,
jp.juzgado_paz_id,
jp.observaciones
FROM juzgados_paz jp
LEFT JOIN ubigeos u ON jp.ubigeo_id=u.ubigeo_id
WHERE nombre_juzgado ilike "%${terminodebusqueda}%`;
return query; }
const updateJuzgadoPaz=(juzgado,juzgadoPazId)=> {
    const args=Object.values(juzgado);
    const keys=Object.keys(juzgado).join(',');

    const argKeys=Object.keys(juzgado).map((obj,index)=>`${index+1}`).join(',');

    const query=`UPDATE juzgados_paz SET (${keys})= (${argKeys}) WHERE

```

```

juzgado_paz_id=${juzgadoPazId}`;

    return query; }

module.exports = {

    getJuzgadoPaz,

    addJuzgadoPaz,

    getCheckJuzgadoPazIdExist,
    checkNombreJuzgadoExists,

    updateJuzgadoPaz,
    getJuzgadoPazPorNombre
}

```

Fuente: Creado por el autor.

Figura 39

Código Fuente: juez de paz vista.

```

import React, { useEffect, useMemo, useRef, useState } from "react";
import Link from "next/link";
import axios from "axios";
import Table from "@components/Table";
import ActualizarJuzgadoPaz from "../actualizar";
import { useDispatch } from "react-redux";
import {
    agregarJuzgadoPaz,
    guardarJuzgadoPazSeleccionado,
} from "../juzgadoPazSlice";
import { useParams } from "react-router-dom";
import { Button, IconButton } from "@mui/material";
import { Add, Edit, Check, Search, Clear } from "@mui/icons-material";
import styles from "../../styles/JuezPaz.module.css";
import globals from "../../styles/Global.module.css";
import ArrowBackIcon from "@mui/icons-material/ArrowBack";
import SideBar from "@components/SideBar";
import BackButton from "@components/BackButton";
import SearchSegment from "@components/SearchSegment";
import EmptySearchResult from "@components/EmptySearchResult";
import { buscarJuzgadosPaz } from "@actions/juzgadoPaz";

const AdministrarJuzgadoPaz = () => {
    const [terminoBusqueda, setTerminoBusqueda] = useState("");
    const [juzgadosPaz, setJuzgadosPaz] = useState();
    const [selectedJuzgado, setSelectedJuzgado] = useState(null);

    const dispatch = useDispatch();

```

```

const onChange = (e) => {
  const termino = e.target.value;
  if (!termino) setJuzgadosPaz([]);
  if (terminoBusqueda !== termino) setTerminoBusqueda(e.target.value);
};

const guardarJuzgadoPazSeleccionadoHandler = (juzgado) => {
  dispatch(guardarJuzgadoPazSeleccionado(juzgado));
};

useEffect(() => {
  if (!terminoBusqueda) setJuzgadosPaz([]);
  else buscarJuzgadosPaz({ termino: terminoBusqueda, setJuzgadosPaz });
}, [terminoBusqueda]);

useEffect(() => {
  guardarJuzgadoPazSeleccionadoHandler(selectedJuzgado);
}, [selectedJuzgado]);

const columnas = useRef([
  {
    Header: "departamento",
    accessor: "nombre_departamento",
  },
  {
    Header: "provincia",
    accessor: "nombre_provincia",
  },
  {
    Header: "distrito",
    accessor: "nombre_distrito",
  },
  {
    Header: "nombre",
    accessor: "nombre_juzgado",
  },
  {
    Header: "Direccion",
    accessor: "direccion",
  },
  {
    Header: "Telefono",
    accessor: "telefono",
  },
]);

const onSearch = (e) => {
  if (!terminoBusqueda) setJuzgadosPaz([]);

```

```

else buscarJuzgadosPaz({ termino: terminoBusqueda, setJuzgadosPaz });
};

return (
  <div>
    <SideBar />
    <BackButton />
    <div className={styles.container}>
      <div className={globals.title_general}>
        ADMINISTRACIÓN DE JUZGADO DE PAZ
      </div>
      <div className={globals.form_container}>
        <SearchSegment
          placeholder="Buscar por nombre de juzgado"
          searchTerm={terminoBusqueda}
          setSearchTerm={setTerminoBusqueda}
          onChange={onChange}
          onSearch={onSearch}
          item={selectedJuzgado}
          showEdit={true}
          newPath="/juzgado-paz/registro"
          editPath="/juzgado-paz/actualizar"
        />
        {!juzgadosPaz?.length && !!terminoBusqueda && <EmptySearchResult />}
        {!!juzgadosPaz?.length && (
          <div>
            <div className={globals.title_subgeneral}>
              LISTA DE JUZGADOS DE PAZ
            </div>
            <Table
              data={juzgadosPaz}
              columns={columnas.current}
              setSelectedItem={setSelectedJuzgado}
            />
          </div>
        )}
      </div>
    </div>
  );
};
export default AdministrarJuzgadoPaz;

```

Fuente: Creado por el autor.

Figura 40

Código Fuente: Recurso Material y Tecnológico Controller.

```
const {respuestaExitosa,respuestaFallida} = require("../utils")
const pool = require('../db');
const queries = require('../models/recursoModel');

const getRecurso = (req, res) => {
  pool.query(queries.getRecurso, (error, results) => {

    const respuesta = error ? respuestaFallida(error) : respuestaExitosa(results.rows);
    const codigoError = error ? 500 : 200;
    res.status(codigoError).json(respuesta)
  });};

const getCheckRecursoIdExist = (req, res) => {
  const recurso_id = parseInt(req.params.recurso_id);
  pool.query(queries.getCheckRecursoIdExist, [recurso_id], (error, results) => {

    const respuesta = error ? respuestaFallida(error) : respuestaExitosa(results.rows);
    const codigoError = error ? 500 : 200;
    res.status(codigoError).json(respuesta)
  });};

const addRecurso = (req, res) => {
  const { nombre, tipo, serie, codigo_patrimonial, color, descripcion,
observacion,juzgado_paz_id } = req.body;
  pool.query(queries.checkSerieExists, [serie], (error, results) => {
    if (results?.rows?.length) {
      const nuevoError = { message: "not ok - Serie already exists." }
      res.send(respuestaFallida(nuevoError))
      return; }
    pool.query(queries.addRecurso,
      [nombre, tipo, serie, codigo_patrimonial, color, descripcion,
observacion,juzgado_paz_id],
      (error, results) => {
        const respuesta = error ? respuestaFallida(error) : respuestaExitosa(null, "ok -
Bien Patrimonial creado exitosamente");
        const codigoError = error ? 500 : 200;
        res.status(codigoError).json(respuesta)
      } );});};

const getRecursoPorNombre = (req, res) => {
  const nombre = req.query.nombre;
  pool.query(queries.getRecursoPorNombre(nombre), (error, results) => {
    const respuesta = error ? respuestaFallida(error) : respuestaExitosa(results.rows);
    const codigoError = error ? 500 : 200;
    res.status(codigoError).json(respuesta)
  });};

const removeRecurso = (req, res) => {
  const recurso_id = parseInt(req.params.recurso_id);
  pool.query(queries.getCheckRecursoIdExist, [recurso_id], (error, results) => {
    const noRecursoFound = !results?.rows?.length;
```

```

        if (noRecursoFound) {
            const nuevoError = { message: "not ok - Bien Patrimonial no Existe en la Base de
Datos." }
            res.send(respuestaFallida(nuevoError))
            return; }
        pool.query(queries.removeRecurso, [recurso_id], (error, results) => {
            const respuesta = error ? respuestaFallida(error) : respuestaExitosa(null, "ok -
Bien Patrimonial eliminado successfully.");
            const codigoError = error ? 500 : 200;
            res.status(codigoError).json(respuesta)
        }); });};

const updateRecurso = (req, res) => {
    const recursoId = parseInt(req.params.recurso_id);
    const { nombre_categoria } = req.body;
    pool.query(queries.getCheckRecursoIdExist, [recursoId], (error, results) => {
        const noRecursoFound = !results?.rows?.length;
        if (noRecursoFound) {
            const nuevoError = { message: "not ok - Bien Patrimonial no Existe en la Base de
Datos." }
            res.send(respuestaFallida(nuevoError))
            return; }
        pool.query(queries.updateRecurso(req.body, recursoId), Object.values(req.body),
(error, results) => {
            const respuesta = error ? respuestaFallida(error) : respuestaExitosa(null, "ok -
Bien Patrimonial Updated successfully.");
            const codigoError = error ? 500 : 200;
            res.status(codigoError).json(respuesta)        }); }); });
module.exports = {
    getRecurso,
    getCheckRecursoIdExist,
    addRecurso,
    removeRecurso,
    updateRecurso,
    getRecursoPorNombre
};

```

Fuente: Creado por el autor.

Figura 41

Código Fuente: Recurso Material y Tecnológico Model.

```

const getcategoria = "SELECT * FROM categorias";
const getCheckCategoriaIdExist = "SELECT * FROM categorias WHERE
categoria_id=$1";
const addCategoria = "INSERT INTO categorias(nombre_categoria) values($1)";
const removeCategoria = "DELETE FROM categorias WHERE categoria_id=$1";
const updateCategoria = "UPDATE categorias SET nombre_categoria=$1 WHERE
categoria_id=$2";
module.exports = {

```

```
getcategoria,  
getCheckCategoriaIdExist,  
addCategoria,  
removeCategoria,  
updateCategoria};
```

Fuente: Creado por el autor.

Figura 42

Código Fuente. Recurso Material y Tecnológico Vista.

```
import react, { useEffect, useMemo, useRef, useState } from "react";  
import Link from "next/link";  
import axios from "axios";  
import Table from "../../components/Table";  
import ActualizarRecurso from "../actualizar";  
import { useDispatch } from "react-redux";  
import {  
  agregarRecurso,  
  guardarRecursoSeleccionado,  
} from "../recursoSlice";  
import { Button, IconButton } from "@mui/material";  
import { Add, Edit, Check, Search, Clear, Close } from "@mui/icons-material";  
  
import globals from "../../styles/Global.module.css";  
import styles from "../../styles/JuezPaz.module.css";  
  
import SideBar from "@components/SideBar";  
import BackButton from "@components/BackButton";  
import ArrowBackIcon from "@mui/icons-material/ArrowBack";  
import SearchSegment from "@components/SearchSegment";  
import EmptySearchResult from "@components/EmptySearchResult";  
import { buscarRecurso } from "@actions/recurso";  
  
const AdministrarRecurso = () => {  
  const [terminoBusqueda, setTerminoBusqueda] = useState("");  
  const [recursos, setRecursos] = useState();  
  const [selectedRecurso, setSelectedRecurso] = useState(null);  
  
  const dispatch = useDispatch();  
  const onChange = (e) => {  
    const termino = e.target.value;  
    if (!termino) setRecursos([]);  
    if (terminoBusqueda !== termino) setTerminoBusqueda(e.target.value);  
  };  
  const onSearch = () => {  
    if (!terminoBusqueda) setRecursos([]);  
    else buscarRecurso({ termino: terminoBusqueda, setRecursos });  
  };  
};
```

```

const guardarRecursoSeleccionadoHandler = (recurso) => {
  dispatch(guardarRecursoSeleccionado(recurso));
};

useEffect(() => {
  if (!terminoBusqueda) setRecursos([]);
  else buscarRecurso({ termino: terminoBusqueda, setRecursos });
}, [terminoBusqueda]);

useEffect(() => {
  guardarRecursoSeleccionadoHandler(selectedRecurso);
}, [selectedRecurso]);

const columnas = useRef([
  {
    Header: "JUZGADO",
    accessor: "nombre_juzgado",
  },
  {
    Header: "NOMBRE ",
    accessor: "nombre",
  },
  {
    Header: "TIPO ",
    accessor: "tipo",
  },
  {
    Header: "SERIE ",
    accessor: "serie",
  },
  {
    Header: "DESCRIPCION ",
    accessor: "descripcion",
  },
  {
    Header: "OBSERACION ",
    accessor: "observacion",
  },
]);

return (
  <div>
    <SideBar />
    <BackButton />
    <div className={styles.container}>
      <div className={globals.title_general}>
        ADMINISTRACIÓN DE RECURSOS MATERIALES Y TECNOLÓGICOS
      </div>
      <div className={globals.form_container}>

```

```

<SearchSegment
placeholder="Buscar por nombre de recurso"
searchTerm={terminoBusqueda}
setSearchTerm={setTerminoBusqueda}
onChange={onChange}
onSearch={onSearch}
item={selectedRecurso}
showEdit={true}
newPath="/recurso/registro"
editPath="/recurso/actualizar"
/>
{!recursos?.length && !terminoBusqueda && (
<EmptySearchResult />
)}

{!recursos?.length && (
<div>
<div className={globals.title_subgeneral}>
LISTA BIEN PATRIMONIAL
</div>
<Table
data={recursos}
columns={columnas.current}
setSelectedItem={setSelectedRecurso}
/>
</div>
)}
</div>
</div>
</div> ); };
export default AdministrarRecurso;

```

Fuente: Creado por el autor.

Figura 43

Código Fuente: Certificados Emitidos Controller.

```

const {respuestaExitosa,respuestaFallida} = require("./utils")
const pool = require('../db');
const queries = require('../models/certificadoEmitidoModel');
const getCertificadosEmitidos = (req, res) => {
  pool.query(queries.getCertificadosEmitidos, (error, results) => {
    const respuesta = error ? respuestaFallida(error) : respuestaExitosa(results.rows);
    const codigoError = error ? 500 : 200;
    res.status(codigoError).json(respuesta)
  });};
const getCertificadoPorNombre = (req, res) => {
  const nombre_template = req.query.nombre_template;
  pool.query(queries.getCertificadoPorNombre(nombre_template), (error, results)
=> {

```

```

    const respuesta = error ? respuestaFallida(error) : respuestaExitosa(results.rows);
    const codigoError = error ? 500 : 200;
    res.status(codigoError).json(respuesta)
  });};

const getCheckcertificadosEmitidosIdExist = (req, res) => {
  const certificados_emitidos_id = parseInt(req.params.certificados_emitidos_id);
  pool.query(queries.getCheckcertificadosEmitidosIdExist,
[certificados_emitidos_id], (error, results) => {
    const respuesta = error ? respuestaFallida(error) : respuestaExitosa(results.rows);
    const codigoError = error ? 500 : 200;
    res.status(codigoError).json(respuesta)
  }); };

const addCertificadosEmitidos = (req, res) => {
  const { nombre_template, nombre_juez, solicitante,
certificado_pdf, costo_arancel, juzgado_paz_id, cuadro_arancel_id } = req.body;
  pool.query(queries.addCertificadosEmitidos,
[nombre_template, nombre_juez, solicitante,
certificado_pdf, costo_arancel, juzgado_paz_id, cuadro_arancel_id],
(error, results) => {
    const respuesta = error ? respuestaFallida(error) : respuestaExitosa(null, "ok -
certificados emitidos creado exitosamente");
    const codigoError = error ? 500 : 200;
    res.status(codigoError).json(respuesta)
  } ); };

const removecertificadosEmitidos = (req, res) => {
  const certificados_emitidos_id = parseInt(req.params.certificados_emitidos_id);
  pool.query(queries.getCheckcertificadosEmitidosIdExist,
[certificados_emitidos_id], (error, results) => {
    const nocertificadosEmitidosFound = !results.rows.length;
    if (nocertificadosEmitidosFound) {
      const nuevoError = { message: "not ok - certificados no Existe en la Base de
Datos." }
      res.send(respuestaFallida(nuevoError))
      return;
    }
    pool.query(queries.removecertificadosEmitidos, [certificados_emitidos_id],
(error, results) => {
      if (error) throw error;
      const respuesta = error ? respuestaFallida(error) : respuestaExitosa(null, "ok -
certificados eliminado successfully.");
      const codigoError = error ? 500 : 200;
      res.status(codigoError).json(respuesta)
    }); } ); };

const updatecertificadosEmitidos = (req, res) => {
  const certificadosEmitidosId = parseInt(req.params.certificados_emitidos_id);
  const { solicitante } = req.body;

```

```

    pool.query(queries.getCheckcertificadosEmitidosIdExist,
[certificadosEmitidosId], (error, results) => {
    const nocertificadosEmitidosFound = !results?.rows?.length;
    if (nocertificadosEmitidosFound) {
    const nuevoError = { message: "not ok - certificados emitidos no Existe en la
Base de Datos." }
    res.send(respuestaFallida(nuevoError))
    return;    }
    pool.query(queries.updatecertificadosEmitidos(req.body, certificadosEmitidosId),
Object.values(req.body),(error,resuts) => {
    const respuesta = error ? respuestaFallida(error) : respuestaExitosa(null, "ok -
certificados emitidos Update successfully.");
    const codigoError = error ? 500 : 200;
    res.status(codigoError).json(respuesta)
    }); });
module.exports = {

    getCertificadosEmitidos,
    getCheckcertificadosEmitidosIdExist,
    addCertificadosEmitidos,
    removecertificadosEmitidos,
    updatecertificadosEmitidos,
    getCertificadoPorNombre};

```

Fuente: Creado por el autor.

Figura 44

Código Fuente: Certificados Emitidos Model.

```

const getCertificadosEmitidos = "select * from certificados_emitidos";
const getCheckcertificadosEmitidosIdExist = "SELECT * FROM certificados_emitidos
WHERE solicitante=$1";
const      addCertificadosEmitidos      =      "INSERT      INTO
certificados_emitidos(nombre_template, nombre_juez, solicitante, certificado_pdf,
costo_arancel,juzgado_paz_id,cuadro_arancel_id) values($1,$2,$3,$4,$5,$6,$7)";
const removecertificadosEmitidos = "DELETE FROM certificados_emitidos WHERE
certificados_emitidos_id=$1";
const updatecertificadosEmitidos=(certificadosEmitidos,certificadosEmitidosId)=>{
    const args=Object.values(certificadosEmitidos);
    const keys=Object.keys(certificadosEmitidos).join(',');
    const
argKeys=Object.keys(certificadosEmitidos).map((obj,index)=>`$$${index+1}`).join(',');
    const query=`UPDATE certificados_emitidos SET (${keys})=(${argKeys})
WHERE certificados_emitidos_id=${certificadosEmitidosId}`;
    return query;  }
const getCertificadoPorNombre=(terminodebusqueda)=>{
    const query=` SELECT
ce.nombre_template,
ce.nombre_juez,
ce.solicitante,

```

```

        ce.fecha_emision,
        ce.certificado_pdf,
        ce.cuadro_arancel_id,
        ce.costo_arancel
    FROM certificados_emitidos ce
    WHERE nombre_template ilike '%${terminodebusqueda}%`;
    return query;
    }
module.exports = {
    getCertificadosEmitidos,
    getCheckcertificadosEmitidosIdExist,
    addCertificadosEmitidos,
    removecertificadosEmitidos,
    updatecertificadosEmitidos,
    getCertificadoPorNombre};

```

Fuente: Creado por el autor.

Figura 45

Código Fuente: Certificados Emitidos Vista.

```

import react, { useEffect, useMemo, useRef, useState } from "react";
import axios from "axios";
import Table from "../../components/Table";
import { useDispatch } from "react-redux";
import { guardarCertificadoSeleccionado } from "../certificadoSlice";

import globals from "../../styles/Global.module.css";
import styles from "../../styles/JuezPaz.module.css";

import SideBar from "@components/SideBar";
import BackButton from "@components/BackButton";
import SearchSegment from "@components/SearchSegment";
import EmptySearchResult from "@components/EmptySearchResult";
import { buscarCertificadoEmitido } from "@actions/emisionCertificado";

const AdministrarCertificados = () => {
    const [terminoBusqueda, setTerminoBusqueda] = useState("");
    const [certificadosEmitidos, setCertificadosEmitidos] = useState();
    const [selectedCertificado, setSelectedCertificado] = useState(null);

    const dispatch = useDispatch();
    const onChange = e => {
        const termino = e.target.value;
        if (!termino) setCertificadosEmitidos([]);
        if (terminoBusqueda !== termino) setTerminoBusqueda(e.target.value);
    }
    const onSearch = () => {
        if (!terminoBusqueda) setCertificadosEmitidos([])
        else buscarCertificadoEmitido({ termino: terminoBusqueda,

```

```

setCertificadosEmitidos });
};

const guardarCertificadoSeleccionadoHandler = (certificado) => {
    dispatch(guardarCertificadoSeleccionado(certificado));
};

useEffect(() => {
    if (!terminoBusqueda) setCertificadosEmitidos([])
    else buscarCertificadoEmitido({ termino: terminoBusqueda,
setCertificadosEmitidos });
}, [terminoBusqueda])

useEffect(() => {
    guardarCertificadoSeleccionadoHandler(selectedCertificado)
}, [selectedCertificado])

const columnas = useRef([
    {
        Header: "NOMBRE TEMPLATE",
        accessor: "nombre_template",
    },
    {
        Header: 'JUEZ DE PAZ ',
        accessor: 'nombre_juez'
    },
    {
        Header: 'SOLICITANTE ',
        accessor: 'solicitante'
    },
    {
        Header: 'FECHA DE EMISION',
        accessor: 'fecha_emision'
    },
    {
        Header: 'ARANCEL ',
        accessor: 'costo_arancel'
    }
])

return (
    <div>
        <SideBar />
        <BackButton />
        <div className={styles.container}>
            <div className={globals.title_general}>
                ADMINISTRACIÓN EMISION DE CERTIFICADOS
            </div>
            <div className={globals.form_container}>

```

```

    <SearchSegment
      placeholder="Buscar por nombre de certificado"
      searchTerm={terminoBusqueda}
      setSearchTerm={setTerminoBusqueda}
      onChange={onChange}
      onSearch={onSearch}
      item={selectedCertificado}
      showEdit={false}
      newPath="/emision-certificado/registro"
      editPath="/emision-certificado/actualizar"
    />
    {!!certificadosEmitidos?.length && !!terminoBusqueda && <EmptySearchResult
  />}

  {!!certificadosEmitidos?.length && (
    <div>
      <div className={globals.title_subgeneral}>
        LISTA CERTIFICADOS EMITIDOS
      </div>
      <Table
        data={certificadosEmitidos}
        columns={columnas.current}
        setSelectedItem={setSelectedCertificado}
      />
    </div>
  )}
</div>
</div>
</div> );};
export default AdministrarCertificados;

```

Fuente: Creado por el autor.

Figura 46

Código Fuente: Asignar Juez Controller.

```

const { respuestaExitosa, respuestaFallida } = require("../utils")
const pool = require("../db");
const queries = require("../models/asignacionJuezModel");

const getAsignacionJuez = (req, res) => {
  pool.query(queries.getAsignacionJuez, (error, results) => {

    const respuesta = error ? respuestaFallida(error) : respuestaExitosa(results.rows);
    const codigoError = error ? 500 : 200;
    res.status(codigoError).json(respuesta)
  });
};

const buscarAsignacionJuez = (req, res) => {
  const termino = req.query.termino;

```

```

    pool.query(queries.buscarAsignacionJuez(termino), (error, results) => {
      const respuesta = error ? respuestaFallida(error) : respuestaExitosa(results.rows);
      const codigoError = error ? 500 : 200;
      res.status(codigoError).json(respuesta)
    });
  };
  const buscarAsignacionJuezPorVigencia = (req, res) => {
    const vigencia = req.query.vigencia;

    pool.query(queries.buscarAsignacionJuezPorVigencia(vigencia), (error, results)
=> {
      const respuesta = error ? respuestaFallida(error) : respuestaExitosa(results.rows);
      const codigoError = error ? 500 : 200;
      res.status(codigoError).json(respuesta)
    });
  };

  const getCheckJuzgadoIdExist = (req, res) => {
    const juzgado_paz_id = parseInt(req.params.juzgado_paz_id);
    pool.query(queries.getCheckJuzgadoIdExist, [juzgado_paz_id], (error, results)
=> {
      const respuesta = error ? respuestaFallida(error) : respuestaExitosa(results.rows);
      const codigoError = error ? 500 : 200;
      res.status(codigoError).json(respuesta)
    });
  };

  const getAsignacionPorJuezId = (req, res) => {

    const juez_titular_id = parseInt(req.params.juez_titular_id);
    pool.query(queries.getAsignacionPorJuezId, [juez_titular_id], (error, results) =>
{
      const respuesta = error ? respuestaFallida(error) : respuestaExitosa(results.rows);
      const codigoError = error ? 500 : 200;
      res.status(codigoError).json(respuesta)
    });
  };

  const getAsignacionPorJuzgadoId = (req, res) => {
    const juzgado_paz_id = parseInt(req.params.juzgado_paz_id);
    pool.query(queries.getAsignacionPorJuzgadoId, [juzgado_paz_id], (error, results)
=> {
      const respuesta = error ? respuestaFallida(error) : respuestaExitosa(results.rows);
      const codigoError = error ? 500 : 200;
      res.status(codigoError).json(respuesta)
    });
  };

  const getAsignacionJuezPorId = (req, res) => {
    const asignacion_juez_id = parseInt(req.params.asignacion_juez_id);
    pool.query(queries.getAsignacionJuezPorId, [asignacion_juez_id], (error, results)
=> {

```

```

        const respuesta = error ? respuestaFallida(error) : respuestaExitosa(results.rows);
        const codigoError = error ? 500 : 200;
        res.status(codigoError).json(respuesta)
    });
};

const addAsignacionJuez = (req, res) => {
    const { juez_titular_id, accesitario1_id, accesitario2_id, juzgado_paz_id,
    resolucion_administrativa_id, estatus } = req.body;

    pool.query(queries.checkJuzgadoIdExist, [juzgado_paz_id], (error, results) => {
        if (results.rows.length) {
            const nuevoError = { message: "not ok - Asignacion already exists." }
            res.send(respuestaFallida(nuevoError))
            return;
        }
        pool.query(queries.addAsignacionJuez,
            [juez_titular_id, accesitario1_id, accesitario2_id, juzgado_paz_id,
            resolucion_administrativa_id, estatus],
            (error, results) => {
                const respuesta = error ? respuestaFallida(error) : respuestaExitosa(null,
                "ok - Asignacion creado exitosadamente");
                const codigoError = error ? 500 : 200;
                res.status(codigoError).json(respuesta)
            }
        );
    })
};

const removeAsignacionJuez = (req, res) => {
    const asignacion_juez_id = parseInt(req.params.asignacion_juez_id);

    pool.query(queries.getCheckJuzgadoIdExist, [asignacion_juez_id], (error,
    results) => {
        const noAsignarJuezFound = !results.rows.length;
        if (noAsignarJuezFound) {
            const nuevoError = { message: "not ok - Asignacion no Existe en la Base de
            Datos." }
            res.send(respuestaFallida(nuevoError))
            return;
        }

        pool.query(queries.removeAsignacionJuez, [asignacion_juez_id], (error, results)
        => {
            const respuesta = error ? respuestaFallida(error) : respuestaExitosa(null, "ok -
            Asignacion eliminado successfully.");
            const codigoError = error ? 500 : 200;
            res.status(codigoError).json(respuesta)
        });
    });
};

```

```

const updateAsignacionJuez = (req, res) => {
  const asignacion_juez_id = parseInt(req.params.asignacion_juez_id);
  const { nombres } = req.body;

  pool.query(queries.getCheckJuzgadoIdExist, [juzgado_paz_id], (error, results)
=> {
    const noAsignarJuezFound = !results.rows.length;
    if (noAsignarJuezFound) {
      const nuevoError = { message: "not ok - Asignacion no Existe en la Base de
Datos." }
      res.send(respuestaFallida(nuevoError));
      return;
    }

    pool.query(queries.updateAsignacionJuez, [asignacion_juez_id], (error, results)
=> {
      const respuesta = error ? respuestaFallida(error) : respuestaExitosa(null, "ok -
Asignacion Updated Successfully");
      const codigoError = error ? 500 : 200;
      res.status(codigoError).json(respuesta)
    });
  });

module.exports = {
  getAsignacionJuez,
  buscarAsignacionJuezPorVigencia,
  buscarAsignacionJuez,
  addAsignacionJuez,
  getCheckJuzgadoIdExist,
  removeAsignacionJuez,
  updateAsignacionJuez,
  getAsignacionJuezPorId,
  getAsignacionPorJuezId,
  getAsignacionJuezPorId,
  getAsignacionPorJuzgadoId};

```

Fuente: Creado por el autor.

Figura 47

Codigo Fuente. Asignar Juez Model.

```

const addAsignacionJuez =
  "INSERT INTO asignacion_jueces(juez_titular_id,
primer_accesitario_id,segundo_accesitario_id,juzgado_paz_id,
resolucion_administrativa_id,estatus) values($1,$2,$3,$4,$5,$6)";
const getAsignacionJuez = "SELECT * FROM asignacion_jueces";
const getCheckJuzgadoIdExist =
  "SELECT j FROM asignacion_jueces j WHERE j.juzgado_paz_id=$1";
const checkJuzgadoIdExist =

```

```

"SELECT j FROM asignacion_jueces j where j.juzgado_paz_id=$1";
const removeAsignacionJuez =
  "DELETE FROM asignacion_jueces WHERE asignacion_juez_id=$1";
const updateAsignacionJuez =
  "UPDATE      asignacion_jueces      SET      juez_titular_1=$1      WHERE
asignacion_juez_id=$2";
const getAsignacionJuezPorId =
  "SELECT * FROM asignacion_jueces where asignacion_juez_id=$1";
const getAsignacionPorJuezId =
  "SELECT * FROM asignacion_jueces where juez_titular_id=$1 OR
primer_accesitario_id=$1 OR segundo_accesitario_id=$1";
const getAsignacionPorJuzgadoId =
  "SELECT * FROM asignacion_jueces where juzgado_paz_id=$1";

const buscarAsignacionJuez = (termino) => {
  const query = `
    SELECT
      aj.asignacion_juez_id,
      ra.resolucion_administrativa_id,
      ra.numero_resolucion,
      ra.fecha_inicio,
      ra.fecha_fin,
      (extract('day' from (fecha_fin-fecha_inicio))) as vigencia,
      jzt.nombres || ' ' || jzt.apellidos as titular,
      jza1.nombres || ' ' || jza1.apellidos as accesitario1,
      jza2.nombres || ' ' || jza2.apellidos as accesitario2,
      jo.nombre_juzgado
    FROM juzgados_paz jo
    LEFT JOIN asignacion_jueces aj ON jo.juzgado_paz_id = aj.juzgado_paz_id
    LEFT JOIN jueces_paz jzt ON jzt.juez_paz_id=aj.juez_titular_id
    LEFT JOIN jueces_paz jza1 ON jza1.juez_paz_id=aj.primer_accesitario_id
    LEFT JOIN jueces_paz jza2 ON jza2.juez_paz_id=aj.segundo_accesitario_id
    LEFT JOIN resoluciones_administrativas ra ON ra.resolucion_administrativa_id
= aj.resolucion_administrativa_id
    WHERE jo.nombre_juzgado ilike '%${termino}%'
    OR ra.numero_resolucion ilike '%${termino}%'
    OR jzt.nombres ilike '%${termino}%'
    OR jzt.apellidos ilike '%${termino}%' `;
  return query;};
const buscarAsignacionJuezPorVigencia = (vigencia) => {
  const query = `
    SELECT
      aj.asignacion_juez_id,
      ra.resolucion_administrativa_id,
      ra.numero_resolucion,
      ra.fecha_inicio,
      ra.fecha_fin,
      (extract('day' from (fecha_fin-NOW()::TIMESTAMP))) as vigencia,
      jzt.nombres || ' ' || jzt.apellidos as titular,

```

```

    jza1.nombres || ' ' || jza1.apellidos as accesitario1,
    jza2.nombres || ' ' || jza2.apellidos as accesitario2,
    jo.nombre_juzgado
FROM juzgados_paz jo
LEFT JOIN asignacion_jueces aj ON jo.juzgado_paz_id = aj.juzgado_paz_id
LEFT JOIN jueces_paz jzt ON jzt.juez_paz_id=aj.juez_titular_id
LEFT JOIN jueces_paz jza1 ON jza1.juez_paz_id=aj.primer_accesitario_id
LEFT JOIN jueces_paz jza2 ON jza2.juez_paz_id=aj.segundo_accesitario_id
LEFT JOIN resoluciones_administrativas ra ON ra.resolucion_administrativa_id =
aj.resolucion_administrativa_id
WHERE extract('day' from (fecha_fin-NOW())::TIMESTAMP))>=0
AND extract('day' from (fecha_fin-NOW())::TIMESTAMP))<= ${vigencia}
`;
return query;
};
module.exports = {
  getAsignacionJuez,
  buscarAsignacionJuez,
  buscarAsignacionJuezPorVigencia,
  addAsignacionJuez,
  getCheckJuzgadoIdExist,
  removeAsignacionJuez,
  updateAsignacionJuez,
  checkJuzgadoIdExist,
  getAsignacionJuezPorId,
  getAsignacionPorJuezId,
  getAsignacionPorJuzgadoId,
};

```

Fuente: Creado por el autor.

Figura 48

Código Fuente: Asignar Juez Vista.

```

import react, { useEffect, useMemo, useRef, useState } from "react";
import Link from "next/link";
import axios from "axios";
import Table from "../../components/Table";
import { useDispatch } from "react-redux";
import { guardarresolucionAdministrativaSeleccionado } from
"./resolucionAdministrativaSlice";
import {
  Button,
  FormControl,
  FormControlLabel,
  FormLabel,
  IconButton,
  Radio,
  RadioGroup,
  TextField,

```

```

} from "@mui/material";
import { Add, Edit, Check, Search, Clear } from "@mui/icons-material";
import styles from "../styles/JuezPaz.module.css";
import globals from "../styles/Global.module.css";
import asignacionStyles from "../styles/Asignacion.module.css";
import SideBar from "@components/SideBar";
import moment from "moment";
import { filters } from "../constants";
import BackButton from "@components/BackButton";
import { isValidNumber } from "../helper";
import EmptySearchResult from "@components/EmptySearchResult";
import { buscarAsignacionJuez, buscarAsignacionJuezPorDiferencia } from
"@actions/asignacionJuez";

const AdministrarResolucionAdministrativa = () => {
  const [terminoBusqueda, setTerminoBusqueda] = useState("");
  const [asignacionJueces, setAsignacionJueces] = useState([]);
  const [selectedResolucion, setSelectedResolucion] = useState(null);
  const [parsedAsignaciones, setParsedAsignaciones] = useState(null);
  const [vigencia, setVigencia] = useState(30);

  const [filter, setFilter] = useState(filters.TODO);

  const dispatch = useDispatch();
  const onChange = (e) => {
    const termino = e.target.value;
    if (!termino) setAsignacionJueces([]);
    if (terminoBusqueda !== termino) setTerminoBusqueda(e.target.value);
  };

  useEffect(() => {
    let asignacionesFiltradas = asignacionJueces;

    if (filter === filters.TODO) {
      asignacionesFiltradas = asignacionJueces;
    } else if (filter === filters.ASIGNADOS) {
      asignacionesFiltradas = asignacionJueces.filter(
        (asignacion) => asignacion.asignacion_juez_id
      );
    } else if (filter === filters.NO_ASIGNADOS) {
      asignacionesFiltradas = asignacionJueces.filter(
        (asignacion) => !asignacion.asignacion_juez_id
      );
    }
    const parsedData = asignacionesFiltradas?.map((asignacion) => {
      const terna = `${asignacion.titular ? `${asignacion.titular} (titular)` : ""}
        ${asignacion.accesitario1 ? `, ${asignacion.accesitario1}` : ""}
        ${asignacion.accesitario2 ? `, ${asignacion.accesitario2}` : ""}
    `;
    });
  });

```

```

const fecha_inicio = asignacion.fecha_inicio
? moment(asignacion.fecha_inicio).format("DD/MM/YYYY")
: "";
const fecha_fin = asignacion.fecha_fin
? moment(asignacion.fecha_fin).format("DD/MM/YYYY")
: "";
return { ...asignacion, terna, fecha_inicio, fecha_fin };
});
setParsedAsignaciones(parsedData);
}, [asignacionJueces, filter]);
const guardarResolucionAdministrativaSeleccionadoHandler = (resolucion) => {
  dispatch(guardarresolucionAdministrativaSeleccionado(resolucion));
};
useEffect(() => {
  if (!terminoBusqueda) setAsignacionJueces([]);
  else buscarAsignacionJuez({ termino: terminoBusqueda, setAsignacionJueces });
}, [terminoBusqueda]);
useEffect(() => {
  guardarResolucionAdministrativaSeleccionadoHandler(selectedResolucion);
}, [selectedResolucion]);
const columnas = useRef([
  {
    Header: "JUZGADO",
    accessor: "nombre_juzgado",
  },
  {
    Header: "NRO RESOLUCION",
    accessor: "numero_resolucion",
  },
  {
    Header: "TERNA",
    accessor: "terna",
  },
  {
    Header: "FECHA INICIO",
    accessor: "fecha_inicio",
  },
  {
    Header: "FECHA FIN",
    accessor: "fecha_fin",
  },
  {
    Header: "VIGENCIA",
    accessor: "vigencia",
  },
]);
const onPorVencer = () => {
  buscarAsignacionJuezPorDiferencia({ vigencia, setAsignacionJueces });
};

```

```

const onSearch = () => {
  if (!terminoBusqueda) setAsignacionJueces([]);
  else buscarAsignacionJuez({ termino: terminoBusqueda, setAsignacionJueces });
};
const onClearSearch = () => {
  setTerminoBusqueda("");
};

const onRadioChange = (e) => {
  setFilter(e.target.value);
};
const onVigenciaChange = (e) => {
  let newValue = e.target.value;
  if (newValue[0] === '0') {
    newValue = newValue.substring(1)
  };
  const isValid = isValidNumber(newValue);
  if (!isValid) setVigencia(0)
  else setVigencia(newValue);
};
return (
  <div>
    <SideBar />
    <BackButton />
    <div className={styles.container}>
      <div className={globals.title_general}>
        ASIGNACIÓN DE JUECES DE PAZ
      </div>
      <div className={globals.form_container}>
        <div className={asignacionStyles.searches_container}>
          <div className={asignacionStyles.search_container}>
            <div>
              <FormLabel>Buscar por juzgado:</FormLabel>
            </div>
            <div className={asignacionStyles.search_sub_container}>
              <input
                type="text"
                className={asignacionStyles.search_input}
                placeholder="Escribe el nombre del juzgado"
                value={terminoBusqueda}
                onChange={onChange}
              />
              {terminoBusqueda && (
                <IconButton
                  size="small"
                  className={asignacionStyles.search_clear_button}
                  onClick={onClearSearch}
                >
                  <Clear />
              )}
            </div>
          </div>
        </div>
      </div>
    </div>
  )

```

```

        </IconButton>
      )}
      <Button
        variant="contained"
        startIcon={<Search />}
        className={asignacionStyles.search_button}
        onClick={onSearch}
      >
        {" "}
        Buscar
      </Button>
    </div>
  </div>
  <div className={asignacionStyles.vigencia_container}>
    <FormLabel>Buscar por días por vencer:</FormLabel>
    <div className={asignacionStyles.vigencia_subcontainer}>
      <div className={asignacionStyles.vigencia_div_container}>
        <input
          className={asignacionStyles.vigencia_text}
          type="number"
          value={vigencia}
          onChange={onVigenciaChange}
        />
      </div>
      <div className={asignacionStyles.button_container}>
        <Link href=" ">
          <Button
            variant="contained"
            className={asignacionStyles.search_button_vencer}
            startIcon={<Search />}
            onClick={onPorVencer}
          >
            Buscar
          </Button>
        </Link>
      </div>
    </div>
  </div>
  </div>
  </div>
  <div>
    <FormControl>
      <FormLabel>Filtrar juzgado por:</FormLabel>
      <RadioGroup row value={filter} onChange={onRadioChange}>
        <FormControlLabel
          value={filters.TODO}
          control={<Radio
/>
}

        label="Todo"

```

```

        />
        <FormControlLabel
        value={filters.ASIGNADOS}
        control={<Radio />}
        label="Asignados"
        />
        <FormControlLabel
        value={filters.NO_ASIGNADOS}
        control={<Radio />}
        label="No asignados"
        />
    </RadioGroup>
    </FormControl>
  </div>
</div>
<div></div>
<div className={globals.buttons_container}>
<div className={globals.button_container}>
<Link href="/asignacion-juez/registro">
  <Button variant="contained" color="success" startIcon={<Add />}>
    {" "}
    Nuevo
  </Button>
</Link>
</div>
</div>
{!parsedAsignaciones?.length && !!terminoBusqueda && <EmptySearchResult
/>}
{!!parsedAsignaciones?.length && (
  <div>
    <div className={globals.title_subgeneral}>
      LISTADO DE ASIGNACIONES DE JUECES DE PAZ
    </div>
    <Table
      data={parsedAsignaciones}
      columns={columnas.current}
      setSelectedItem={setSelectedResolucion}
    />
  </div>
  )
}
</div>
</div>
</div>
);
};
export default AdministrarResolucionAdministrativa;

```

Fuente: Creado por el autor.

Tabla 60*Reporte de Pruebas Unitarias. Primera Iteración.*

N° HU	N° TI	NOMBRE CLASE CONTROL	RESULTADO
2	1	juezpazController(accion=addjuez)	Aprobado
	2	juezpazController(accion=updateJuezPaz)	Aprobado
	3	juezpazController(accion=getJuezpaz)	Aprobado
3	1	juzgadopazController(accion=addJuzgadoPazez)	Aprobado
	2	juzgadopazController(accion=updateJuzgadoPaz)	Aprobado
	3	juzgadopazController(accion=getJuzgadoPazpaz)	Aprobado
5	1	certificadosEmitidosController(accion=addcertificadosEmitidos)	Aprobado
6	1	asignarJuezController(accion=addAsignarJuez)	Aprobado
	2	asignarJuezController(accion=getbuscarAsignarJuez)	Aprobado
7	1	asignarJuezController(accion=getbuscarAsignarPorVigencia)	Aprobado

Fuente: Creado por el autor.

Tabla 61*Reporte de pruebas de integración. Primera iteración.*

N° HU	N° TI	NOMBRE CLASE CONTROL	RESULTADO
2	1	Usuario registrado (administrador) es capaz de registrar juez de paz	Aprobado
	2	Usuario registrado (administrador) es capaz de realizar la actualización de datos de los juez de paz	Aprobado
	3	Usuario registrado (administrador) es capaz de realizar la búsqueda de jueces de paz	Aprobado
3	1	Usuario registrado (administrador) es capaz de realizar el registro del juzgado de paz	Aprobado
	2	Usuario registrado (administrador) es capaz de realizar la actualización de datos del juzgado de paz	Aprobado
	3	Usuario registrado (administrador) es capaz de realizar la búsqueda de juzgados de paz	Aprobado
5	1	Usuario registrado (juez de paz) es capaz de emitir certificado escritura pública de transferencia de bienes inmuebles no registrables. Generar PDF y descargar	Aprobado
6	1	Usuario registrado (administrador) es capaz de realizar el registro del juzgado de paz y la asignación de su respectiva terna.	Aprobado
	2	Usuario registrado (administrador) es capaz de realizar la búsqueda de juzgados de paz con terna	Aprobado
7	1	Usuario registrado (administrador) es capaz de realizar la búsqueda de juzgados con ternas próximos a vencer	Aprobado

Fuente: Creado por el autor.

PRUEBAS DE ACEPTACIÓN

Tabla 62

Reporte de pruebas de aceptación. Administrar Usuarios del Sistema.

Nº CASO PRUEBA	1
Propósito	acceder al sistema de justicia de paz
ACTIVIDAD	
Inicialización	Ingresar al sistema web de justicia de paz, e ingresar al menú de inicio sesión.
Descripción de datos de entrada	usuario, password
RESULTADOS	
Esperado	sr. : Max aqui bautista, usuario: administrador
Reales	Acceder al sistema web justicia de paz.

Fuente: Creado por el autor.

Tabla 63

Reporte de Pruebas de Aceptación: Administrar Juez de Paz.

Nº CASO PRUEBA	2
Propósito	tener el registro de jueces de paz
ACTIVIDAD	
Inicialización	registrar los datos requeridos del juez de paz
Descripción de datos de entrada	nombre del juez de paz, apellidos, DNI
RESULTADOS	
Esperado	Sr. : Max aqui bautista, usuario :administrador
Reales	Contar con un registro de jueces de paz de todo el distrito judicial de Ayacucho.

Fuente: Creado por el autor.

Tabla 64

Reporte de pruebas de aceptación: Administrar Juzgado de Paz.

Nº CASO PRUEBA	3
Propósito	tener el registro de juzgados de paz

ACTIVIDAD	
Inicialización	registrar los datos requeridos del juzgado de paz
Descripción de datos de entrada	Nombre del juzgado de paz, departamento, provincia, distrito, dirección, categoría, teléfono, observaciones.
RESULTADOS	
Esperado	sr. : Max aqui bautista, usuario :administrador
Reales	Tener registro de todos los juzgados de paz del distrito judicial de Ayacucho.

Fuente: Creado por el autor.

Tabla 65

Reporte de pruebas de aceptación. Administrar Recursos Materiales y Tecnológicos.

Nº CASO PRUEBA	4
Propósito	Tener el registro de recursos materiales y tecnológicos con los cuales cuenta cada juzgado de paz.
ACTIVIDAD	
Inicialización	registrar todo los recursos materiales y tecnológicos de los juzgados de paz
Descripción de datos de entrada	Nombre del juzgado de paz, nombre, tipo, serie, código patrimonial, color, descripción, observación.
RESULTADOS	
Esperado	sr. : Max aqui bautista, usuario :administrador
Reales	Tener listado de los recursos materiales y tecnológicos con el cual cuenta el juzgado.

Fuente: Creado por el autor.

Tabla 66

Reporte de pruebas de aceptación. Emitir constancias de transferencia.

Nº CASO PRUEBA	5
Propósito	Emitir certificado escritura pública de transferencia de bienes inmuebles no registrables.
ACTIVIDAD	
Inicialización	Seleccionar el tipo de certificado a generar e ingresar los datos necesarios.
Descripción de datos de	Nombres y apellidos, DNI, estado civil, ocupación,

entrada	domicilio, datos del receptor, provincia, limites, contexto de la adquisición de fecha de adquisición modo de pago y folios.
RESULTADOS	
Esperado	sr. : Max aqui bautista, usuario : administrador
Reales	Tener la constancia de emitir certificado escritura pública de transferencia de bienes inmuebles no registrables.

Fuente: Creado por el autor.

Tabla 67

Reporte de pruebas de aceptación. Asignar Terna al Juzgado.

Nº CASO PRUEBA	6
Propósito	Asignar jueces de paz a los respectivos juzgados de paz.
ACTIVIDAD	
Inicialización	Asignar terna al juzgado.
Descripción de datos de entrada	Nro. De resolución, fecha de inicio. fecha fin, fecha de resolución PDF, nombre de juzgado, juez titular, accesitario 1 y accesitario 2
RESULTADOS	
Esperado	Sr. : Max aqui bautista, usuario : administrador
Reales	Tener juzgados de paz con sus respectivos jueces de paz.

Fuente: Creado por el autor.

Tabla 68

Reporte de pruebas de aceptación. Consultar Juzgados con Ternas Próximos a Vencer.

Nº CASO PRUEBA	7
Propósito	Tener el listado de juzgados de paz pero con listado de jueces de paz que están próximos a vencer el periodo asignado.
ACTIVIDAD	
Inicialización	Consultar juzgados con ternas próximas a vencer.
Descripción de datos de	Ingresar el número de días que faltan para que finalice

entrada	el periodo designado a la terna de jueces de paz.
RESULTADOS	
Esperado	sr. : Max auqui bautista, usuario: administrador
Reales	Contar con el listado de juzgados que tienen jueces de paz asignados y están próximos a vencer.

Fuente: Creado por el autor.

Tabla 69

Reporte de pruebas de aceptación: Administrar Cuadro de Aranceles.

Nº CASO PRUEBA	10
Propósito	Registrar los aranceles judiciales
ACTIVIDAD	
Inicialización	administrar cuadro de aranceles
Descripción de datos de entrada	Concepto, costo del arancel.
RESULTADOS	
Esperado	sr. : Max auqui bautista, usuario: administrador
Reales	Contar con los conceptos y precios de cada arancel.

Fuente: Creado por el autor.

4.2. DISCUSIÓN

- a. Flores, A., & Roque, M., en su trabajo de investigación, concluyen que la implementación del sistema de registro único y gestión documentaria acelera el tiempo de búsqueda y emisión de documentos en los juzgados de paz del distrito judicial de Puno. En la presente investigación, la funcionalidad desarrollada, que permite a los jueces de paz y al responsable de ODAJUP encontrar más rápidamente los documentos que necesitan para realizar sus funciones, mejora la eficiencia y la eficacia del trabajo de ambos.
- b. Purificación J. & Valverde J. concluyen en su tesis que la implementación del sistema de información con tecnología web moderniza la verificación de la formalidad en la verificación de autenticidad de los documentos expedidos por jueces de paz en la zona registral N° VII Sede – Huaraz. Evitando así el uso de oficios en papel y la pérdida de los mismos, ello hacía más dificultoso la precalificación de títulos, pero actualmente eso ha mejorado significativamente. En la presente investigación se tiene los formularios en un sistema web, los cuales son almacenados en la nube y están disponibles para cuando lo requieran.

CAPÍTULO V

CONCLUSIONES

5.1. CONCLUSIONES

- a. Se ha logrado desarrollar la aplicación web para la gestión de recursos humanos como se observa en las tablas 15, 16, 20, 21, 25, 26, 29 y las figuras 13, 14, 19, 20, 21, 22.
- b. Se ha logrado desarrollar la aplicación web para la gestión de recursos financieros como se observa en las tablas 19, 24, 30 y las figuras 15, 16, 23, 24, 25, 26, 27.
- c. Se ha logrado desarrollar la aplicación web para la gestión de recursos materiales como se observa en las tablas 18, 26 y las figuras 17,18.
- d. Se ha logrado desarrollar la aplicación web la gestión de recursos tecnológicos se observa en las tablas 18, 26 y las figuras 30, 31.

CAPÍTULO VI RECOMENDACIONES

6.1. RECOMENDACIONES

- a. Investigar el desarrollo de una aplicación móvil para usuarios que serian los jueces de paz, para facilitar la consulta de los certificados emitidos en su juzgado de paz.
- b. Se debe estudiar el desarrollo del módulo de exhortos judiciales para los jueces de paz, para que facilite la notificación de resoluciones judiciales.
- c. Se debe estudiar el desarrollo del módulo de administración de capacitaciones para los jueces de paz, para que facilite la recepción e invitaciones a capacitaciones y acceso a los recursos brindados en las capacitaciones.

REFERENCIAS BIBLIOGRÁFICAS

- Apolo, N., Delgado, K. y Quesada, C. (2018). Procesos y Fundamentos de la Investigación Científica. UTMACH.
- Asimbaya A. (2019). Automatización de pruebas unitarias en el marco de desarrollo ágil para cartera de tickets electrónicos de parqueadero. [Trabajo de grado, Pontificia Universidad Católica del Ecuador]. Archivo digital.
- Barquero, A. (2005). La Administración de Recurso Humano (I parte)/ EUNED.
- Bernal, C. (2010). Metodología de investigación. (3ra Ed.). Pearson Educación.
- Beck, K. (2000). Extreme programming explained: Embrace change (2.a ed.). Addison-Wesley.
- Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Fowler, M., Highsmith, J., ... & Thomas, D. (2001). Extreme programming explained: Embrace change (2.a ed.). Addison-Wesley.
- Buedo, D., Marin A. y Trujillo Y. (2020). Estrategia de pruebas para organizaciones desarrolladoras de software. Revista Cubana de Ciencias Informáticas. 14(3), 83-104. http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S2227-18992020000300083&lng=es&tlng=es.
- Castro, E. (2016). Teoría y práctica de la investigación científica. Perugraph S.R.L.
- Congreso de la República del Perú. (1985). Ley Orgánica del Poder Judicial (Texto Único Ordenado). Lima: Imprenta del Congreso de la República.
- Congreso de la República del Perú. (2013). Ley Orgánica del Poder Judicial (Texto Único Ordenado). Lima: Imprenta del Congreso de la República.
- Chinarro, E. (2019). Definición e implementación del proceso de pruebas de software basado en la NTP-ISO/IEC 12207:2016 aplicado a una empresa consultora de software. [Tesis pregrado, Universidad Nacional Mayor de San Marcos]. Archivo digital. <https://cybertesis.unmsm.edu.pe/handle/20.500.12672/17023>.
- Díaz, E. y Heler, M. (2000). El conocimiento científico. Tomo I: Cap. 1. Tomo II: Cap. 5, Punto 2. EUDEBA, Buenos Aires.
- Dankhe, G. (1986). Métodos de investigación del marketing. México: McGraw-Hill.
- Garlan, D. y Shaw, M. (1997, enero). An Introduction to Software Architecture [Una Introducción a la Arquitectura de Software]. Advances in Software Engineering and Knowledge Engineering, (1, 39).https://userweb.cs.txstate.edu/~rp31/papers/intro_softarch.pdf.
- Huizinga, D. y Kolawa, A. (2007). Automated Defect Prevention. Best Practices in Software Management [Prevención de defectos automatizada. Mejores prácticas en la gestión de software]. John Wiley & Sons, Inc.
- Hernández, R., Fernández, C., & Baptista, P. (2020). Metodología de la investigación (6a. ed.). México: McGraw-Hill.
- Freeman, E., & Robson, E. (2014). Head First JavaScript Programming. Sebastopol: O'Reilly Media.
- L. Bass, L., Clements, P., Kazman, R., (2003). Software Architecture in Practice [Arquitectura de Software en Práctica], (2.a Ed.), Addison Wesley.
- Marchesi, M., Succi, G., Wells, D., & Williams, L. (2002). Agile software development: Principles, patterns, and practices. Addison-Wesley.
- Oficina Nacional de Justicia de Paz y Justicia Indígena/Fondo Editorial del Poder Judicial. (2020). La función notarial del juez de paz (1.a ed.). Lima: Oficina Nacional de Justicia de Paz y Justicia Indígena/Fondo Editorial Del Poder Judicial.
- Palomino, J. (2020). Automatización plan de pruebas funcional: Plataforma web de la Universidad Politécnica de Madrid [Proyecto fin de carrera, Escuela Técnica

- Superior de Ingenieros Informáticos de la Universidad Politécnica de Madrid]. Archivo digital. <https://oa.upm.es/63115/>
- Pardo, H. (2005). Un modelo de aplicación web institucional universitaria. El caso de los webcom: sitios de facultades de comunicación de Iberoamérica [Tesis doctoral, Universidad Autónoma de Barcelona]. Archivo digital. <https://www.tesisenred.net/bitstream/handle/10803/4122/hpk1de2.pdf>.
- Pérez, M. (2021, julio). Definición de método. Concepto Definición. <https://conceptodefinicion.de/metodo/>.
- Poder Judicial, Gerencia General, Gerencia de Planificación, Subgerencia de Estadística (2022), Juzgados de Paz (nro 10), Gobierno del Perú).
- Parker, R. (2020). Extreme programming explained: Embrace change (4th ed.). Addison-Wesley Professional.
- Rodríguez, I. (2018). Manual de metodología de investigación aplicada a las ciencias sociales y jurídicas: Perspectiva de género. Edición Kindle.
- Sander, R. (2017). Continuous Integration, Delivery, and Deployment [Integración, entrega y despliegue continuo]. Packt Publishing.
- Serna, E. y Arango, F. (2011). Prueba del software: más que una fase en el ciclo de vida. Revista de Ingeniería, 1(35), 34-40.
- Silberschatz, A., Korth, H. F., & Sudarshan, S. (2012). Database system concepts (6th ed.). McGraw-Hill Education.
- Sommerville, I. (2020). Engineering Software Products. An Introduction to Modern Software Engineering [Productos de ingeniería de software. Introducción a la ingeniería de software moderna]. Pearson.
- Suehring, S. (2013). Javascript Step by Step Third Edition. Sebastopol: O'Reilly.

Lista de Abreviaturas

ODAJUP: Oficina Distrital de Justicia de Paz
ONAJUP: Oficina Nacional de Justicia de Paz
XP: Programación Extrema
API: Interfaz de programación de aplicaciones
HU: Historia de usuario
GUI: Interfaz gráfica de usuario

Glosario

Historia de usuario o user story

Es una ficha que contiene toda la información relevante disponible sobre una funcionalidad del product backlog. Es la pieza fundamental del product backlog y del sprint backlog.

Anexos

Anexo 1. Matriz de consistencia

Título: Aplicación Web para Gestión de Recursos en los Juzgados de Paz del Distrito Judicial de Ayacucho, 2023

PROBLEMAS	OBJETIVOS	VARIABLES	METODOLOGÍA
PROBLEMA GENERAL ¿Cómo realizar la gestión de recursos en los Juzgados de Paz del Distrito Judicial de Ayacucho, 2023? PROBLEMAS ESPECÍFICOS ¿Cómo realizar la gestión del recurso humano, financiero, material y tecnológico?	OBJETIVO GENERAL Desarrollar una aplicación web para realizar la gestión de recursos en los Juzgados de Paz del Distrito Judicial de Ayacucho, 2023 OBJETIVOS ESPECÍFICOS Explorar, planificar e iterar la gestión del recurso humano Explorar, planificar e iterar la gestión del recurso financiero Explorar, planificar e iterar la gestión material Explorar, planificar e iterar la gestión del recurso tecnológico	VARIABLE DE INTERÉS X: Gestión de recursos VARIABLES DESCRIPTIVAS X1: Recurso humano X2: Recurso financiero X3: Recurso material X4: Recurso tecnológico	TIPO DE INVESTIGACIÓN Observacional, prospectivo, transversal y descriptivo NIVEL DE INVESTIGACIÓN Descriptivo DISEÑO No experimental, prospectivo-transversal POBLACIÓN La población estará compuesta por cinco procesos de recursos humanos, cinco procesos de recursos financieros, tres procesos de recursos materiales, tres procesos de recurso tecnológicos, de los Juzgados de Paz del Distrito Judicial de Ayacucho el año 2023. MUESTRA No existe muestra, será un censo, porque se estudiará; cinco procesos

			de recursos humanos, cinco procesos de recursos financieros, tres procesos de recursos materiales, tres procesos de recurso tecnológicos, de los Juzgados de Paz del Distrito Judicial de Ayacucho el año 2023. TÉCNICA Entrevista INSTRUMENTO Guía de entrevista
--	--	--	--

Anexo 2. Guía de Entrevista

GUIA DE ENTREVISTA PARA EL RESPONSABLE DEL ODAJUP

Esta entrevista tiene como objetivo entender la importancia de la organización y disponibilidad de información en la Oficina de Apoyo a los Juzgados de Paz (ODAJUP), cómo influye en la eficiencia y funcionamiento de dichos juzgados.

Pregunta 1: ¿Qué documento formaliza la designación de jueces de paz?

Pregunta 2: ¿Cómo identifica usted que un juzgado de paz tiene terna de jueces asignadas?

Pregunta 3: ¿Cómo identifica usted que un juzgado de paz no tiene terna de jueces asignadas?

Pregunta 4: ¿Cómo identifica usted que juzgados de paz tienen terna de jueces asignadas próximas a vencer?

Pregunta 5: ¿Con cuánto tiempo de anticipación se debe realizar la convocatoria de una nueva terna jueces de paz?

Pregunta 6: ¿Cómo sabe usted con qué mobiliarios cuenta el juzgado de paz?

Pregunta 7: ¿Qué tipo de mobiliario son útiles para el buen funcionamiento del juzgado de paz?

Pregunta 8: ¿Qué características es necesario saber para identificar el bien patrimonial del juzgado de paz?

Pregunta 9: ¿Cómo sabe usted con qué equipo de procesamiento de datos cuenta el juzgado de paz?

Pregunta 10: ¿Qué tipo de equipo de procesamiento de datos son útiles para el buen funcionamiento del juzgado de paz?

Pregunta 11: ¿Qué características es necesario saber para identificar el equipo de procesamiento de datos del juzgado de paz?

GUIA DE ENTREVISTA PARA JUECES DE PAZ

El objetivo de esta entrevista es recopilar información valiosa de los jueces de paz para diseñar un software eficiente y funcional que facilite la emisión de certificados. Su experiencia y conocimientos son fundamentales para desarrollar una herramienta útil y precisa.

Pregunta 1: ¿Cómo obtiene el juez de paz los aranceles judiciales?

Pregunta 2: ¿Cómo será el formato predefinido para la emisión de certificados?

Pregunta 3: ¿Cómo verifica qué certificados se emitieron en el juzgado de paz?

Pregunta 4: ¿Qué elementos visuales ayudan al juez de paz a prevenir errores en la emisión de certificados?

Pregunta 5: ¿Cuáles son los formatos más usados para la emisión de certificados?

Pregunta 6: ¿Cómo obtiene los aranceles judiciales para la emisión de certificados?

Anexo 3. Certificado de validez de contenido del instrumento que mide método para la Gestión de Recursos en el Juzgado de Paz

Señor juez o experto se requiere su colaboración para determinar la validez de contenido del instrumento registro, también usted puede sugerir retirar ítems, incorporar ítems o cambiar el contenido de un ítem. Se agradece su colaboración.

¹Relevancia: El ítem corresponde al concepto teórico formulado.

²Representatividad: El ítem es apropiado para representar al componente o dimensión específica del contenido

³Claridad: se entiende sin dificultad alguna el enunciado del ítem, es conciso, exacto y directo

Nota: Suficiencia, se dice suficiencia cuando los ítems planteados son suficientes para medir la dimensión

Nº	Dimensiones / ítems	Relevancia ¹		Representatividad ²		Claridad ³		Sugerencias/Observaciones
		Sí	No	Sí	No	Sí	No	
RESPONSABLE DEL ODAJUP								
1	¿Qué documento formaliza la designación de jueces de paz?							
2	¿Cómo identifica usted que un juzgado de paz tiene terna de jueces asignadas?							
3	¿Cómo identifica usted que un juzgado de paz no tiene terna de jueces asignadas?							
4	¿Cómo identifica usted que juzgados de paz tienen terna de jueces asignadas próximas a vencer?							
5	¿Con cuánto tiempo de anticipación se debe realizar la convocatoria de la nueva terna?							
6	¿Cómo sabe usted que un juzgado de paz tiene mobiliarios asignados?							

7	¿Cómo sabe usted que un juzgado de paz tiene bienes tecnológicos asignados?							
8	¿Cómo ODAJUP garantiza que los jueces de paz tengan la información más reciente sobre los aranceles judiciales?							
JUECES DE PAZ								
9	¿Sería beneficioso para su trabajo que el software tuviera formatos predefinidos para la emisión de certificados?							
10	¿Cómo puede usted verificar qué certificados se emitieron en su juzgado?							
11	¿Cuánto tiempo se demora en realizar la redacción de un certificado?							
12	¿Qué elementos visuales podrían ayudar al juez de paz a prevenir errores en la emisión de certificados?							
13	¿Sería beneficioso para usted tener el listado formatos predeterminados para la emisión de certificados?							
14	¿Cómo puede usted acceder a información actualizada sobre los aranceles judiciales para la emisión de certificados?							

Observaciones (precisar si hay observaciones):

Opinión de aplicabilidad: Aplicable [] Aplicable después de corregir [] No aplicable []

Apellidos y nombres del juez validador Dr. / Mg / Ing:

DNI O CE:

Especialidad del validador:

.....

Firma

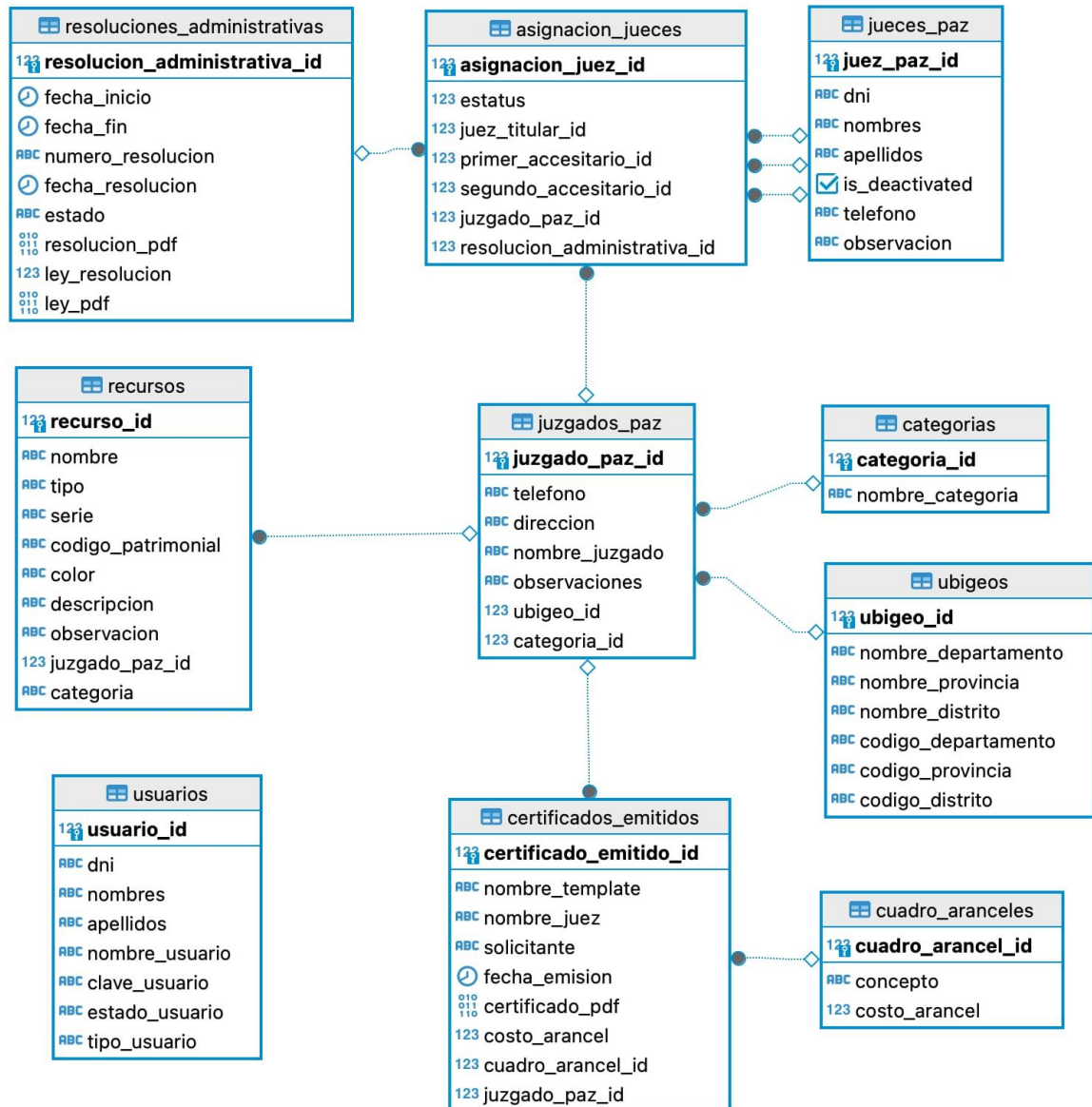
¹ **Relevancia:** El ítem corresponde al concepto teórico formulado.

² **Representatividad:** El ítem es apropiado para representar al componente o dimensión específica del constructo

³ **Claridad:** Se entiende sin dificultad alguna el enunciado del ítem, es conciso, exacto y directo

Nota: Suficiencia, se dice suficiencia cuando los ítems planteados son suficientes para medir la dimensión.

ANEXO 4 Modelo de Base de Datos Física



**UNSCH****FACULTAD DE
INGENIERÍA
DE MINAS, GEOLOGÍA Y CIVIL****ACTA No 034-2024-FIMGC: ACTA DE SUSTENTACION DE TESIS****PARA OPTAR EL TÍTULO PROFESIONAL DE INGENIERO DE SISTEMAS**

En la Universidad Nacional de San Cristóbal de Huamanga de la ciudad de Ayacucho, en cumplimiento a la **Resolución Decanal N° 420-2024-FIMGC-D**, a los veinticuatro días del mes de junio de 2024, siendo las 10:00 a.m., reunidos en el Auditorio de la Escuela Profesional de Ingeniería de Minas, bajo la presidencia del MSc. Ing. José Ernesto ESTRADA CÁRDENAS y los miembros; Mg. Ing^a. Edith Felicitas GUEVARA MOROTE, Mg. Ing. Juan Carlos CARREÑO GAMARRA, actuando como secretario docente el MSc. Kelvis BERROCAL ARGUMEDO, para proceder a la sustentación de tesis para optar el Título Profesional de Ingeniero de Sistemas, del bachiller:

MAX AUQUI BAUTISTA

Quien presentó la tesis denominada:

**SISTEMA WEB PARA LA GESTIÓN DE RECURSOS EN LOS JUZGADOS DE PAZ DEL
DISTRITO JUDICIAL DE AYACUCHO, 2023**

Los señores miembros del jurado luego de expuesto la tesis y absueltas las preguntas, delibera y lo declaran:

APROBADO CON NOTA QUINCE (15)

Siendo las 12:15 p.m. del día 24 de junio de 2024, culmina el acto de sustentación de tesis, y en conformidad a lo actuado los miembros del jurado firmamos al pie del presente.

MSc. Ing. José Ernesto ESTRADA CÁRDENAS
Presidente

Mg. Ing^a. Edith Felicitas GUEVARA MOROTE
Miembro

Mg. Ing. Juan Carlos CARREÑO GAMARRA
Miembro

MSc. Ing. Kelvis BERROCAL ARGUMEDO
Miembro

cc:

Archivo



CONSTANCIA DE ORIGINALIDAD DE TRABAJO DE INVESTIGACIÓN

CONSTANCIA N° 009-2024-KPS-FIMGC/UNSCH

El que suscribe; responsable verificador de originalidad de trabajos de tesis de pregrado con el software Turnitin, en segunda instancia para las **Escuelas Profesionales** de la **Facultad de Ingeniería de Minas, Geología y Civil**; en cumplimiento a la **Resolución de Consejo Universitario N° 039-2021-UNSCH-CU**, Reglamento de Originalidad de Trabajos de Investigación de la Universidad Nacional San Cristóbal de Huamanga y **Resolución Decanal N° 473-2023-FIMGC-D**, deja constancia de originalidad de trabajo de investigación, que el/la Sr./Srta.

Nombres y Apellidos : Max Auqui Bautista

Escuela Profesional : INGENIERÍA DE SISTEMAS

Título de la Tesis : SISTEMA WEB PARA LA GESTIÓN DE RECURSOS EN LOS JUZGADOS DE PAZ DEL DISTRITO JUDICIAL DE AYACUCHO, 2023

Evaluación de la Originalidad : 22% Índice de Similitud

Identificador de la entrega : 2474895374

Por tanto, según los Artículos 12, 13 y 17 del Reglamento de Originalidad de Trabajos de Investigación, es **PROCEDENTE** otorgar la **Constancia de Originalidad** para los fines que crea conveniente.

En señal de conformidad y verificación se firma la presente constancia

Ayacucho, 09 de octubre de 2024



Firmado digitalmente por:
PERALTA SOTOMAYOR Karel
FAU 20143660754 soft
Motivo: Soy el autor del documento
Fecha: 09/10/2024 22:08:45-0500

SISTEMA WEB PARA LA GESTIÓN DE RECURSOS EN LOS JUZGADOS DE PAZ DEL DISTRITO JUDICIAL DE AYACUCHO, 2023

por Max Auqui Bautista

Fecha de entrega: 04-oct-2024 09:22a.m. (UTC-0500)

Identificador de la entrega: 2474895374

Nombre del archivo: Max_Auqui_Bautista.pdf (5.29M)

Total de palabras: 30637

Total de caracteres: 180973

SISTEMA WEB PARA LA GESTIÓN DE RECURSOS EN LOS JUZGADOS DE PAZ DEL DISTRITO JUDICIAL DE AYACUCHO, 2023

INFORME DE ORIGINALIDAD

22%

INDICE DE SIMILITUD

23%

FUENTES DE INTERNET

3%

PUBLICACIONES

13%

TRABAJOS DEL ESTUDIANTE

FUENTES PRIMARIAS

1

repositorio.unsch.edu.pe

Fuente de Internet

6%

2

Submitted to Universidad Nacional de San Cristóbal de Huamanga

Trabajo del estudiante

6%

3

hdl.handle.net

Fuente de Internet

2%

4

www.pj.gob.pe

Fuente de Internet

1%

5

ela.kpi.ua

Fuente de Internet

1%

6

repositorio.unap.edu.pe

Fuente de Internet

1%

7

repositorio.upch.edu.pe

Fuente de Internet

<1%

8

repositorio.unasam.edu.pe

Fuente de Internet

<1%

9	repositorio.unheval.edu.pe Fuente de Internet	<1 %
10	baixardoc.com Fuente de Internet	<1 %
11	1library.co Fuente de Internet	<1 %
12	dspace.utpl.edu.ec Fuente de Internet	<1 %
13	Submitted to Pontificia Universidad Catolica del Peru Trabajo del estudiante	<1 %
14	IDOM CONSULTING, ENGINEERING, ARCHITECTURE, S.A. SUCURSAL DEL PERU. "Plan de Recuperación de Área Degradada por Residuos Sólidos Municipales, como Instrumento de Gestión Ambiental Complementario del Proyecto de Recuperación del Área Degradada por Residuos Sólidos El Relleno, Sector El Delfín, Distrito de Pozuzo, Provincia de Oxapampa, Departamento de Pasco-IGA0015385", R.G. N° 008-2021-GRB-M.P.O, 2022 Publicación	<1 %
15	repositorio.uladech.edu.pe Fuente de Internet	<1 %
16	es.slideshare.net Fuente de Internet	

<1 %

17

repositorio.uncp.edu.pe

Fuente de Internet

<1 %

18

Submitted to Instituto Tecnológico de Costa Rica

Trabajo del estudiante

<1 %

19

repositorio.uss.edu.pe

Fuente de Internet

<1 %

20

repositorio.utelesup.edu.pe

Fuente de Internet

<1 %

21

Submitted to Universidad Cesar Vallejo

Trabajo del estudiante

<1 %

22

ria.utn.edu.ar

Fuente de Internet

<1 %

23

Submitted to Universidad Internacional de la Rioja

Trabajo del estudiante

<1 %

24

repositorio.uancv.edu.pe

Fuente de Internet

<1 %

25

idoc.pub

Fuente de Internet

<1 %

26

www.buenastareas.com

Fuente de Internet

<1 %

27	Submitted to Universitas Gadjah Mada	<1 %
	Trabajo del estudiante	
28	repositorio.usanpedro.edu.pe	<1 %
	Fuente de Internet	
29	essuir.sumdu.edu.ua	<1 %
	Fuente de Internet	
30	www.diputados.gob.mx	<1 %
	Fuente de Internet	
31	Submitted to Corporación Universitaria Minuto de Dios, UNIMINUTO	<1 %
	Trabajo del estudiante	
32	qdoc.tips	<1 %
	Fuente de Internet	
33	dspace.ups.edu.ec	<1 %
	Fuente de Internet	
34	repositorio.unan.edu.ni	<1 %
	Fuente de Internet	

Excluir citas

Activo

Excluir coincidencias < 30 words

Excluir bibliografía

Activo