



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

CARRERA SOFTWARE

SISTEMA MÓVIL CON SEGURIDAD BIOMÉTRICA PARA GESTIONAR TICKETS DE INCIDENCIAS TI EN LAS SUCURSALES DE RIOBAMBA DE LA COOPERATIVA DE AHORRO Y CRÉDITO FERNANDO DAQUILEMA

Trabajo de Integración Curricular

Tipo: Proyecto Técnico

Presentado para optar el grado académico de:

INGENIERO DE SOFTWARE

AUTOR: DENNYS MAURICIO MEJIA BRONCANO

DIRECTOR: ING. RAÚL HERNÁN ROSERO MIRANDA

Riobamba – Ecuador

2024

© 2024, **Dennys Mauricio Mejia Broncano**

Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo cita bibliográfica del documento, siempre y cuando se reconozca el Derecho del Autor.

Yo, Dennys Mauricio Mejia Broncano, declaro que el presente Trabajo de Integración Curricular es de mi autoría y los resultados de este son auténticos. Los textos en el documento que provienen de otras fuentes están debidamente citados y referenciados.

Como autor asumo la responsabilidad legal y académica de los contenidos de este Trabajo de Integración Curricular; el patrimonio intelectual pertenece a la Escuela Superior Politécnica de Chimborazo.

Riobamba, 02 de mayo del 2024



Dennys Mauricio Mejia Broncano




060583002-5

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

CARRERA SOFTWARE

El Tribunal del Trabajo de Integración Curricular certifica que: El Trabajo de Integración Curricular; Tipo Proyecto Técnico, **SISTEMA MÓVIL CON SEGURIDAD BIOMÉTRICA PARA GESTIONAR TICKETS DE INCIDENCIAS TI EN LAS SUCURSALES DE RIOBAMBA DE LA COOPERATIVA DE AHORRO Y CRÉDITO FERNANDO DAQUILEMA**, ha sido minuciosamente revisado por los Miembros del Trabajo de Integración Curricular, el mismo que cumple con los requisitos científicos, técnicos, legales, en tal virtud el Tribunal Autoriza su presentación.

	FIRMA	FECHA
Dr. Danilo Mauricio Pastor Ramirez PRESIDENTE DEL TRIBUNAL		2024/05/02
Ing. Raúl Hernán Rosero Miranda, Ph.D DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR		2024/05/02
Ing. María Belén Paredes Regalado, MSc ASESORA DEL TRABAJO DE INTEGRACIÓN CURRICULAR		2024/05/02

DEDICATORIA

A mis padres, cuya fuente inagotable de amor y apoyo ha sido mi mayor inspiración en este arduo camino académico. A mis hermanos, por ser mis compañeros de vida y cómplices en cada desafío. Su constante aliento y risas compartidas han iluminado incluso los días más difíciles. Este logro es tanto de ustedes como mío. Gracias por ser los pilares inquebrantables de mi éxito y por compartir conmigo esta travesía llena de aprendizajes y logros.

Dennys.

AGRADECIMIENTO

Quiero expresar mi profundo agradecimiento a mis padres y hermanos, quienes han sido el cimiento de mi éxito académico. Su amor incondicional y apoyo constante han sido mi mayor fortaleza. A mis amigos, por compartir risas y aliento en cada paso de este camino. Este logro es también suyo. Gracias por estar siempre a mi lado, inspirándome y motivándome a alcanzar mis metas. Este éxito es tan suyo como mío. ¡Gracias por ser mi fuente de amor, aliento y constante inspiración!

Dennys.

ÍNDICE DE CONTENIDO

INDICE DE TABLAS.....	ix
INDICE DE ILUSTRACIONES.....	x
INDICE DE ANEXOS.....	xi
RESUMEN.....	xii
SUMMARY.....	¡Error! Marcador no definido.
INTRODUCCIÓN.....	1

CAPÍTULO I

1	PLANTEAMIENTO DEL PROBLEMA.....	2
1.1	Antecedentes.....	2
<i>1.1.1</i>	<i>Formulación del Problema.....</i>	<i>3</i>
<i>1.1.2</i>	<i>Sistematización del Problema.....</i>	<i>3</i>
1.2	Justificación.....	4
<i>1.2.1</i>	<i>Justificación teórica.....</i>	<i>4</i>
<i>1.2.2</i>	<i>Justificación aplicativa.....</i>	<i>4</i>
1.3	Objetivos.....	6
<i>1.3.1</i>	<i>Objetivo general.....</i>	<i>6</i>
<i>1.3.2</i>	<i>Objetivos específicos.....</i>	<i>6</i>

CAPÍTULO II

2	MARCO TÉORICO.....	7
2.1	Soporte Técnico.....	7
2.2	Eficiencia.....	7
2.3	Lenguaje de programación Dart.....	7
2.4	Flutter.....	8
2.5	Redmine.....	9
2.6	Scrum.....	10
2.7	Sistemas para la gestión de incidencias existentes.....	10
<i>2.7.1</i>	<i>IntegriaIMS (Integria Incident Manager Sistem).....</i>	<i>10</i>
<i>2.7.2</i>	<i>Open Technology Real Services (OTRS).....</i>	<i>11</i>
<i>2.7.3</i>	<i>Manage Engine Service Desk Plus.....</i>	<i>11</i>
2.8	Trabajos relacionados.....	11

CAPÍTULO III

3	MARCO METODOLÓGICO.....	13
3.1	Tipo de estudio.....	13
<i>3.1.1</i>	<i>Investigación Aplicada.....</i>	<i>13</i>
<i>3.1.2</i>	<i>Investigación cuantitativa.....</i>	<i>14</i>
<i>3.1.3</i>	<i>Investigación cuasiexperimental.....</i>	<i>14</i>
3.2	Métodos y Técnicas.....	14
<i>3.2.1</i>	<i>Unidad de análisis.....</i>	<i>15</i>

3.2.2	<i>Instrumentos</i>	15
3.2.3	<i>Población y muestra</i>	15
3.2.4	<i>Operacionalización conceptual y metodológica de las variables</i>	15
3.3	Ambientes de prueba	16
3.3.1	<i>Análisis Estadístico</i>	16
3.4	Proceso de gestión de incidencias de la cooperativa	17
3.4.1	<i>Proceso actual de gestión de incidencias de la cooperativa</i>	17
3.4.2	<i>Proceso actual de gestión de incidencias con el sistema implementado</i>	18
3.5	Estudio de factibilidad	18
3.5.1	<i>Factibilidad técnica</i>	18
3.5.2	<i>Factibilidad operativa</i>	20
3.5.3	<i>Factibilidad económica</i>	20
3.6	Desarrollo de la aplicación mediante Scrum	21
3.6.1	<i>Fase de planificación</i>	21
3.6.2	<i>Fase de diseño</i>	30
3.6.3	<i>Fase de desarrollo</i>	36

CAPÍTULO IV

4	ANÁLISIS E INTERPRETACIÓN DE RESULTADOS	39
4.1	Formulación de hipótesis	39
4.1.1	<i>Formulación de hipótesis nula</i>	39
4.1.2	<i>Formulación de hipótesis alterna</i>	39
4.2	Medición de la eficiencia	39
4.2.1	<i>Encuesta</i>	39
4.2.2	<i>Análisis de tendencias</i>	40
4.2.3	<i>Prueba de normalidad de Shapiro Wilk</i>	43
4.2.4	<i>Prueba de Wilcoxon</i>	46

CAPÍTULO V

5	CONCLUSIONES Y RECOMENDACIONES	52
5.1	Conclusiones	52
5.2	Recomendaciones	52

GLOSARIO

BIBLIOGRAFÍA

ANEXO

INDICE DE TABLAS

Tabla 3-1: Objetivo, métodos, técnicas y fuentes a usar en base a los objetivos específicos especificados.....	14
Tabla 3-2: Sub-características de eficiencia de desempeño con cada métrica respectivamente	16
Tabla 3-3: Recursos de hardware requeridos.....	19
Tabla 3-4: Recursos de software requeridos.....	19
Tabla 3-5: Recursos humanos requeridos.....	19
Tabla 3-6: Costos.....	20
Tabla 3-7: Miembros y roles.....	22
Tabla 3-8: Usuario-cliente.....	22
Tabla 3-9: Usuario-técnico.....	23
Tabla 3-10: Requisito funcional 1.....	24
Tabla 3-11: Requisito funcional 2.....	25
Tabla 3-12: Requisito funcional 3.....	25
Tabla 3-13: Requisito funcional 4.....	26
Tabla 3-14: Requisito funcional 5.....	26
Tabla 3-15: Extensión 1.....	27
Tabla 3-16: Extensión 2.....	27
Tabla 3-17: Extensión 3.....	27
Tabla 3-18: Requisito no funcional 2.....	28
Tabla 3-19: Usuario-cliente.....	29
Tabla 3-20: Usuario-técnico.....	29
Tabla 3-21: Uprint backlog.....	37
Tabla 3-22: Historia de usuario hc-001.....	38
Tabla 4-1: Tabla de asignación de valores a las respuestas de las preguntas.....	40
Tabla 4-2: Tabulación de resultados.....	41
Tabla 4-3: Cálculos prueba shapiro wilk.....	44
Tabla 4-4: Resultados prueba shapiro wilk.....	45
Tabla 4-5: Resumen de resultados prueba de wilcoxon.....	51

INDICE DE ILUSTRACIONES

Ilustración 3-1: Flujo de proceso de tickets.	18
Ilustración 3-2: Casos de uso – usuarios.	23
Ilustración 3-3: Casos de uso – técnico.....	24
Ilustración 3-4: Arquitectura.	31
Ilustración 3-5: Organización de carpetas backend.....	31
Ilustración 3-6: Organización de carpetas frontend.	32
Ilustración 3-7: Diagrama entidad-relación.	33
Ilustración 3-8: Pantalla de inicio de sesión.....	35
Ilustración 3-9: Pantalla de creación de un ticket.....	36

INDICE DE ANEXOS

ANEXO A: CARTA DE RECEPCIÓN Y ACEPTACIÓN DEL PRODUCTO DE SOFTWARE.

ANEXO B: HISTORIAS DE USUARIO

ANEXO C: ENCUESTA GOOGLE FORMS

ANEXO D: RESULTADO DE LA ENCUESTA

ANEXO E: INTERFACES DE USUARIO

ANEXO F: TEST DE SHAPIRO WILK PARA LAS RESPUESTAS DESDE LA PREGUNTA

4 HASTA LA PREGUNTA 10.

ANEXO G: MANUAL TÉCNICO

RESUMEN

En esta investigación, se presenta el desarrollo del sistema móvil con seguridad biométrica para gestionar tickets de incidencias TI en las sucursales de Riobamba de la cooperativa de ahorro y crédito Fernando Daquilema. A lo largo de esta tesis, denominaremos este sistema como "Daquinet". Para su construcción, se aplicó la metodología ágil Scrum, adoptando prácticas como iteraciones frecuentes, historias de usuario y un enfoque centrado en el usuario. Utilizamos el framework Flutter para el desarrollo del frontend y Spring Boot para el backend. Además, la integración con la API de Redmine permitió una sincronización efectiva con el sistema de gestión de incidencias existente. La Cooperativa de Ahorro y Crédito Fernando Daquilema ha estado presente a lo largo de todo el proceso, estableciendo medidas de calidad y eficiencia. La interfaz de Daquinet se diseñó de manera intuitiva para mejorar la experiencia del usuario. Los resultados obtenidos al final del desarrollo evidencian una mejora significativa en eficiencia en comparación con el sistema anterior. Específicamente, la aceptación de tickets por parte de los técnicos desde su creación experimentó una reducción considerable de tiempo. Daquinet no solo refuerza la seguridad en la gestión de incidencias, sino que también demuestra ser una solución más eficiente en las sucursales de Riobamba de la Cooperativa de Ahorro y Crédito Fernando Daquilema. Los objetivos del trabajo de titulación incluyen determinar el proceso actual de generación de tickets, analizar las herramientas utilizadas para el desarrollo, desarrollar los módulos del sistema y evaluar la mejora en la eficiencia del nuevo sistema automatizado en comparación con el sistema anteriormente utilizado.

Palabras clave: <DAQUINET>, <SEGURIDAD BIOMÉTRICA>, <GESTIÓN DE INCIDENCIAS>, <TECNOLOGÍAS DE LA INFORMACIÓN>, <SCRUM>, <FLUTTER>, <SPRING BOOT>, <API DE REDMINE>.



ABSTRACT

This research presents the development of a mobile system with biometric security to manage IT incident tickets at the Riobamba branches of the Fernando Daquilema Savings and Credit Cooperative. This thesis will refer to this system as "Daquinet." The agile Scrum methodology was applied for its construction, adopting practices such as frequent iterations, user stories, and a user-centered approach. We used the Flutter framework for frontend development and Spring Boot for the backend. Additionally, integration with the Redmine API allowed effective synchronization with the existing incident management system. The Fernando Daquilema Savings and Credit Cooperative has been present throughout the process, establishing quality and efficiency measures. The Daquinet interface was intuitively designed to enhance user experience. The results obtained at the end of development show a significant improvement in efficiency compared to the previous system. Specifically, technicians' acceptance of tickets from their creation experienced a considerable reduction in time. Daquinet not only reinforces security in incident management but also proves to be a more efficient solution in the Riobamba branches of the Fernando Daquilema Savings and Credit Cooperative. The objectives of the thesis work include determining the current process of ticket generation, analyzing the tools used for development, developing the system modules, and evaluating the improvement in efficiency of the new automated system compared to the previous system used.

Keywords:<DAQUINET>, <BIOMETRIC SECURITY>, <INCIDENT MANAGEMENT>, <INFORMATION TECHNOLOGY>, <SCRUM>, <FLUTTER>, <SPRING BOOT>, <REDMINE API>.



Prof. Nelly Padilla. Mgs.

0603818717

DOCENTE FIE

INTRODUCCIÓN

En la era actual de la Tecnología de la Información (TI), las instituciones financieras se enfrentan a desafíos constantes para adaptarse al entorno tecnológico en evolución. La Cooperativa de Ahorro y Crédito Fernando Daquilema, reconocida por su excelencia en servicios financieros, lidia con complejidades en la gestión de incidentes en su entorno de TI. El proceso actual, lento, afecta la eficiencia operativa y la satisfacción de los miembros, lo que podría disminuir la confianza y lealtad hacia la cooperativa. Basándonos en investigaciones previas sobre gestión de incidencias, surge la necesidad de implementar un sistema automatizado de emisión de tickets, llamado "Daquinet". Este sistema busca mejorar la gestión de incidencias, agilizar la comunicación entre miembros y soporte técnico, y reducir los tiempos de respuesta. Al adoptar Daquinet, la cooperativa mejorará su eficiencia operativa y permitirá al equipo de soporte resolver problemas de manera oportuna. Este proyecto propone una solución integral para fortalecer la gestión de incidencias, respaldada por automatización y buenas prácticas de TI.

Capítulo I: En este capítulo se presentan los antecedentes del problema, la justificación del proyecto, los objetivos y la descripción detallada de cómo Daquinet puede agilizar y facilitar el proceso de aceptación de incidencias.

Capítulo II: En este capítulo, se profundiza en el marco teórico de la investigación, incorporando información relevante de fuentes bibliográficas, tales como la obra de Vásquez & Ely(2022), y el estándar ISO 25010. Se destaca particularmente la atención dada a la variable de eficiencia. Es crucial señalar que la finalidad principal de esta revisión teórica es establecer las bases para la medición de la eficiencia, con la norma ISO 25010 asumiendo un papel central en este proceso.

Capítulo III: Se detallan los métodos y técnicas utilizados en el proyecto, incluyendo la metodología de desarrollo Scrum y un estudio de factibilidad para respaldar la viabilidad del sistema Daquinet.

Capítulo IV: Este capítulo presenta los resultados del análisis de la variable de la eficiencia en Daquinet. Se concluye con observaciones finales y recomendaciones derivadas del trabajo realizado en este proyecto de integración curricular.

CAPÍTULO I

1 PLANTEAMIENTO DEL PROBLEMA

1.1 Antecedentes

La Cooperativa de Ahorro y Crédito Fernando Daquilema se destaca como una renombrada institución financiera, que ha estado desempeñando un papel esencial ofreciendo sus servicios financieros a sus miembros. Sin embargo, en el entorno actual de Tecnología de la Información (TI), la cooperativa encuentra con diversos desafíos y complejidades. En particular, la gestión de incidentes en el entorno de TI está siendo un proceso lento e ineficiente en la cooperativa. El sistema actual ha demostrado ser inadecuado para satisfacer las necesidades cambiantes y exigentes de la entidad. Administrar correctamente las revisiones de forma manuales, las solicitudes de soporte y los problemas de tipo técnicos requiere mucho tiempo y esfuerzo, este proceso tiene un impacto negativo en la eficiencia operativa de la cooperativa y la satisfacción de los miembros. Los retrasos en la resolución de problemas y la falta de una supervisión adecuada pueden generar insatisfacción, lo que puede provocar una disminución de su confianza y lealtad hacia la cooperativa.

Palomo Pastor (2019) definen las incidencias como “cualquier interrupción o disminución no prevista de la calidad de un servicio”. En el contexto de este estudio, se da especial atención a esta conceptualización, ya que proporciona un marco claro para la comprensión de posibles desafíos en la calidad del servicio.

La investigación de Vásquez & Ely (2022), resalta la significativa relevancia de la gestión de incidencias en el ámbito del soporte informático. Su objetivo central radica en proponer un sistema informático que potencie la eficacia de esta gestión, al tiempo que identifica los criterios esenciales para abordar los desafíos inherentes a la mejora de dicho proceso. En la construcción de su marco teórico, se exploran antecedentes tanto a nivel nacional como internacional, incorporando fundamentos teóricos que se centran en la definición de un sistema de help desk. Además, se enfoca en la integración de ITIL (Information Technology Infrastructure Library) y aborda el desarrollo basado en la Metodología XP (Extreme Programming). Asimismo, se dedica a la delineación de un proceso de gestión de incidencias. A la luz de estas contribuciones, se establece un vínculo con la presente investigación, donde se optará por el enfoque Scrum en la gestión de incidencias, siguiendo las recomendaciones y pautas que esta tesis proporciona encontrando así la metodología y el enfoque que se necesitará para la construcción de Daquinet.

Calle (2015) en su tesis Diseño del Sistema de Control Interno en la Empresa Ab Optical de la Ciudad de Loja concluye que “el control interno es un curso de acción para alcanzar los objetivos de las empresas y promover el ordenamiento de los procesos. Además, destaca que el control interno ayuda a controlar las desviaciones de las tareas y los principios organizativos, lo que a su vez fortalece la gestión de calidad, eficiencia y eficacia, lo cual se traduce en una mejor prestación de servicios”.

Ante esta problemática, se plantea la necesidad de implementar un sistema de emisión de tickets automatizado que permita mejorar el proceso de gestión de incidencias y, en última instancia, brindar un mejor servicio a los miembros de la cooperativa. Mediante la automatización de este proceso, se espera agilizar la comunicación entre los miembros y el equipo de soporte técnico, reducir los tiempos de respuesta y proporcionar una mayor transparencia en el seguimiento de las incidencias.

La adopción de un sistema de emisión de tickets no solo mejoraría la eficiencia operativa de la cooperativa, sino que también permitiría al equipo de soporte técnico de la cooperativa de Fernando Daquilema concentrarse en resolver los problemas de manera oportuna y efectiva. Así mismo, se proporcionaría un mecanismo para recopilar datos y generar informes, lo que facilitaría la identificación de patrones de incidencias recurrentes y la toma de decisiones basadas en datos.

1.1.1 Formulación del Problema

¿Cómo la implementación de un sistema automatizado de tickets influye en la eficiencia del proceso de gestión de incidencias en la cooperativa de ahorro y crédito Fernando Daquilema?

1.1.2 Sistematización del Problema

¿Cuál es el proceso de la generación de tickets en la cooperativa?

¿Cómo se puede utilizar la herramienta Flutter, Spring Boot y Redmine para mejorar la gestión de incidencias?

¿Cuál es el nivel de mejora de la eficiencia con la implementación de un sistema automatizado de tickets en la cooperativa?

1.2 Justificación

1.2.1 Justificación teórica

Al explorar investigaciones anteriores, se puede identificar cómo los sistemas automatizados de emisión de tickets han demostrado mejorar la eficiencia de la gestión de incidencias. Herramientas como Redmine facilitan la comunicación entre los usuarios y el equipo de soporte técnico, lo que permite una rápida asignación de incidentes, un seguimiento efectivo y tiempos de respuesta más cortos.

Campos (2013) explica que Redmine permite gestionar múltiples proyectos desde una sola interfaz de una ventana de navegador. Los proyectos pueden ser totalmente independientes unos de otros, o complementarios siguiendo una estructura jerárquica. La navegación es muy sencilla y se puede saltar y cambiar de proyecto en cualquier momento.

El framework Flutter permite una personalización mayor a los beneficios que nos ofrece Redmine, el autor Lizárraga (2020) presenta ventajas del uso de Flutter como el desarrollo multiplataforma que evita que construyamos por separado dándonos la aplicación utilizable para los diferentes sistemas operativos actuales y a su vez nos da acceso a las funciones nativas de cada sistema.

Ramírez et al. (2018) indica que Scrum permite un desarrollo más ágil que es apropiado a la hora de realizar este tipo de sistemas por la gran necesidad de mantener comunicación en todo momento con los involucrados, se recomienda en proyectos con entornos complejos, con requerimientos que se modifican y que además es necesario obtener resultados rápidos, en donde la innovación, la flexibilidad y la productividad son básicas.

1.2.2 Justificación aplicativa

La implementación de un sistema automatizado de tickets trae importantes beneficios prácticos a la cooperativa.

En primer lugar, agilizaría la comunicación entre los miembros y el equipo de soporte. Esto da como resultado una asignación de tickets más rápida y eficiente, lo que reduce los tiempos de respuesta y mejora la satisfacción de los miembros.

En segundo lugar, libera tiempo y recursos del equipo de la mesa de ayuda. Esto les permite concentrarse en la resolución oportuna y efectiva de problemas, lo que mejora la eficiencia operativa de la cooperativa al reducir el tiempo de inactividad y minimizar los impactos negativos en las operaciones de los miembros.

En tercer lugar, proporciona a la cooperativa un mejor seguimiento de los incidentes. Esto permite a la cooperativa generar informes y analizar datos para identificar patrones y tendencias de incidentes recurrentes. Esta información podría usarse para tomar decisiones informadas y proactivas para abordar problemas recurrentes y mejorar continuamente los servicios ofrecidos.

En general, la implementación de un sistema automatizado de tickets de incidencias sería una inversión que valdría la pena para la Cooperativa de Ahorro y Crédito Fernando Daquilema. Mejoraría la satisfacción de los miembros, la eficiencia operativa y el monitoreo y seguimiento de incidentes.

Los módulos que formarían parte del sistema de tickets serían:

Módulo de creación de tickets: Permite a los miembros de la cooperativa generar nuevos tickets o incidencias, proporcionando detalles sobre el problema o consulta que desean reportar.

Módulo de asignación de tickets: Facilita la asignación rápida y eficiente de los tickets a los miembros del equipo de soporte técnico o especialistas correspondientes. Este módulo puede utilizar criterios como la disponibilidad, las habilidades o la carga de trabajo para asignar los tickets de manera óptima.

Módulo de seguimiento de tickets: Permite a los miembros de la cooperativa y al equipo de soporte técnico realizar un seguimiento del estado y el progreso de los tickets.

Módulo de generación de informes: Recopila y analiza los datos relacionados con los tickets para generar informes y estadísticas sobre la cantidad de tickets, tiempos de respuesta, problemas recurrentes, entre otros indicadores clave. Estos informes pueden utilizarse para evaluar la eficiencia y la calidad del servicio, identificar áreas de mejora y respaldar la toma de decisiones basada en datos.

Esta investigación se enmarca en el Plan Nacional de Desarrollo en el eje económico, teniendo como objetivo impulsar los sistemas económicos con reglas claras que fomenten el comercio

exterior, turismo, la atracción de los inversionistas y la modernización de los sistemas en las entidades financieras nacionales. La política 2.2 del Plan Nacional de Desarrollo enfatiza en promover un adecuado entorno de negocios que permita la atracción de inversiones y las asociaciones público-privadas, lo cual se relaciona perfectamente con la gestión adecuada de las incidencias que se pueden presentar dentro de la cooperativa.

La presente investigación se encuentra alineada dentro de la ESPOCH con el eje de TICS dentro de la línea de Tecnologías de la información y comunicación del programa de Ingeniería en software, permitiendo así automatizar los procesos y fomentar la actualización de los sistemas para garantizar una mejor gestión de los procesos.

1.3 Objetivos

1.3.1 Objetivo general

Desarrollar un sistema móvil con seguridad biométrica para gestionar tickets de incidencias TI en las sucursales de Riobamba de la Cooperativa de Ahorro y Crédito Fernando Daquilema.

1.3.2 Objetivos específicos

Determinar el proceso actual de generación de tickets en la cooperativa.

Analizar los conceptos y características de las herramientas Flutter, Spring Boot y Redmine para el desarrollo de sistemas de software.

Desarrollar los módulos de creación de tickets, asignación de tickets, seguimiento de tickets y generación de informes de la justificación utilizando la metodología SCRUM.

Evaluar la mejora en la eficiencia entre el sistema automatizado de tickets desarrollado y el sistema de gestión de incidencias actualmente utilizado en la cooperativa.

CAPÍTULO II

2 MARCO TEÓRICO

2.1 Soporte Técnico

El soporte técnico es una parte clave en cualquier organización que dependa de la tecnología de la información (TI) para sus operaciones, pues esta proporciona asistencia y resuelve los problemas técnicos que puedan surgir en los sistemas y aplicaciones utilizados. En el contexto del sistema a desarrollar, el soporte técnico juega un papel clave en garantizar la eficacia y utilidad de este a largo plazo.

La rápida respuesta del soporte técnico es primordial para minimizar el tiempo de inactividad, cuando un sistema falla o los usuarios encuentran dificultades técnicas, contar con un equipo de soporte técnico ágil y bien preparado puede marcar la diferencia entre una breve interrupción y una interrupción prolongada que afecte las operaciones y los resultados del producto. Mantener una rápida asignación de tickets y facilitar el seguimiento del estado de resolución de cada solicitud de soporte, asegurando una respuesta oportuna y una solución eficiente de la mano de una eficiente resolución de problemas técnicos, también juega un papel importante en la optimización de los recursos de TI para el producto. Este sistema permite la clasificación y priorización de los tickets según su nivel de urgencia, gravedad o impacto en las operaciones.

2.2 Eficiencia

Server Izquierdo & Navarro (2001) define a la eficiencia como el grado de bondad u optimalidad alcanzada en el uso de los recursos para la producción de los servicios bancarios. Es asociado con la proximidad entre el nivel que existe en la productividad, definido por la relación técnica que existe entre recursos utilizados y la producción de un bien o servicio financiero obtenido de un negocio en particular y el valor máximo que se puede obtener en determinadas circunstancias.

2.3 Lenguaje de programación Dart

Fernández Muñoz (2017) menciona que dart es un lenguaje de programación que surgió en 2011, de código abierto y desarrollado por Google. Este lenguaje reemplazó al JavaScript de Google y se complementa con el framework Flutter dándole gran poder y alcance.

Amazon (2018) menciona que una de las características de Dart al complementarse con Flutter es la seguridad de nulos. Esta seguridad de los nulos de Dart puede detectar fácilmente los errores más comunes, llamados errores nulos. Esta característica reduce el tiempo que los desarrolladores dedican a mantener el código, dándoles más tiempo para centrarse en la creación de sus aplicaciones.

Google (2023) en su documentación oficial de Dart explica que es un lenguaje versátil que se puede usar para desarrollar una amplia variedad de aplicaciones, incluidas aplicaciones web, aplicaciones móviles y aplicaciones de escritorio. También es una buena opción para el desarrollo del lado del servidor. Dart es un lenguaje popular entre los desarrolladores y está creciendo en popularidad.

Estas son algunas de las características de Dart según Google:

“Rápido: las aplicaciones de Dart se compilan en código de máquina, lo que las hace más rápidas que las aplicaciones de JavaScript.

Tipo seguro: Dart es un lenguaje de tipo seguro, lo que significa que el compilador puede comprobar si hay errores en el momento de la compilación.

Versátil: Dart se puede utilizar para desarrollar una amplia variedad de aplicaciones, incluidas aplicaciones web, aplicaciones móviles y aplicaciones de escritorio.

Popular: Dart es un lenguaje popular entre los desarrolladores y está creciendo en popularidad”.
Google (2023).

2.4 Flutter

Tashildar et al. (2020) define Flutter como un Software Development Kit (SDK) de código abierto destinado a la creación de aplicaciones móviles de alto rendimiento y confiabilidad, diseñadas para operar en sistemas operativos como iOS y Android. Entre las características destacadas de Flutter se encuentra su capacidad de compilación Just-In-Time (JIT), que ejecuta el código de manera dinámica durante la ejecución del programa en tiempo de ejecución, en contraste con los métodos de compilación previa a la ejecución. Esta característica específica contribuye significativamente a la mejora del rendimiento y la eficiencia en el desarrollo de aplicaciones móviles.

Según el autor del sitio web Amazon (2018) señala que Flutter destaca en varios aspectos como marco de desarrollo multiplataforma:

Rendimiento casi nativo, El lenguaje de programación utilizado por Flutter se llama Dart y la programación hecha en esta se compila en código de máquina. Los dispositivos entienden el código que fue generado, lo que garantiza un rendimiento rápido y eficiente.

Posee un rendimiento bastante rápido, consistente y personalizable. En vez de estar ligados al uso de herramientas de renderizado específicas de la plataforma, Flutter se promueve el uso de la biblioteca de gráficos Skia de código abierto de Google para la renderización de las interfaces de usuario.

Es una herramienta utilizada para desarrolladores. Google tiene como uno de sus objetivos que su framework Flutter sea fácil de manejar. La recarga en vivo permite a los desarrolladores poseer una gran ventaja puesto que permite previsualizar el aspecto de los cambios de código sin perder el estado.

Según Devs (2019) indica algunas de las características principales del framework Flutter.

Las aplicaciones nativas en Flutter se desarrollan directamente para el sistema operativo, por lo general este framework aprovecha al máximo las aplicaciones nativas y así poder lograr resultados de calidad.

Flutter utiliza el Material Design de Google y Cupertino de Apple, lo que permite a los usuarios utilizar herramientas desarrolladas por la propia empresa que han solicitado dicho diseño.

Las razones principales para deshacerse de una aplicación es el tiempo que se requiere en ello instalando y viceversa, es por eso por lo que el tiempo de carga del framework de Flutter es nada menos que de un segundo en cualquier dispositivo de iOS o Android, cuenta con esa gran compatibilidad.

2.5 Redmine

Según Gata & Gata (2010) redmine es un gestor de proyectos escrito usando Ruby en Rails framework y se puede utilizar en múltiples plataformas y bases de datos. Es una aplicación de código abierto y es lanzado por GNU General Licencia Pública v2 (GPL), en su funcionamiento

Redmine tiene muchas herramientas para apoyar la gestión de proyectos, tales como: soporte para múltiples proyectos, por lo que puede gestionar todos los proyectos en curso, flexibilidad en lo básico, reglas de control de acceso, flexibles en la tarea de rastreo del sistema, también muestra el Diagrama de Gantt y calendario de trabajo del proyecto, noticias, documentación y gestión de archivos, también tiene la funcionalidad de enviar un correo electrónico de notificación.

Lesyuk, I. (2014) indica que nadie puede decir si Redmine es más una herramienta de gestión de proyectos o un rastreador de problemas, un buen rastreador de problemas debe tener algo de gestión de proyectos. En Redmine estos dos componentes se combinan a la perfección. Pero lo que hace que sea una aplicación de seguimiento de problemas perfecta es el flujo de trabajo totalmente configurable, que permite definir permisos de cambio de estado de problemas para cada rol y rastreador (Tipo de problema). Como rastreador de problemas, Redmine también admite funciones esenciales como prioridades, subproblemas, visualización, comentarios, campos personalizados, filtros de lista y más.

2.6 Scrum

Takeuchi et al. (1986) mencionan que Scrum es una de las metodologías de desarrollo ágil de Software más reconocidas a nivel mundial, su concepción data de los años 80 en análisis realizados por Ikujiro Nonaka e Hirotaka Takeuchi, en el cual resaltaron el trabajo en equipo para el desarrollo de productos y la autonomía que estos equipos deben tener.

Según Schwaber, K. y Sutherland, J. (2020) nos muestra que la metodología Scrum tiene como núcleo el empirismo y el pensamiento Lean. Esta metodología indica que el conocimiento se abstrae de la experiencia y que las decisiones se las toma basándose en la observación. El pensamiento Lean trata de la reducción del desperdicio y se centra en lo esencial. Scrum utiliza un enfoque iterativo e incremental que permite la optimización, previsibilidad y control de riesgos, permitiendo involucrar a personas que en un contexto grupal poseen todas las habilidades y experiencia para lograr el trabajo establecido y hacerlos compartir o adquirir estas habilidades según vaya siendo necesario.

2.7 Sistemas para la gestión de incidencias existentes

2.7.1 *IntegriaIMS (Integria Incident Manager Sistem)*

Integria (2018) indica que IntegriaIMS es un software que sirve para brindar asistencia sencilla, potente y es muy fácil de utilizar. Dispone de inventario automatizado y controlado desde una sola interfaz. Permite agilizar los procesos generando reducción en los tiempos y costes de la gestión de proyectos. Emite informes de manera personalizada para observar de manera fácil la gestión y calidad del servicio en tiempo real.

2.7.2 *Open Technology Real Services (OTRS)*

OTRS (2017) dice que es un software gratuito que posee su código abierto y brinda soluciones de software de negocios profesionales. OTRS es uno de los varios sistemas de emisión de tickets en línea con gran flexibilidad y se utiliza en la asistencia a los clientes y la gestión de los múltiples servicios que puede presentarse dentro de la informática. De implementación fácil y con una adaptación sencilla a diversidad de necesidades.

2.7.3 *Manage Engine Service Desk Plus*

Zoho Corp (2017) señala que esta herramienta permite consolidar todos los recursos de soporte para múltiples clientes en una única plataforma de software, todos los recursos de soporte para múltiples clientes. Se puede personalizar el servicio para cada cliente y gestionar de forma individual los aspectos más importantes del servicio. Unas de las prestaciones que brinda son la facturación automática, catálogo de servicios, asignación automática, gestión de incidencias, inventario de hardware y software, entre otros.

2.8 Trabajos relacionados

Según dice Fernández Montesinos (2014) en el campo de la gestión de incidencias existen multitud de soluciones informáticas que se pueden considerar de gran utilidad a las empresas o a grupos de individuos que son parte de una organización que utiliza sistemas informáticos.

En la actualidad, los sistemas de tickets para el soporte técnico han evolucionado significativamente para satisfacer las demandas de las organizaciones en términos de eficiencia y gestión de solicitudes, como reconoce Acuña (2013) “El control interno dentro de las empresas es de gran importancia al igual que la influencia en el buen andar de las finanzas. Un sistema de control interno óptimo permite un nivel operativo eficiente y por consiguiente trasciende en los resultados.”

Aguirre Zegarra (2019) demostró que el sistema de tickets es indispensable para la mejora del servicio de Help Desk en la empresa minera, lo que contribuye a ahorrar tiempos en atender las incidencias, llevar un control de estos y brindar un excepcional servicio a los usuarios de la mina.

Según Pozo & Javier (2017) es primordial que se optimicen los recursos en todas las entidades a fin de garantizar la satisfacción el cliente. Además, cabe destacar que la Informática en general es una de las herramientas más trascendentales para el desarrollo institucional de un ente económico, de ahí que se necesita renovar la tecnología para tener una mejor perspectiva empresarial y mejorar sus recursos financieros, profesionales y materiales como indica el investigador que en su tesis plantea el desarrollo de un sistema de administración de tickets para atención de clientes y mejorar la solución de incidencias y reclamos.

En la actualidad, los sistemas de tickets para el soporte técnico se integran con herramientas de gestión de servicios y ofrecen capacidades avanzadas de análisis y generación de informes, además de medir la satisfacción del usuario. Estas características mejoran la eficiencia y calidad del servicio de soporte técnico.

Según Gonzales (2021) en su investigación planteó como objetivo, como el implementar un Service Desk mejora las incidencias que los clientes presentan en una institución pública, utilizando la metodología SCRUM para aplicar mejoras a lo largo del proyecto implemento el sistema de tickets para la gestión de incidencias con el fin de administrar y desarrollar los diversos servicios utilizando diferentes herramientas para identificar los procesos principales.

De igual manera, Benites et al., (2017) en su investigación determinaron que el principal problema es que no existe un sistema que genere tickets para atención en el área de Ceuci de la Universidad Villareal, lo que causa un descontrol en la secuencia de orden de atención, confusión entre las áreas y que no sea oportuna la respuesta a las incidencias. De esta manera, al igual que en el estudio mencionado, la introducción de un sistema automatizado de tickets en la cooperativa no solo contribuirá a la eficiencia en la gestión de incidencias, sino que también establecerá un orden entre los departamentos involucrados, generará ahorro de tiempo, priorizará las incidencias y permitirá identificar patrones para la toma de decisiones beneficiosas a corto y largo plazo para la cooperativa. Así, la propuesta de implementar un sistema móvil automatizado encuentra su fundamento en la necesidad de superar el descontrol identificado en la gestión de tickets, siguiendo la línea de pensamiento y los beneficios evidenciados.

CAPÍTULO III

3 MARCO METODOLÓGICO

3.1 Tipo de estudio

El presente trabajo consta de tres diferentes tipos de estudio que son:

3.1.1 *Investigación Aplicada*

Según Pozo & Javier (2017) es primordial que se optimicen los recursos en todas las entidades a fin de garantizar la satisfacción el cliente. Además, cabe destacar que la Informática en general es una de las herramientas más trascendentales para el desarrollo institucional de un ente económico, de ahí que se necesita renovar la tecnología para tener una mejor perspectiva empresarial y mejorar sus recursos financieros, profesionales y materiales como indica el investigador que en su tesis plantea el desarrollo de un sistema de administración de tickets para atención de clientes y mejorar la solución de incidencias y reclamos.

En la actualidad, los sistemas de tickets para el soporte técnico se integran con herramientas de gestión de servicios y ofrecen capacidades avanzadas de análisis y generación de informes, además de medir la satisfacción del usuario. Estas características mejoran la eficiencia y calidad del servicio de soporte técnico.

Según Gonzales (2021) en su investigación planteó como objetivo, como el implementar un Service Desk mejora las incidencias que los clientes presentan en una institución pública, utilizando la metodología SCRUM para aplicar mejoras a lo largo del proyecto implemento el sistema de tickets para la gestión de incidencias con el fin de administrar y desarrollar los diversos servicios utilizando diferentes herramientas para identificar los procesos principales.

De igual manera, Benites et al., (2017) en su investigación determinaron que el principal problema es que no existe un sistema que genere tickets para atención en el área de Ceuci de la Universidad Villareal, lo que causa un descontrol en la secuencia de orden de atención, confusión entre las áreas y que no sea oportuna la respuesta a las incidencias. De esta manera, al igual que en el estudio mencionado, la introducción de un sistema automatizado de tickets en la cooperativa no solo contribuirá a la eficiencia en la gestión de incidencias, sino que también establecerá un orden

entre los departamentos involucrados, generará ahorro de tiempo, priorizará las incidencias y permitirá identificar patrones para la toma de decisiones beneficiosas a corto y largo plazo para la cooperativa. Así, la propuesta de implementar un sistema móvil automatizado encuentra su fundamento en la necesidad de superar el descontrol identificado en la gestión de tickets, siguiendo la línea de pensamiento y los beneficios evidenciados.

3.1.2 Investigación cuantitativa

La necesidad de recopilar datos cuantitativos relacionados con el proceso de resolución de incidencias establece a este trabajo como una investigación cuantitativa. Estos datos numéricos permitirán evaluar la magnitud del problema a medir y el impacto de las soluciones propuestas.

3.1.3 Investigación cuasiexperimental

Se utiliza este tipo de investigación puesto que se va a trabajar con variables en un entorno controlado y sin asignación aleatoria de los participantes.

3.2 Métodos y Técnicas

En la Tabla 3-1 se detalla los diferentes métodos y técnicas que se utilizarán para cumplir el presente trabajo de integración curricular.

Tabla 3-1: Objetivo, métodos, técnicas y fuentes a usar en base a los objetivos específicos especificados.

Objetivo	Métodos	Técnicas	Fuentes
Determinar el proceso actual de generación de tickets en la cooperativa.	Análisis	Análisis y entrevistas, BPMN (estándar para representación de procesos), observación directa.	Documentación interna, entrevistas con empleados y miembros.
Analizar los conceptos y características de las herramientas Flutter y Redmine para el desarrollo de sistemas de software.	Inductivo	Revisión de documentación	Documentación oficial, comunidades en línea y libros, papers y revistas.
Desarrollar los módulos de creación de tickets,	SCRUM	Reuniones. Planificación de sprint.	Flujo de trabajo actual. Personal de soporte

asignación de tickets, seguimiento de tickets y generación de informes de la justificación utilizando la metodología SCRUM.		Revisiones.	técnico.
Evaluar la mejora en la eficiencia entre el sistema automatizado tickets desarrollados y el sistema de gestión de incidencias actualmente utilizado en la Cooperativa.	Experimentación	Observación y análisis de datos	Aplicación, software para medir la eficiencia.

Fuente: Mejia, D. 2024

Realizado por: Mejia, D. 2024

3.2.1 Unidad de análisis

Para medir la eficiencia se tomó como unidad de análisis a los trabajadores de las sucursales en Riobamba de la Cooperativa de Ahorro y Crédito Fernando Daquilema.

3.2.2 Instrumentos

Se utilizarán cuestionarios como instrumentos para recopilar información, facilitando así el análisis estadístico.

3.2.3 Población y muestra

La población son los trabajadores de las sucursales en Riobamba de la Cooperativa de Ahorro y Crédito Fernando Daquilema.

3.2.4 Operacionalización conceptual y metodológica de las variables

En el marco de la evaluación de la eficiencia de Daquinet, se adopta el estándar ISO/IEC 25010, el cual desglosa la eficiencia en dos sub-características fundamentales: desempeño y uso eficiente de recursos.

En el marco de la evaluación de la eficiencia de Daquinet, se presenta la Tabla 3-2 Sub características de eficiencia de desempeño con cada métrica respectivamente para detallar las dimensiones críticas de rendimiento y uso eficiente de recursos

Tabla 3-2: Sub-características de eficiencia de desempeño con cada métrica respectivamente.

Sub-característica de rendimiento	Métricas
Comportamiento en el tiempo	<ul style="list-style-type: none"> ● Tiempo de respuesta ● Tiempo de espera ● Rendimiento
Utilización de recursos	<ul style="list-style-type: none"> ● Utilización de CPU ● Utilización de memoria ● Utilización de dispositivos de Entrada y Salida
Capacidad	<ul style="list-style-type: none"> ● Número de peticiones en línea ● Número de accesos simultáneos ● Sistemas de transmisión de ancho de banda

Fuente: Mejia, D. 2024

Realizado por: Mejia, D. 2024

3.3 Ambientes de prueba

3.3.1 Análisis Estadístico

Las respuestas de la encuesta serán sometidas a un análisis estadístico para evaluar la eficiencia percibida en la gestión de incidencias entre el sistema anterior y el nuevo sistema "Daquinet". Se empleará un enfoque de análisis de tendencias para identificar patrones generales en la percepción de velocidad y eficiencia. Posteriormente, se aplicará la prueba de pares relacionados de Wilcoxon para muestras independientes, con el objetivo de determinar si existen diferencias estadísticamente significativas entre las medias de los dos grupos. Los resultados obtenidos en este análisis proporcionarán información valiosa sobre la eficiencia percibida en la gestión de incidencias, respaldando la evaluación comparativa entre el sistema anterior y el nuevo sistema "Daquinet".

3.3.1.1 Prueba de Wilcoxon

La prueba no parametrizada de Wilcoxon se utiliza para comparar dos muestras relacionadas de un mismo grupo de personas que han experimentado tanto con el sistema anterior como con el

sistema nuevo Daquinet. La prueba de Wilcoxon se la utiliza cuando nuestra muestra de datos no cumple con una prueba de normalidad que requieren los métodos paramétricos.

3.3.1.2 Análisis de tendencias

El análisis de tendencias se lleva a cabo con el fin de identificar patrones en las respuestas de los participantes en relación con la eficiencia percibida en la gestión de incidencias. Se calculan medidas estadísticas para cada conjunto de datos, proporcionando una visión completa de la distribución y tendencia central de las respuestas.

3.4 Proceso de gestión de incidencias de la cooperativa

3.4.1 Proceso actual de gestión de incidencias de la cooperativa

1. Recepción de la incidencia:

- El cliente envía un correo electrónico a la dirección de soporte técnico de la empresa o mediante una solicitud desde el sistema redmine de gestión de tickets interno de la empresa.
- Un agente de soporte técnico lee el correo electrónico y extrae la información relevante, como el tipo de incidencia, la prioridad y la descripción del problema.

2. Asignación del ticket:

- El agente de soporte técnico decide a qué agente o equipo asignar la incidencia en función de la disponibilidad, la experiencia y el tipo de incidencia.
- El agente o equipo asignado recibe una notificación por correo electrónico o a través del sistema redmine interno de gestión de tickets.

3. Resolución de la incidencia:

- El agente o equipo asignado analiza la información del ticket y se comunica con el cliente para obtener más información si es necesario.
- El agente o equipo resuelve la incidencia.
- El agente o equipo actualiza el estado del ticket en un sistema interno de gestión de tickets o manualmente.

4. Cierre del ticket:

- El agente o equipo informa al cliente de la resolución de la incidencia.
- El cliente cierra el ticket o solicita una revisión adicional.

5. Seguimiento de la incidencia:

El seguimiento de la incidencia se realiza de forma manual, lo que puede ser ineficiente y propenso a errores.

3.4.2 *Proceso actual de gestión de incidencias con el sistema implementado*

Se exhibe la Ilustración 1-3: Flujo de Proceso de Tickets para brindar una representación gráfica del ciclo de vida y la secuencia de acciones en la gestión de tickets mediante Daquinet. Esta ilustración sirve como una herramienta visual integral, proporcionando una comprensión clara y detallada de la dinámica operativa del sistema en el ámbito de la gestión de incidencias.

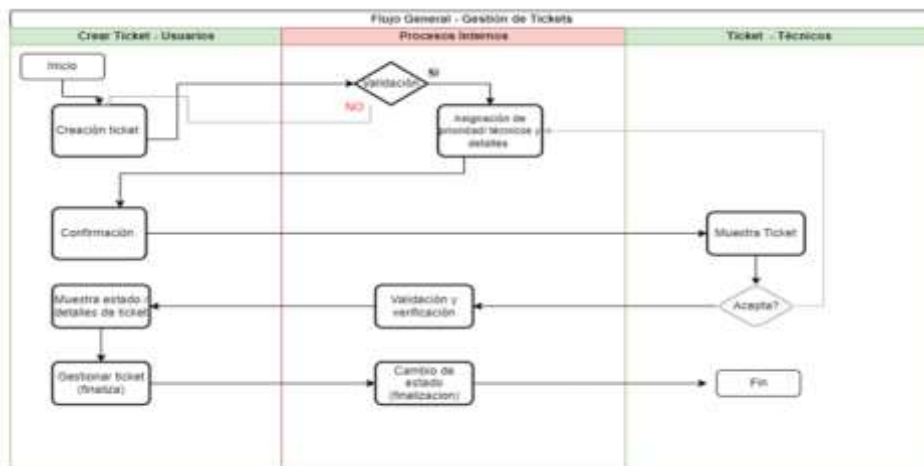


Ilustración 3--1: Flujo de proceso de tickets.

Realizado por: Mejía, D. 2024

3.5 Estudio de factibilidad

El estudio de factibilidad determinará si la implementación del nuevo sistema móvil para la gestión de tickets es viable desde el punto de vista técnico, económico y operativo.

3.5.1 *Factibilidad técnica*

La factibilidad técnica es un componente crucial del estudio de factibilidad para la implementación del nuevo sistema móvil de gestión de tickets. Este análisis determinará si es posible desarrollar e implementar el sistema con la tecnología disponible, considerando los recursos de hardware, software y humanos necesarios. En la Tabla 3-3, 3-4 y 3-5 se detalla los recursos de requeridos.

Tabla 1-3: Recursos de hardware requeridos.

Cantidad	Descripción	Observación
1	Laptop Dell Inspiron 3593	Utilizada para el desarrollo de la aplicación
1	Impresora EPSON	Utilizada para la impresión de documentos
1	Celular Android	Pruebas
1	Celular Iphone	Pruebas

Fuente: Mejia, D. 2024

Realizado por: Mejia, D. 2024

Tabla 3-4: Recursos de software requeridos.

Cantidad	Descripción	Observación
1	Visual Studio Code	Utilizada para el desarrollo frontend de la aplicación.
1	Eclipse	Utilizada para el desarrollo backend de la aplicación.
1	Flutter	Framework para el frontend
1	Spring Boot	Framework para el backend
1	MySQL	Gestor de base de datos

Fuente: Mejia, D. 2024

Realizado por: Mejia, D. 2024

Tabla 3-5: Recursos humanos requeridos.

Cantidad	Descripción	Observación
1	Desarrollador	Conocimiento en Flutter y Spring Boot

Fuente: Mejia, D. 2024

Realizado por: Mejia, D. 2024

Basado en la información proporcionada en las Tablas 3-3, 4-3 y 5-3, se puede concluir que la aplicación es viable desde el punto de vista técnico puesto que el software requerido es libre y el hardware faltante es proporcionado por la cooperativa.

3.5.2 *Factibilidad operativa*

Teniendo en cuenta el concepto de factibilidad operativa, se puede determinar que la aplicación móvil para la gestión de incidencias es viable porque es fácil de usar, eficiente, permite el seguimiento y control de los envíos, es segura y ofrece soporte técnico a los usuarios.

3.5.3 *Factibilidad económica*

Se determinará el costo beneficio de la aplicación móvil de gestión de incidencias que se desarrollará y nos permitirá observar si es factible o no su elaboración.

3.5.3.1 *Costos*

El presente estudio tiene como objetivo analizar la viabilidad económica del sistema móvil de gestión de incidencias. Para ello, se ha realizado una estimación de los costos del proyecto, que se presentan en la tabla 3-6.

Tabla 3-6: Costos.

Descripción	Costo/Mes	Mes	Total (\$)
Luz	15	4	40.00
Internet	30	4	120.00
Laptop Dell Inspiron 3593			1200.00
Celular Android			250.00
Celular Iphone			600.00
Costo Total del proyecto			2210.00

Fuente: Mejia, D. 2024

Realizado por: Mejia, D. 2024

3.5.3.2 *Beneficios*

Un sistema móvil de gestión de incidencias puede ofrecer una serie de beneficio, entre ellos:

Para los empleados:

- Mayor facilidad para reportar incidencias: Los empleados pueden reportar incidencias de forma rápida y sencilla desde cualquier lugar, utilizando su teléfono móvil.
- Mejora en la comunicación: La aplicación móvil puede facilitar la comunicación entre los empleados y el departamento de mantenimiento, permitiendo a los empleados realizar un seguimiento del estado de sus incidencias y recibir retroalimentación sobre las mismas.
- Mayor eficiencia: La aplicación móvil puede ayudar a agilizar el proceso de resolución de incidencias, lo que puede mejorar la productividad de los empleados.

Para la empresa:

- Mejora en la gestión de incidencias: El sistema móvil puede ayudar a las empresas a mejorar la gestión de incidencias al proporcionarles una herramienta centralizada para registrar, rastrear y resolver incidencias.
- Reducción de costos: El sistema móvil puede ayudar a las empresas a reducir los costos asociados a la gestión de incidencias, al reducir el tiempo y los recursos necesarios para resolverlas.
- Mejora en la satisfacción del cliente: El sistema móvil puede ayudar a mejorar la satisfacción del cliente al proporcionarles una forma rápida y sencilla de reportar incidencias y recibir información sobre el estado de estas.
- Mejora en la seguridad: El sistema móvil cuenta con un ingreso mediante seguridad biométrica por ello la seguridad para su acceso es mucho mejor.

3.6 Desarrollo de la aplicación mediante Scrum

SCRUM es una metodología ágil que facilita la comunicación directa con los interesados y se enfoca en la entrega constante de incrementos del producto final, conocidos como "Sprints", con el objetivo de validar las funcionalidades del producto de manera recurrente.

3.6.1 Fase de planificación

Durante la fase de planificación, se lleva a cabo un análisis exhaustivo del problema con el objetivo de definir los tipos de usuarios del sistema y los requisitos establecidos por el cliente. Para recopilar esta información de manera efectiva, se realizan reuniones con todos los participantes clave en el proyecto.

3.6.1.1 Miembros y Roles

En la Tabla 3-7, se presenta una descripción detallada de los miembros del equipo y sus respectivos roles en el desarrollo del proyecto. Esta información es crucial para establecer una comunicación clara y una distribución eficiente de responsabilidades.

Tabla 3-7: Miembros y roles.

Miembro	Rol
Ing. Fredy Janeta	Product Owner y Scrum Master
Dennys Mejia	Development

Fuente: Mejia, D. 2024

Realizado por: Mejia, D. 2024

3.6.1.2 Tipos de usuarios

En la Tabla 3-8, se detalla al Usuario- Cliente junto con sus roles correspondientes, cada uno asignado con funcionalidades específicas dentro del sistema.

Tabla 3-8: Usuario-Cliente.

Actor	Cliente
Rol	Empleado de la empresa
descripción	Este es el actor específico que asume el rol de “Cliente o Empleado de la Empresa” en el sistema. Cada empleado que accede y utiliza la aplicación móvil para interactuar con la funcionalidad de asistencia representa este actor. Los usuarios, generan, cancelan, visualizan las solicitudes de asistencia a través de la creación de tickets.
Requerimientos asociados	<ul style="list-style-type: none">● RF01● RF02● RF03● RF04● RF05

Fuente: Mejia, D. 2024

Realizado por: Mejia, D. 2024

En la Tabla 3-9, se detalla al Usuario- Técnico junto con sus roles correspondientes, cada uno asignado con funcionalidades específicas dentro del sistema.

Tabla 3-9: Usuario-Técnico.

Actor	Técnico
Rol	Empleado de la empresa específicamente del departamento de Soporte (TI)
descripción	Este es el actor específico que asume el rol de “Técnico de Soporte” en el sistema. Cada técnico que interactúa con la aplicación para revisar y responder a las solicitudes de asistencia representa este actor. Los técnicos se encargan de atender las solicitudes asignadas y proporcionar soluciones o respuestas apropiadas.
Requerimientos asociados	<ul style="list-style-type: none"> ● RF01 ● RF02 ● RF03 ● RF04 ● RF05

Fuente: Mejia, D. 2024

Realizado por: Mejia, D. 2024

3.6.1.3 Diagramas de casos de uso

En la Ilustración 3-2: Casos de uso - Usuarios, se presenta de manera gráfica la representación visual de los diversos casos de uso que han sido identificados para los usuarios en el contexto del sistema Daquinet.

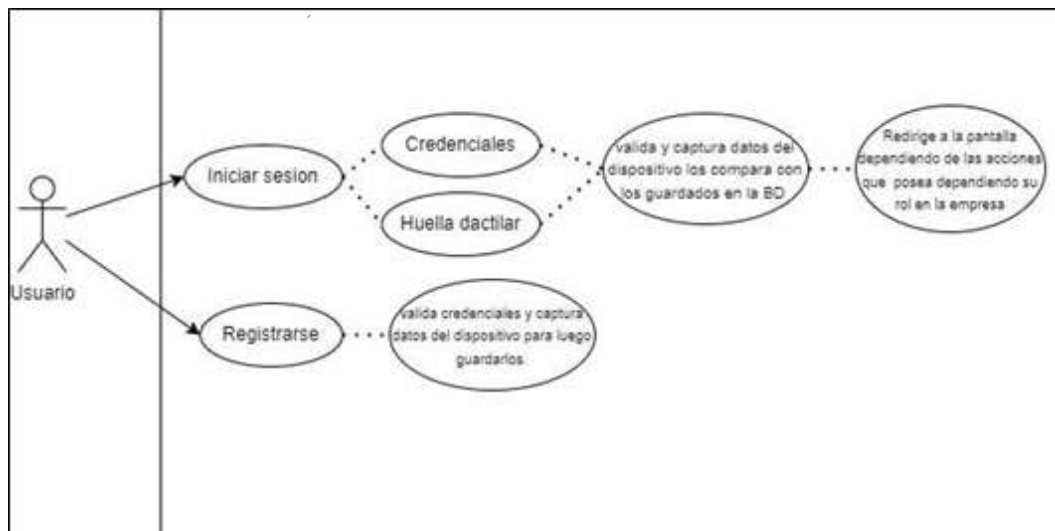


Ilustración 3-2: Casos de uso – usuarios.

Realizado por: Mejia, D. 2024

La Ilustración 3-3: Casos de Uso – Técnico proporciona una visión gráfica de los distintos casos de uso identificados para los técnicos dentro del sistema Daquinet.

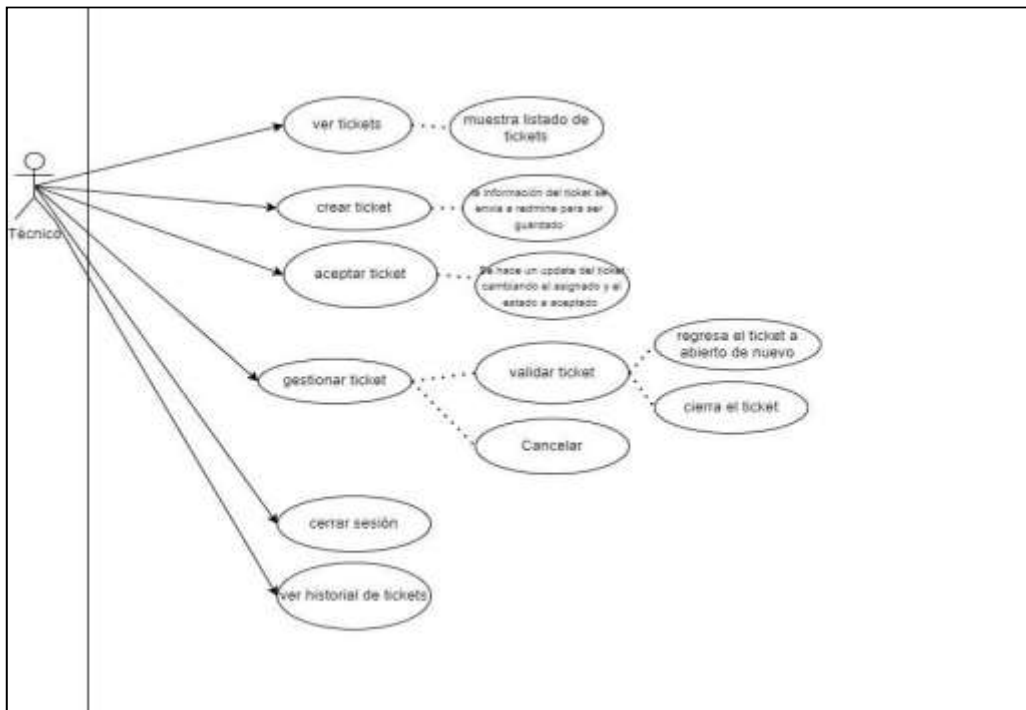


Ilustración 1-3: Casos de uso – técnico.

Realizado por: Mejia, D. 2024

3.6.1.4 Requerimientos

La sección de requisitos juega un papel fundamental en el proceso de desarrollo, ya que establece las funcionalidades esenciales y los criterios no funcionales que guiarán la implementación del sistema. En este contexto, se han identificado tanto requisitos funcionales como no funcionales para el diseño y desarrollo de la aplicación, que sigue la metodología ágil SCRUM. A continuación, se detallan los principales requisitos que abarcan aspectos cruciales, desde la autenticación hasta la seguridad de los datos, para garantizar un despliegue exitoso y una experiencia robusta para los usuarios.

La Tabla 3-10 hasta 3-14 presenta de forma organizada los requisitos funcionales esenciales para el sistema Daquinet, proporcionando una visión detallada y clara para el desarrollo de la aplicación.

Tabla 3-10: Requisito funcional 1.

ID	RF01
-----------	-------------

Nombre	Inicio de sesión
Prioridad	Alta
Descripción	Permite a los usuarios autenticarse en el sistema proporcionando sus credenciales para acceder a funcionalidades restringidas.
Actores	Usuarios - Cliente - Técnico
Dependencias	Ninguna
Requerimientos No funcionales	RNF01.
Comentarios	Se debe brindar las funcionalidad y permisos según el rol que se tenga el usuario.

Fuente: Mejia, D. 2024

Realizado por: Mejia, D. 2024

Tabla 3-11: Requisito funcional 2.

ID	RF02
Nombre	Registro de dispositivo
Prioridad	Alta
Descripción	Permite a los usuarios registrar su dispositivo, proporcionando la información de su dispositivo móvil para registrarse.
Actores	- Técnico - Cliente
Dependencias	Ninguna.
Requerimientos No funcionales	RNF01.
Comentarios	Para crear una cuenta, primeramente, se debe validar las credenciales mismas que deben constar en el registro (BD) de la cooperativa. El rol y la prioridad y más detalles se determinará según el tipo de cargo que este ocupe dentro de la cooperativa.

Fuente: Mejia, D. 2024

Realizado por: Mejia, D. 2024

Tabla 3-12: Requisito funcional 3.

ID	RF03
Nombre	Gestionar ticket
Prioridad	Alta
Descripción	Permite a los usuarios autenticados gestionar tickets.
Actores	- Cliente - Técnico
Dependencias	RF01 (para realizar esta acción es necesario que el usuario este logeado)
Requerimientos No funcionales	RNF01.

Comentarios	<p>El sistema debe permitir el cambio de estado según los estados disponibles:</p> <ul style="list-style-type: none"> - Nueva - En curso - Finalizada con éxito - Finalizado sin éxito. <p>Dentro de las operaciones disponibles se encuentran:</p> <ul style="list-style-type: none"> - Crear ticket - Modificar ticket - Ver ticket
--------------------	--

Fuente: Mejia, D. 2024

Realizado por: Mejia, D. 2024

Tabla 3-13: Requisito funcional 4.

ID	RF04
Nombre	Ver perfil
Prioridad	Media
Descripción	Permite a los usuarios ver el perfil, misma que debe constar con su información básica.
Actores	<ul style="list-style-type: none"> - Técnico - Cliente
Dependencias	RF01 (para realizar esta acción es necesario que el usuario este logeado)
Requerimientos No funcionales	RNF01.
Comentarios	Ninguna.

Fuente: Mejia, D. 2024

Realizado por: Mejia, D. 2024

Tabla 3-14: Requisito funcional 5.

ID	RF05
Nombre	Cerrar sesión
Prioridad	Alta
Descripción	Permite a los usuarios cerrar su sesión actual, desconectándose del sistema y finalizando su acceso a las funcionalidades restringidas.
Actores	<ul style="list-style-type: none"> Técnico Cliente Administrador
Dependencias	RF01 (para realizar esta acción es necesario que el usuario este logeado)
Requerimientos No funcionales	RNF01.
Comentarios	Ninguna.

Fuente: Mejia, D. 2024

Realizado por: Mejia, D. 2024

La Tabla 3-15 hasta 3-17 ofrece una estructura organizada que destaca las extensiones de requisitos para el sistema Daquinet. Estas tablas proporcionan una visión clara y detallada para orientar el desarrollo de la aplicación.

Tabla 3-15: Extensión 1.

SUBPROCESO	
Id de Requerimiento Padre	RF03
Id de Subproceso	SP01
Nombre	Aceptar ticket (Gestionar ticket)
Prioridad	Alta
Descripción	Permite a un técnico aceptar una solicitud de servicio (ticket) que se le ha sido asignado.
Actores	- Técnico
Comentarios	Es una extensión del requerimiento de GESTIONAR TICKET

Fuente: Mejia, D. 2024

Realizado por: Mejia, D. 2024

Tabla 3-16: Extensión 2.

SUBPROCESO	
Id de Requerimiento Padre	RF03
Id de Subproceso	SP02
Nombre	Modificar tickets (Gestionar ticket)
Prioridad	Alta
Descripción	Permite a un técnico modificar el estado de un ticket de acuerdo los diferentes estados que se manejen.
Actores	- Técnico - Cliente
Comentarios	Es una extensión del requerimiento de GESTIONAR TICKET

Fuente: Mejia, D. 2024

Realizado por: Mejia, D. 2024

Tabla 3-17: Extensión 3.

SUBPROCESO	
Id de Requerimiento Padre	RF03
Id de Subproceso	SP03
Nombre	Cerrar ticket (Gestionar ticket)

Prioridad	Alta
Descripción	Permite a un cliente que haya solicitado asistencia cerrar el proceso de atención donde: <ul style="list-style-type: none"> 1. Únicamente puede modificar el estado a: <ul style="list-style-type: none"> - Finalizar con éxito - Finalizar sin éxito
Actores	- Cliente
Comentarios	Es una extensión del requerimiento de gestionar ticket

Fuente: Mejia, D. 2024

Realizado por: Mejia, D. 2024

La Tabla 3-18 ofrece una estructura organizada que destaca de los requisitos no funcionales para el sistema Daquinet. Estas tablas proporcionan una visión clara y detallada para orientar el desarrollo de la aplicación.

Tabla 3-18: Requisito no funcional 2.

ID	RNF01
Categoría	Seguridad
Prioridad	Alta
Descripción	El sistema debe garantizar la confidencialidad y protección de los datos de los usuarios almacenados en la base de datos.
Objetivo	Asegurar que los datos de los usuarios no sean accesibles ni manipulables por personas no autorizadas.
Dependencias	Requisitos funcionales *
Restricciones	- El sistema debe brindar acceso únicamente a los usuarios debidamente autenticados.
Comentarios	- Incluir JWT para el manejo de tokens

Fuente: Mejia, D. 2024

Realizado por: Mejia, D. 2024

3.6.1.5 Product Backlog

El Product Backlog, una herramienta clave en metodologías ágiles como Scrum, es un cuadro dinámico que destila las necesidades y características del producto. La tabla 3-19 se enfoca en las historias de usuario para representar las funcionalidades desde la perspectiva de los usuarios-clientes y en la tabla 3-20 de los usuarios-técnicos. La asignación de puntos de historia es una práctica común en las metodologías ágiles, como Scrum. A menudo se utiliza la escala de Fibonacci (por ejemplo, 1, 2, 3, 5, 8, 13, 21) para expresar la incertidumbre inherente en la estimación y para evitar la precisión excesiva.

Tabla 3-19: Usuario-Cliente.

ID	Historia de usuario	Puntos estimados	Prioridad	descripción
HC-001	Iniciar Sesión como Cliente	5	Alta	Validación de credenciales y asignación de roles.
HC-002	Gestionar Tickets como Cliente	13	Alta	Crear, modificar y cerrar tickets con diferentes estados.
HC-003	Ver Perfil como Cliente	5	Media	Acceso al perfil después de iniciar sesión.
HC-004	Cerrar Sesión como Cliente	3	Alta	Finalización segura de la sesión actual.
HC-005	Registrar Dispositivo como Cliente	8	Alta	Se registra los datos del dispositivo.
HC-006	Consultar Estado de Tickets	5	Media	Visualizar el estado actual de los tickets creados.
ID	Historia de usuario	Puntos estimados	Prioridad	descripción
HC-007	Filtrar Tickets por Estado	8	Alta	Aplicar filtros para ver tickets según su estado
HC-008	Evaluar Experiencia de Usuario	8	Media	Proporcionar retroalimentación sobre la experiencia del usuario al cerrar un ticket.
HC-009	Acceder a Historial de Tickets	5	Media	Visualizar un historial de tickets anteriores y sus estados.

Fuente: Mejia, D. 2024

Realizado por: Mejia, D. 2024

Tabla 3-20: Usuario-Técnico.

ID	Historia de usuario	Puntos estimados	Prioridad	descripción
HT-001	Iniciar Sesión como Técnico	5	Alta	Validación de credenciales y asignación de roles.
HT-002	Gestionar Tickets como Técnico	13	Alta	Ver, aceptar y cerrar tickets asignados.
HT-003	Enviar a Revisión como Técnico	8	Alta	Enviar un ticket resuelto a

				revisión del cliente.
HT-004	Cerrar Ticket como Técnico	5	Alta	Cierre de tickets con éxito o sin éxito.
HT-005	Asignar Prioridades a Tickets	8	Alta	Asignar prioridades a los tickets dependiendo del rol que tenga el cliente en la cooperativa.
HT-006	Modificar Estado de Tickets Asignados	5	Media	Cambiar el estado de los tickets asignados
HT-007	Visualizar Historial de Tickets Asignados	5	Baja	Ver el historial de tickets asignados y sus estados.

Fuente: Mejia, D. 2024

Realizado por: Mejia, D. 2024

3.6.2 Fase de diseño

En la fase de diseño, el equipo se embarcó en actividades fundamentales de modelado y diagramación para esbozar con precisión la arquitectura del sistema. La ilustración 3-4 es una ilustración referencial a la arquitectura empleada.

3.6.2.1 Diseño de la arquitectura Backend

El sistema móvil es una API REST desarrollada con Spring Boot 3 que implementa una arquitectura hexagonal. En los siguientes puntos se detallarán los aspectos más importantes que reflejan la aplicación de la arquitectura en este proyecto entre otros aspectos.

Arquitectura

Para el presente proyecto se ha contemplado el uso de la arquitectura MVC, lo cual nos permite desacoplar entre la lógica de negocio central y los detalles de infraestructura como el diseño de la interfaz de usuario. Esta arquitectura promueve desacoplar el dominio de negocio del código de infraestructura. Las entradas y salidas (inputs y outputs) estarán a cargo de objetos de transporte de datos (DTO) siendo este el único medio de comunicación externa.

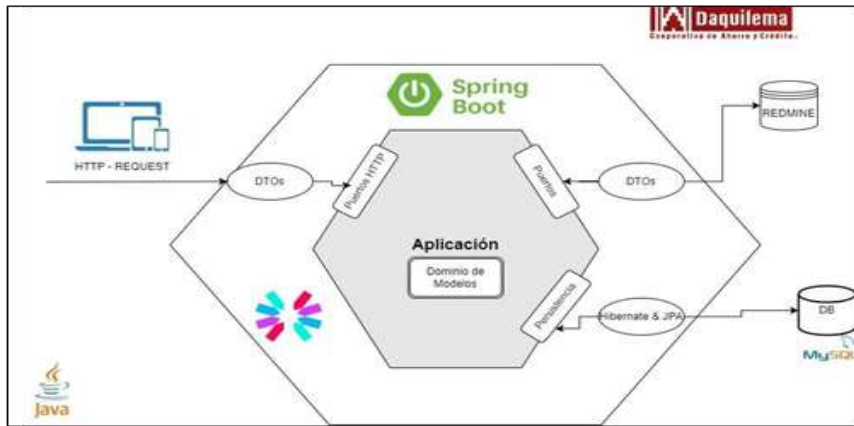


Ilustración 3-4: Arquitectura.
 Realizado por: Mejia, D. 2024

Organización de carpetas

El sistema móvil permite crear tareas en Redmine de asistencia técnica desde mi cliente usando a mi API de SPRING BOOT como puente.

La organización carpetas se la puede visualizar en la ilustración 3-5.

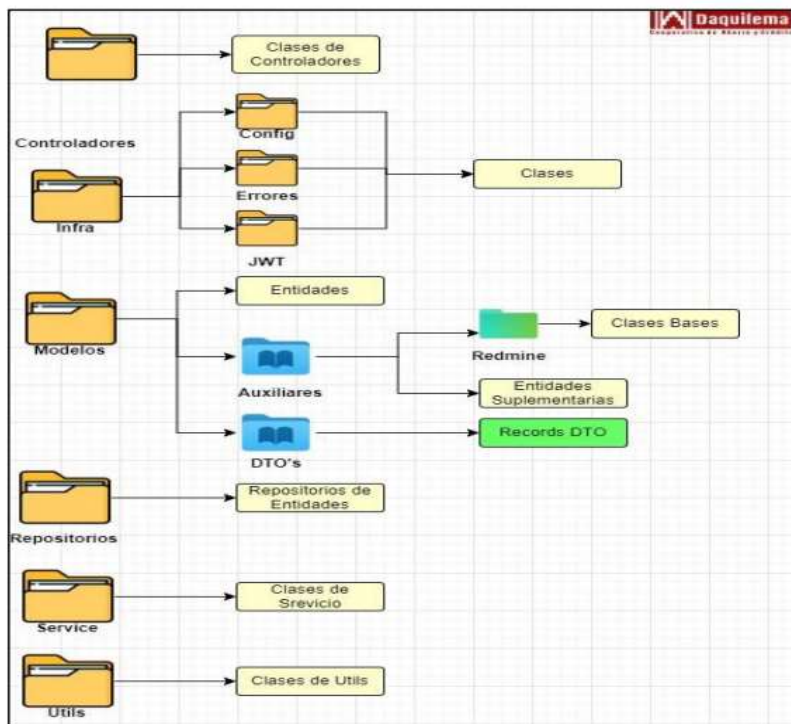


Ilustración 3-5: Organización de carpetas backend.
 Realizado por: Mejia, D. 2024

3.6.2.2 *Diseño de la arquitectura*

Se utiliza una arquitectura de tres capas que proporciona una estructura organizada y modular para el desarrollo de aplicaciones, facilitando la mantenibilidad, escalabilidad y la reutilización de código. Es una elección común en el desarrollo de software para separar las responsabilidades y crear sistemas más robustos y flexibles.

- Capa de presentación (UI): Desarrollada con widgets de Flutter, contiene toda la interfaz de usuario y los elementos visuales que interactúan con el usuario.
- Capa de lógica de negocio: Concentra los casos de uso referentes al flujo de tickets y las reglas de negocio. Independiente de la UI.
- Capa de datos: Proporciona una interfaz común para acceder al repositorio de datos, aislando la lógica de negocio de cómo se obtienen o persisten los datos interna o externamente.
- Otros: Proporciona funciones, métodos reutilizables según el contexto.

Organización de carpetas

En la ilustración 3-6 se puede visualizar la estructura de las carpetas del frontend.

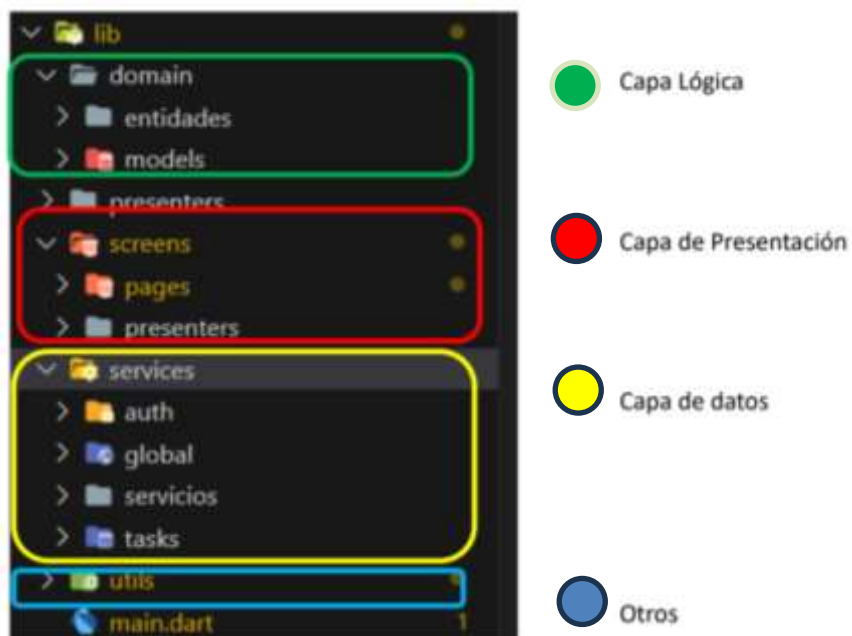


Ilustración 3-6: Organización de carpetas.
Realizado por: Mejia, D. 2024

3.6.2.3 Diseño de la base de datos

Diagrama entidad-relación

Entidades: TUSUARIO, TDISPOSITIVO, TDISPOSITIVOHIST, TPERMISOS

Relaciones:

- TDISPOSITIVO tiene FK a TUSUARIO
- TDISPOSITIVOHIST no tiene FK
- TUSUARIOHIST no tiene FK
- TPERMISOS no tiene FK

La ilustración 3-7 muestra el diagrama entidad-relación.

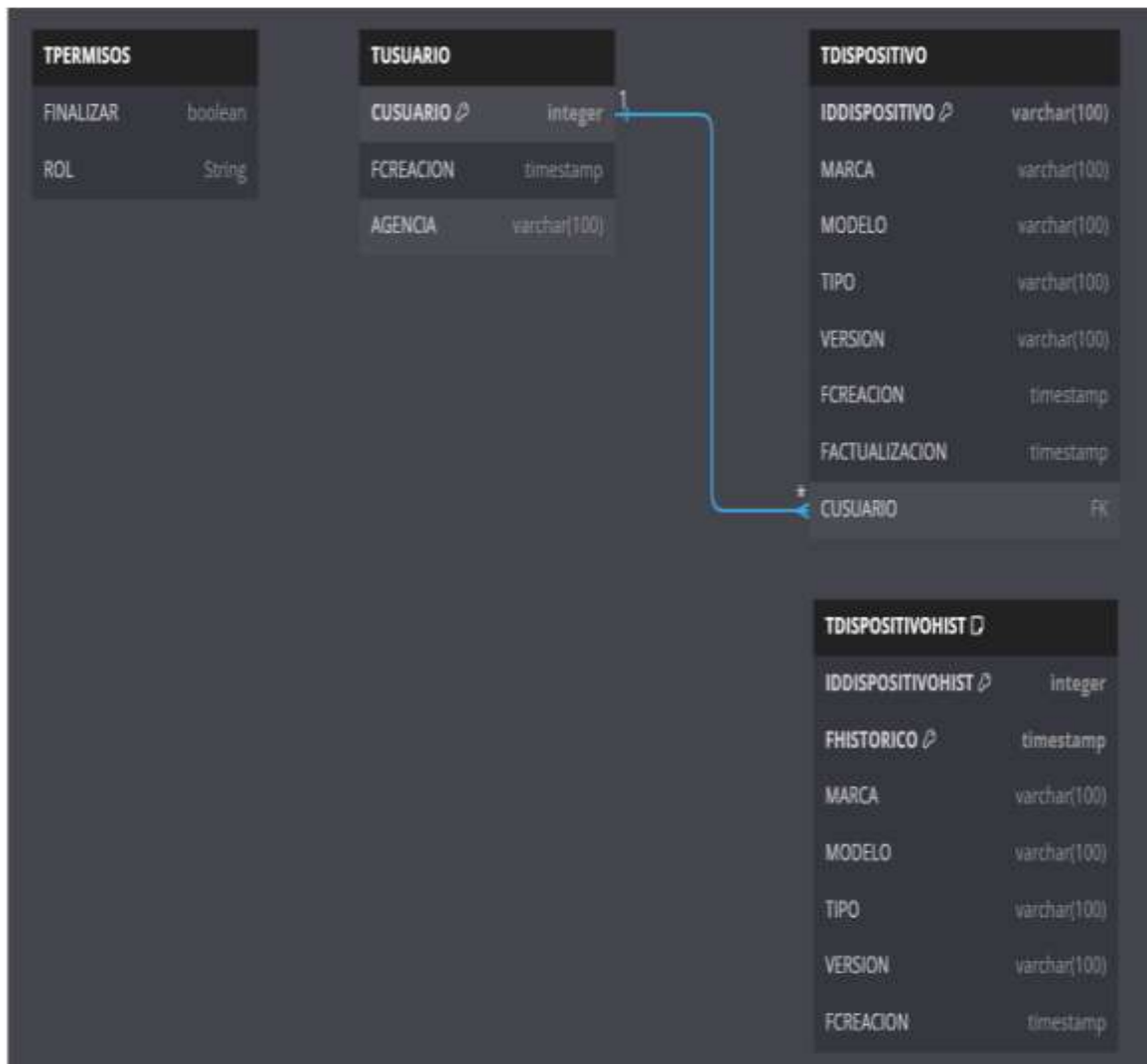


Ilustración 3-7: Diagrama entidad-relación.

Realizado por: Mejia, D. 2024

3.6.2.4 Diccionario de datos

TPERMISOS: Tabla de estados

- FINALIZAR: Esta columna almacena valores booleanos que indican si los roles tienen permiso para finalizar ciertas acciones o tareas. Puede ser verdadero (true) o falso (false).
- ROL: Almacena el nombre o identificador del rol al que se le asigna o retira el permiso de finalización.

TUSUARIO: Tabla de usuarios

- CUSUARIO: llave primaria, identificación única del Usuario
- FCREACION: fecha creación registro

TDISPOSITIVO: Tabla de dispositivos

- IDDISPOSITIVO: llave primaria
- MARCA: marca del dispositivo
- MODELO: modelo del dispositivo
- TIPO: tipo de dispositivo
- VERSION: versión del dispositivo
- FCREACION: fecha creación registro
- FACTUALIZACION: fecha actualización registro
- CUSUARIO: FK a TUSUARIO

TDISPOSITIVOHIST: Tabla histórica de dispositivos

- IDDISPOSITIVOHIST: llave primaria
- FHISTORICO: fecha del registro histórico, llave primaria
- MARCA: marca del dispositivo
- MODELO: modelo del dispositivo
- TIPO: tipo de dispositivo
- VERSION: versión del dispositivo

- FCREACION: fecha creación registro

3.6.2.5 *Diseño de interfaces de usuario*

En el ámbito del desarrollo, el diseño de interfaces de usuario (UI) juega un papel crucial al actuar como el puente visual entre los usuarios y el sistema. Este apartado se enfoca en la creación de interfaces atractivas y funcionales que cumplen con los requisitos previamente identificados.

En la ilustración 3-8, se observa el diseño de la pantalla de inicio de sesión del sistema móvil donde cuenta con la opción de usuario y contraseña donde permitirá ingresar las credenciales de la cooperativa para poder registrar el dispositivo y la opción de Huella donde una vez registrado el dispositivo se podrá ingresar mediante la huella dactilar por medio de cada dispositivo.



Ilustración 3-8: Pantalla de inicio de sesión.

Realizado por: Mejia, D. 2024

En la ilustración 3-9 se puede observar la pantalla de creación de un ticket donde tendrá que ingresar datos básicos respecto a la incidencia a resolver.



Ilustración 3-9: Pantalla de creación de un ticket.

Realizado por: Mejia, D. 2024

El resto de los diseños de la interfaz de usuario se encuentran en el Anexo C.

3.6.3 Fase de desarrollo

La fase de desarrollo marca el corazón del proyecto, donde la planificación y el diseño se materializan en un producto funcional. Este periodo implica la implementación meticulosa de los requisitos identificados, con un enfoque en la eficiencia y calidad del código.

3.6.3.1 Spring Backlog

Para concretar el desarrollo de la aplicación móvil, se ha delineado un plan que comprende la ejecución de 4 Sprints. Cada Sprint cuenta con una cantidad de puntos estimados, considerando que en nuestro contexto un punto equivale entre una y dos horas de trabajo asegurando eficiencia y calidad en cada etapa del proceso.

En la Tabla 3-21 se visualiza el sprint backlog.

Tabla 3-21: Sprint Backlog.

Sprint	ID	Historia de usuario	Puntos estimados	Fecha Inicio	Fecha Fin	Puntos totales
1	HC-001	Iniciar Sesión como Cliente	7	30/11/2023	02/12/2023	35
	HT-001	Iniciar Sesión como Técnico	7	30/11/2023	02/12/2023	
	HC-005	Registrar Dispositivo como Cliente	14	03/12/2023	06/12/2023	
	HT-005	Asignar Prioridades a Tickets	7	07/12/2023	08/12/2023	
2	HC-002	Gestionar Tickets como Cliente	14	09/12/2023	11/12/2023	49
	HT-002	Gestionar Tickets como Técnico	14	09/12/2023	11/12/2023	
	HT-004	Cerrar Ticket como Técnico	7	12/12/2023	15/12/2023	
	HC-007	Filtrar Tickets por Estado	7	16/12/2023	18/12/2023	
	HC-006	Consultar Estado de Tickets	7	16/12/2023	18/12/2023	
3	HC-003	Ver Perfil como Cliente	7	19/12/2023	21/12/2023	35
	HT-003	Enviar a Revisión como Técnico	14	22/12/2023	25/12/2023	
	HC-008	Evaluar Experiencia de Usuario	14	26/12/2023	28/12/2023	
4	HC-009	Acceder a Historial de Tickets	7	29/12/2023	31/12/2023	21
	HT-006	Modificar Estado de Tickets Asignados	7	01/01/2024	03/01/2024	
	HT-007	Visualizar Historial de Tickets Asignados	7	04/01/2024	07/01/2024	

Fuente: Mejia, D. 2024

Realizado por: Mejia, D. 2024

3.6.3.2 Historias de usuario

En la Tabla 3-22 se detalla la historia de usuario de inicio de sesión como usuario de tipo cliente.

Tabla 3-22: Historia de usuario HC-001.

Identificación de la Historia de Usuario:	HC-001
Nombre del Requerimiento :	Iniciar Sesión como Cliente
Usuarios:	Clientes
Descripción:	Permitir a los clientes iniciar sesión en el sistema.
Iteración Asignada:	Sprint 1
	Prioridad del requerimiento: Alta

Fuente: Mejia, D. 2024

Realizado por: Mejia, D. 2024

En el Anexo B, se presenta una detallada descripción de las historias de usuario, cada una identificada por un nombre específico. Cada historia está dirigida a un grupo específico de usuarios, ya sea clientes o técnicos, y se proporciona una breve, pero completa descripción de la funcionalidad que se busca implementar. Además, se asigna a cada historia de usuario a una iteración específica, lo que facilita su seguimiento y planificación dentro del proceso de desarrollo. La prioridad de cada historia también se ha definido, brindando una guía clara sobre la importancia relativa de cada requerimiento en el contexto del proyecto. Este conjunto detallado de información en el Anexo B es esencial para comprender y gestionar eficientemente las historias de usuario a lo largo del ciclo de vida del proyecto.

CAPÍTULO IV

4 ANÁLISIS E INTERPRETACIÓN DE RESULTADOS

Los datos presentados se basan en un mes de operaciones, periodo durante el cual se registraron 82 tickets generados por 34 usuarios en las sucursales de Riobamba de la Cooperativa de Ahorro y Crédito Fernando Daquilema. Esta selección temporal se justifica por la necesidad de abordar las dinámicas operativas y de interacción de los usuarios con el sistema Daquinet en condiciones de uso real. Es fundamental considerar que no todos los usuarios enfrentan problemas de incidencias de manera constante, y la muestra refleja la variabilidad en la frecuencia y naturaleza de las interacciones con el sistema durante el periodo de observación.

4.1 Formulación de hipótesis

4.1.1 *Formulación de hipótesis nula*

H0: No hay diferencia significativa en el tiempo medio requerido entre el sistema de gestión de incidencias Daquinet y el método manual, en términos de eficiencia.

4.1.2 *Formulación de hipótesis alterna*

H1: Existe diferencia significativa en el tiempo medio requerido entre el sistema de gestión de incidencias Daquinet y el método manual, en términos de eficiencia.

4.2 Medición de la eficiencia

4.2.1 *Encuesta*

Las preguntas formuladas en la encuesta han sido diseñadas cuidadosamente tomando como referencia la sub-característica de eficiencia definida en el estándar ISO/IEC 25010, aspecto crucial para medir la eficiencia en la gestión de incidencias.

Las preguntas están divididas en dos bloques distintos, uno destinado a evaluar la experiencia del usuario con el sistema anterior y el otro enfocado al sistema nuevo Daquinet. Cada bloque contiene preguntas diseñadas para medir la percepción de velocidad y eficiencia por parte del usuario centrándose en aspectos como rapidez de respuesta. Además, se incluyen dos preguntas

que permiten a los participantes comparar directamente la velocidad percibida entre ambos sistemas sin entrar en detalles técnicos.

Las preguntas de la encuesta se las encuentra en el Anexo C:

4.2.2 *Análisis de tendencias*

En la Tabla 23-4 se muestra detalladamente la asignación de valores a cada respuesta de las preguntas, cave recalcar que las preguntas 1,2 no se tomaran en cuenta para la asignación de valores puesto que es información demográfica, para la pregunta 3-6 se utiliza una escala de Likert, donde el valor máximo es 4 y valor mínimo 1 y para las preguntas 4-7 y 5-8 se utiliza una escala de rangos, donde el valor máximo es 3 y valor mínimo 1 y finalmente las preguntas 9 y 10 con una escala de Likert con un valor máximo de 3 y mínimo de 1. Se asignan estos valores por uniformidad y simplicidad de la interpretación de los datos obtenidos, el detalle se muestra en la Tabla 4-1.

Tabla 4-1: Tabla de asignación de valores a las respuestas de las preguntas.

Preguntas	Respuestas	Valores
Preguntas 3 y 6	Muy rápida	4
	Rápida	3
	Moderada	2
	Lenta	1
Preguntas 4 y 7	Menos de 3 minutos	3
	3-5 minutos	2
	más de 5 minutos	1
Preguntas 5 y 8	Menos de 15 minutos	4
	15-30 minutos	3
	30 minutos - 1 hora	2
	Más de 1 hora	1
Preguntas 9 y 10	Sí, ha mejorado significativamente	3

	Sí, ha mejorado moderadamente	2
	No ha mejorado, ha empeorado	1

Fuente: Mejía, D. 2024

Realizado por: Mejía, D. 2024

Para realizar el análisis de tendencias, primero, es necesario tener los datos organizados de manera clara. A continuación, en la Tabla 4-2, se presenta los datos de las respuestas de la encuesta en un formato tabular para facilitar el análisis.

Tabla 4-2: Tabulación de resultados.

Encuesta do	Pregunta 3	Pregunta 4	Pregunta 5	Pregunta 6	Pregunta 7	Pregunta 8	Pregunta 9	Pregunta 10
1	4	3	4	2	1	2	3	3
2	4	3	4	3	1	1	3	3
3	4	2	4	2	1	2	3	3
4	3	3	4	1	1	1	3	3
5	4	3	3	2	1	1	2	2
6	4	3	3	2	1	1	3	3
7	3	3	4	2	1	2	2	3
8	3	2	4	2	1	2	3	3
9	4	3	4	2	1	1	3	3
10	4	3	3	2	1	2	2	2
11	4	3	3	2	2	1	3	3
12	3	2	4	1	3	1	3	2
13	4	3	4	2	3	2	2	3
14	4	3	3	1	3	1	3	3
15	4	3	4	2	3	2	2	2
16	3	2	4	2	3	2	3	2
17	3	2	4	2	3	2	3	3
18	4	2	4	1	3	1	2	2
19	3	2	3	1	3	1	2	2
20	3	3	3	2	3	2	2	2
21	3	2	3	2	3	1	2	3
22	4	3	3	2	3	1	3	3
23	4	3	4	2	3	2	2	2
24	4	3	4	2	3	2	3	2
25	3	2	4	1	3	1	3	3
26	4	2	3	2	3	2	2	3

27	4	2	3	2	3	1	3	3
28	4	3	4	2	3	2	3	2
29	4	3	4	1	3	1	3	3
30	4	3	3	1	3	2	3	3
31	3	2	3	1	3	2	2	3
32	4	2	4	2	3	1	3	3
Promedio	3,66	2,59	3,59	1,75	2,34	1,50	2,63	2,66

Fuente: Mejia, D. 2024

Realizado por: Mejia, D. 2024

Sistema nuevo Daquinet

Pregunta 3 - Velocidad de Respuesta del Sistema Nuevo al Abrir, Cerrar o Actualizar Tickets:

El promedio obtenido es de 3.66. Esto indica que la mayoría de los encuestados calificaron la velocidad de respuesta del sistema nuevo como "Muy Rápida" o cercana a esa categoría.

Pregunta 4 - Tiempo Promedio de Creación de Tickets con el Sistema Nuevo:

El promedio es de 2.59. Esto sugiere que, en promedio, los encuestados tardaban entre "Menos de 3 minutos" y "3-5 minutos" en crear un ticket utilizando el sistema nuevo.

Pregunta 5 - Tiempo Promedio de Aceptación de Tickets por un Técnico con el Sistema Nuevo:

El promedio es de 3.59. Esto indica que la mayoría de los encuestados experimentaban una aceptación rápida de sus tickets por parte de los técnicos utilizando el sistema nuevo.

Sistema anterior

Pregunta 6 - Velocidad de Respuesta del Sistema Anterior al Abrir, Cerrar o Actualizar Tickets:

El promedio es de 1.75. Esto sugiere que la mayoría de los encuestados calificaron la velocidad de respuesta del sistema anterior como "Lenta" o cercana a esa categoría.

Pregunta 7 - Tiempo Promedio de Creación de Tickets con el Sistema Anterior:

El promedio es de 2.34. Esto indica que, en promedio, los encuestados tardaban entre "Menos de 3 minutos" y "15-30 minutos" en crear un ticket utilizando el sistema anterior.

Pregunta 8 - Tiempo Promedio de Aceptación de Tickets por un Técnico con el Sistema Anterior:

El promedio es de 1.50. Esto sugiere que la mayoría de los encuestados experimentaban una aceptación rápida de sus tickets por parte de los técnicos utilizando el sistema anterior.

Pregunta 9 - Mejora en la Eficiencia Percibida con el Sistema Nuevo:

El promedio es de 2.63. Esto indica que la mayoría de los encuestados percibieron una mejora moderada a significativa en la eficiencia al utilizar el sistema nuevo en comparación con el anterior.

Pregunta 10 - Mejora en el Tiempo de Atención del Ticket con el Sistema Nuevo:

El promedio es de 2.66. Esto sugiere que la mayoría de los encuestados percibieron una mejora moderada a significativa en el tiempo de atención de sus tickets al utilizar el sistema nuevo en comparación con el anterior.

En general, se observa una tendencia hacia una mejor eficiencia y tiempos de respuesta más rápidos con el sistema nuevo en comparación con el sistema anterior.

La mayoría de los encuestados percibieron una mejora moderada a significativa en la eficiencia y el tiempo de atención de los tickets con el sistema nuevo.

Para completar este análisis se realizará un proceso estadístico para pares relacionados.

4.2.3 Prueba de normalidad de Shapiro Wilk

Para elegir un proceso estadístico adecuado iniciamos con el test de normalidad de Shapiro-Wilk y determinaremos si la distribución de los datos es Normal o no. En caso de que el valor p resultante sea inferior al coeficiente de referencia obtenido mediante el nivel de significancia predefinido (0.05), se procederá a rechazar la hipótesis nula que establece que los datos siguen

una distribución normal. Los resultados específicos de esta prueba se encuentran detallados en la Tabla 4-3, proporcionando así los cálculos pertinentes obtenidos durante el análisis.

Shapiro Wilk test de las respuestas de la pregunta 3

Población: 32

H0: $X_i \sim N(\mu, \sigma^2)$

H1: $X_i \not\sim N(\mu, \sigma^2)$

Nivel de significancia: 0,05

Tabla 4-3: Cálculos prueba Shapiro Wilk

i	X_i	$(X_i - \text{MED})^2$	A_i	$X_i \text{ INV}$	Dif ($X_i - X_i \text{ INV}$)
1	3	0,43066406	0,4188	4	-1
2	3	0,43066406	0,2898	4	-1
3	3	0,43066406	0,2463	4	-1
4	3	0,43066406	0,2141	4	-1
5	3	0,43066406	0,1878	4	-1
6	3	0,43066406	0,1651	4	-1
7	3	0,43066406	0,1449	4	-1
8	3	0,43066406	0,1265	4	-1
9	3	0,43066406	0,1093	4	-1
10	3	0,43066406	0,0931	4	-1
11	3	0,43066406	0,0777	4	-1
12	4	0,11816406	0,0629	4	0
13	4	0,11816406	0,0485	4	0
14	4	0,11816406	0,0344	4	0
15	4	0,11816406	0,0206	4	0
16	4	0,11816406	0,0068	4	0
17	4	0,11816406		4	
18	4	0,11816406		4	
19	4	0,11816406		4	
20	4	0,11816406		4	
21	4	0,11816406		4	
22	4	0,11816406		3	
23	4	0,11816406		3	
24	4	0,11816406		3	

25	4	0,11816406		3	
26	4	0,11816406		3	
27	4	0,11816406		3	
28	4	0,11816406		3	
29	4	0,11816406		3	
30	4	0,11816406		3	
31	4	0,11816406		3	
32	4	0,11816406		3	

Fuente: Mejia, D. 2024

Realizado por: Mejia, D. 2024

Una vez obtenidos los cálculos en la Tabla 25-4 debemos obtener la correlación (P) de Shapiro Wilk que se obtiene de la Suma-Producto $(A_i * Dif)^2 / \sum (X_i - MED)^2$ que posteriormente se comparara con la correlación mínima necesario que nos brinda Shapiro Wilk para el nivel de significancia de 0.05 y una población de 32.

Los resultados se detallan en la Tabla 4-4.

Tabla 4-4: Resultados prueba Shapiro Wilk

MEDIA Xi	3,65625
$\sum (X_i - MED)^2$	7,21875
SUMA-PRODUCTO $(A_i * Dif)$	-2,0734
SHAPIRO WILK correlación (P)	0,59553074
SHAPIRO WILK porcentaje mínimo	0,930

Fuente: Mejia, D. 2024

Realizado por: Mejia, D. 2024

Una vez obtenido la correlación $P = 0.59553074$ y tomando en cuenta que el porcentaje mínimo que nos indica Shapiro Wilk con nuestros datos que es de 0.930 podemos rechazar la hipótesis nula y aceptar la hipótesis alternativa que nos indica que los valores obtenidos en la encuesta **NO** pertenecen a una distribución normal.

El análisis de la pregunta 4 hasta la pregunta 10 se encuentran en el Anexo F.

Luego de realizar un análisis exhaustivo de todos los datos, se concluye que estos no siguen una distribución normal. Con esta información al **NO** ser datos normales, se empleará la prueba de Wilcoxon para pares de datos relacionados. Esta elección se basa en la naturaleza no paramétrica

de la prueba de Wilcoxon, la cual no asume una distribución normal de los datos y, por lo tanto, es apropiada para analizar comparaciones entre dos conjuntos de datos relacionados en estas condiciones.

4.2.4 Prueba de Wilcoxon

Se utilizará un nivel de significancia de 0.05 en la prueba de Wilcoxon para muestras relacionadas. Este nivel de significancia representa la probabilidad aceptada de cometer un error tipo I, es decir, la probabilidad de rechazar incorrectamente la hipótesis nula. Al fijar el nivel de significancia en 0.05, estamos dispuestos a aceptar un 5% de probabilidad de concluir erróneamente que hay una diferencia significativa cuando, de hecho, no la hay.

Este umbral es comúnmente aceptado en la investigación científica y proporciona un equilibrio entre la sensibilidad del análisis y el control del error tipo I.

Los valores críticos para la prueba de Wilcoxon fueron obtenidos de la tabla de distribución de Wilcoxon para un nivel de significancia $\alpha=0.05$ y un tamaño de muestra $n=34$.

4.2.4.1 Preguntas 3-6

Primero se toma las preguntas 3 y 6 y se crea una hipótesis nula y una alternativa.

Hipótesis Nula (H0): No hay diferencia significativa en la calificación de la velocidad de respuesta general entre el sistema nuevo "Daquinet" y el sistema anterior al abrir, cerrar o actualizar tickets.

Hipótesis Alternativa (H1): Existe una diferencia significativa en la calificación de la velocidad de respuesta general entre el sistema nuevo "Daquinet" y el sistema anterior al abrir, cerrar o actualizar tickets.

Enlistamos los datos de la pregunta 3 y 6 para luego emparejarlos para poder obtener las diferencias ordenadas y rangos de diferencias ordenadas obteniendo así la suma de rangos para diferencias positivas (W+).

Datos:

Pregunta 3 (Sistema Nuevo Daquinet): [4, 4, 4, 3, 4, 4, 3, 3, 4, 4, 4, 3, 4, 4, 3, 4, 3, 3, 4, 3, 4, 4, 4, 3, 4, 4, 4, 4, 3, 4].

Pregunta 6 (Sistema Anterior): [3, 3, 4, 2, 2, 3, 2, 2, 2, 2, 1, 2, 1, 2, 2, 2, 1, 3, 1, 2, 2, 2, 1, 2, 2, 2, 3, 3, 1, 2].

Diferencias:

Diferencias: [4 - 3, 4 - 3, 4 - 4, 3 - 2, 4 - 2, 4 - 3, 3 - 2, 3 - 2, 4 - 2, 4 - 2, 4 - 2, 3 - 1, 4 - 2, 4 - 1, 3 - 2, 4 - 2, 3 - 2, 3 - 1, 4 - 3, 3 - 1, 4 - 2, 4 - 2, 4 - 2, 3 - 1, 4 - 2, 4 - 2, 4 - 2, 4 - 3, 4 - 3, 3 - 1, 4 - 2].

Diferencias Ordenadas: [1, 1, 0, 1, 2, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 4, 4, 4, 4].

Rangos de Diferencias Ordenadas: [31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62].

Suma de Rangos para Diferencias Positivas (W^+): 232.

Estadístico de Wilcoxon:

Tamaño de la muestra (n): 34 (total de respuestas).

Suma de Rangos para Diferencias Positivas (W^+): 232.

Interpretación:

Se debe comparar la Suma de Rangos para Diferencias Positivas (W^+) con el valor crítico de Wilcoxon para determinar si hay una diferencia significativa en la velocidad de respuesta entre el sistema nuevo y el sistema anterior.

Para $n=34$ y $\alpha=0.05$, el valor crítico de W^+ es 199.

Con una Suma de Rangos para Diferencias Positivas (W^+) de 232, que supera significativamente el valor crítico de Wilcoxon de 199 para un nivel de significancia $\alpha=0.05$ y una muestra de tamaño $n=34$, se rechaza la hipótesis nula y se acepta la hipótesis alternativa. Por lo tanto, hay evidencia estadística para afirmar que existe una diferencia significativa en la calificación de la velocidad

de respuesta general entre el sistema nuevo 'Daquinet' y el sistema anterior al abrir, cerrar o actualizar tickets.

4.2.4.2 Preguntas 4-7

Al igual que con las preguntas 3-6, procedemos a formular una hipótesis nula y una alternativa para las preguntas 4 y 7, seguido de un análisis de Wilcoxon.

Hipótesis Nula (H0): No hay diferencia significativa en el tiempo promedio de creación de tickets entre el sistema nuevo "Daquinet" y el sistema anterior.

Hipótesis Alternativa (H1): Existe una diferencia significativa en el tiempo promedio de creación de tickets entre el sistema nuevo "Daquinet" y el sistema anterior.

Enlistamos los datos de las preguntas 4 y 7 para obtener las diferencias ordenadas y los rangos de diferencias ordenadas.

Datos:

Pregunta 4 (Sistema Nuevo Daquinet): [3, 3, 2, 3, 3, 3, 3, 2, 3, 3, 3, 2, 3, 3, 3, 2, 2, 3, 3, 3, 3, 3, 2, 3, 3, 2, 2, 3, 3, 2, 2, 3].

Pregunta 7 (Sistema Anterior): [2, 1, 2, 1, 2, 2, 2, 2, 2, 1, 3, 2, 2, 2, 1, 1, 2, 2, 2, 2, 2, 3, 2, 3, 1, 2, 2, 3, 2, 3].

Diferencias:

Diferencias: [3 - 2, 3 - 1, 2 - 2, 3 - 1, 3 - 2, 3 - 2, 3 - 2, 2 - 2, 3 - 2, 3 - 2, 3 - 1, 2 - 3, 3 - 2, 3 - 2, 3 - 2, 2 - 1, 2 - 1, 3 - 2, 3 - 2, 3 - 2, 3 - 2, 3 - 2, 2 - 2, 3 - 3, 3 - 2, 2 - 3, 2 - 1, 2 - 2, 3 - 3, 2 - 2, 2 - 3].

Diferencias Ordenadas:

[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3].

Rangos de Diferencias Ordenadas:

[30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61].

Suma de Rangos para Diferencias Positivas (W^+): 207.

Estadístico de Wilcoxon:

Tamaño de la muestra (n): 34 (total de respuestas).

Suma de Rangos para Diferencias Positivas (W^+): 207.

Interpretación:

Al comparar la Suma de Rangos para Diferencias Positivas (W^+) con el valor crítico de Wilcoxon (199 para $\alpha=0.05$), obtenemos un W^+ de 207, superando significativamente el valor crítico. Por lo tanto, se rechaza la hipótesis nula y se acepta la hipótesis alternativa. Se concluye que hay evidencia estadística para afirmar que existe una diferencia significativa en el tiempo promedio de creación de tickets entre el sistema nuevo "Daquinet" y el sistema anterior.

4.2.4.3 Preguntas 5-8

Para las preguntas 5 y 8, se plantea la hipótesis nula y alternativa.

Hipótesis Nula (H_0): No hay diferencia significativa en el tiempo promedio de creación de un ticket hasta su aceptación por un técnico entre el sistema nuevo "Daquinet" y el sistema anterior.

Hipótesis Alternativa (H_1): Existe una diferencia significativa en el tiempo promedio de creación de un ticket hasta su aceptación por un técnico entre el sistema nuevo "Daquinet" y el sistema anterior.

Se enlistan los datos de las preguntas 5 y 8 y se emparejan para obtener las diferencias ordenadas y los rangos de diferencias ordenadas, calculando así la Suma de Rangos para Diferencias Positivas (W^+).

Datos:

Pregunta 5 (Sistema Nuevo Daquinet): [4, 4, 4, 3, 3, 3, 4, 4, 4, 3, 3, 4, 4, 3, 3, 4, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 2, 2, 3, 3].

Pregunta 8 (Sistema Anterior): [3, 3, 3, 3, 3, 2, 2, 2, 3, 3, 2, 2, 2, 2, 2, 2, 2, 3, 3, 1, 3, 3, 3, 2, 3, 3, 3, 2, 3, 3, 3].

Diferencias:

Diferencias: [4 - 3, 4 - 3, 4 - 3, 3 - 3, 3 - 3, 3 - 2, 4 - 2, 4 - 2, 4 - 3, 3 - 3, 3 - 2, 4 - 2, 4 - 2, 3 - 2, 3 - 2, 4 - 2, 3 - 2, 3 - 3, 3 - 3, 3 - 1, 3 - 3, 3 - 3, 3 - 2, 3 - 3, 3 - 3, 3 - 3, 3 - 2, 3 - 3, 2 - 3, 2 - 3, 3 - 3].

Diferencias Ordenadas: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2].

Rangos de Diferencias Ordenadas: [22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53].

Suma de Rangos para Diferencias Positivas (W^+): 218.

Estadístico de Wilcoxon:

Tamaño de la muestra (n): 34 (total de respuestas).

Suma de Rangos para Diferencias Positivas (W^+): 218.

Interpretación:

Se compara la Suma de Rangos para Diferencias Positivas (W^+) con el valor crítico de Wilcoxon para determinar si hay una diferencia significativa en el tiempo promedio de creación de un ticket hasta su aceptación por un técnico entre el sistema nuevo y el sistema anterior.

Para $n=34$ y $\alpha=0.05$, el valor crítico de W^+ es 199. Con una Suma de Rangos para Diferencias Positivas (W^+) de 218, que supera significativamente el valor crítico de Wilcoxon de 199 para un nivel de significancia $\alpha=0.05$ y una muestra de tamaño $n=34$, se rechaza la hipótesis nula y se acepta la hipótesis alternativa. Por lo tanto, hay evidencia estadística para afirmar que existe una diferencia significativa en el tiempo promedio de creación de un ticket hasta su aceptación por un técnico entre el sistema nuevo 'Daquinet' y el sistema anterior.

La Tabla 4-5 muestra un resumen de los resultados obtenidos después de haber realizado la prueba de Wilcoxon.

Tabla 4-5: Resumen de resultados prueba de Wilcoxon.

Comparación	Tamaño de la muestra (n)	Suma de rangos para diferencias positivas (W+)	Valor crítico de Wilcoxon ($\alpha = 0.05$)	Interpretación
Preguntas 3-6	34	232	199	Existe una diferencia significativa en la calificación de la velocidad de respuesta general.
Preguntas 4-7	34	207	199	Existe una diferencia significativa en el tiempo promedio de creación de tickets.
Preguntas 5-8	34	218	199	Existe una diferencia significativa en el tiempo promedio de creación de un ticket hasta su aceptación.

Fuente: Mejia, D. 2024

Realizado por: Mejia, D. 2024

Basándonos en los resultados resumidos en la Tabla 25-4, podemos concluir que hay evidencia estadística para rechazar la hipótesis nula (H_0) de que no hay diferencia significativa en el tiempo medio requerido entre el sistema de gestión de incidencias Daquinet y el método manual en términos de eficiencia. En su lugar, se acepta la hipótesis alternativa (H_1) de que existe una diferencia significativa en el tiempo medio requerido entre ambos métodos. Esto sugiere que el sistema de gestión de incidencias Daquinet tiene un impacto significativo en la eficiencia en comparación con el sistema anterior. Los resultados respaldan la idea de que el uso del sistema Daquinet puede mejorar el tiempo requerido para manejar incidencias en comparación con los métodos manuales tradicionales.

CAPÍTULO V

5 CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones

La construcción de la aplicación Daquinet utilizando el framework Flutter y el sistema Redmine ha demostrado ser una estrategia efectiva para mejorar la gestión de incidencias en la cooperativa. La implementación de estas tecnologías ha simplificado la creación, seguimiento y resolución de tickets, proporcionando una interfaz amigable para los usuarios finales y herramientas robustas para los técnicos de soporte.

La revisión detallada del proceso de generación de tickets en la cooperativa destacó una serie de pasos cruciales que implicaban tanto a los usuarios finales como a los técnicos de soporte. La introducción de un sistema automatizado de tickets, como se pone de manifiesto en los resultados de la investigación, revela una mejora notable en la eficiencia en comparación con el sistema anterior. La adopción de tecnologías modernas, como Flutter y Redmine, ha simplificado y agilizado de manera significativa el proceso de creación y gestión de tickets. Esta simplificación no solo ha optimizado los tiempos de respuesta ante incidencias, sino que también ha mejorado la experiencia general de los usuarios finales al proporcionar una solución más rápida y eficiente.

El análisis de la eficiencia, medido a través de la velocidad de respuesta y tiempos de creación de tickets, indica que la implementación del sistema automatizado de tickets ha generado una mejora sustancial en comparación con el sistema antiguo. Los resultados detallados en la Tabla 25-4 respaldan la conclusión de que la eficiencia operativa se ha incrementado de manera significativa. La adopción de tecnologías modernas como Flutter y Redmine ha facilitado una gestión de incidencias más rápida y efectiva, lo que se traduce en beneficios tangibles para la cooperativa y sus usuarios finales.

5.2 Recomendaciones

Se recomienda explorar oportunidades de integración del sistema automatizado de tickets con otros sistemas corporativos existentes en la cooperativa. La integración efectiva puede mejorar la sinergia entre diferentes procesos y proporcionar una visión holística de la operación, facilitando la toma de decisiones informadas y la eficiencia general.

Se recomienda explorar la integración del sistema Daquinet con herramientas de colaboración, como plataformas de comunicación interna. Esto facilitará una mayor colaboración entre los equipos, mejorando la comunicación y la eficiencia en la resolución de incidencias al proporcionar un entorno centralizado para la gestión de información y recursos

También se recomienda la integración del sistema Daquinet con las plataformas de monitoreo de la cooperativa para que se pueda crear tickets automáticamente ante incidencias auto detectada

GLOSARIO

Framework: Marco de trabajo o entorno de trabajo, es una estructura o conjunto de herramientas y bibliotecas que proporciona un conjunto común de funciones y soluciones para facilitar el desarrollo de software. Estas funcionalidades suelen estar orientadas a tareas específicas, como la creación de interfaces de usuario, el manejo de bases de datos, la gestión de redes, entre otras.

Flutter: Framework que posee código abierto realizado por Google para permitir la creación de aplicaciones nativas de alta calidad para plataformas Android, iOS, web y de escritorio desde una sola base de código. Flutter utiliza el lenguaje de programación Dart y proporciona un conjunto de widgets y herramientas que facilitan la creación de interfaces de usuario atractivas y responsivas.

Redmine: Es una herramienta de gestión de proyectos y seguimiento de incidencias de código abierto. Proporciona una plataforma integral para la colaboración en proyectos, permitiendo a los equipos rastrear tareas, gestionar proyectos, asignar recursos, realizar seguimiento del tiempo y mantener un registro de incidencias o errores en el desarrollo de software. Redmine es altamente personalizable y puede adaptarse a diferentes tipos de proyectos y flujos de trabajo.

Incidencias: En el contexto de desarrollo de software, una incidencia se refiere a un problema, error o defecto que se encuentra en una aplicación o sistema. Estas incidencias suelen ser reportadas por usuarios, testers o desarrolladores y deben ser abordadas para mejorar la calidad del software. El seguimiento y la resolución de incidencias son una parte crítica del ciclo de desarrollo, ya que permiten identificar y corregir problemas para asegurar que la aplicación funcione correctamente y cumpla con los requisitos y expectativas establecidas. Sistemas como Redmine se utilizan para gestionar y dar seguimiento a estas incidencias a lo largo del ciclo de vida del proyecto.

BIBLIOGRAFÍA

1. **ACUÑA, Fátima.** “Control interno y su incidencia en la rentabilidad de empresa cobros del norte S.A”. *Revista Científica de FAREM-Estelí*. [En línea], 2013, No. 7. [Consultado: 6 julio 2023]. Disponible en: <https://revistasnicaragua.cnu.edu.ni/index.php/RCientifica/article/view/732>
2. **AGUIRRE ZEGARRA, Leonardo.** “Implementación de una estrategia de mejora continua basada en ITIL para mejorar el servicio de Service Desk en una empresa minera”. *Universidad Ricardo Palma*. [En línea], 2019, [Consultado: 6 julio 2023]. Disponible en: <https://repositorio.urp.edu.pe/handle/20.500.14138/2325?locale-attribute=en>
3. **AMAZON.** “Flutter”. *Amazon.com*. [En línea], 2018. [Consultado: 6 julio 2023]. Disponible en: <https://aws.amazon.com/es/whatis/flutter/>
4. **BENITES, Álvarez; et al.** “Sistema Web De Generación De Ticket De Atención”. *Edu.pe*. [En línea], 2017, [Consultado: 6 julio 2023]. Disponible en: http://repositorio.ulasamericas.edu.pe/bitstream/handle/upa/201/SistemaWeb_TicketsIncidencia_Emondragon_1503%20%283%29.pdf?sequence=1&isAllowed=y
5. **CALLE, Milena.** Diseño del sistema de control interno en la empresa AB OPTICAL de la ciudad de Loja. [En línea]. (Trabajo de titulación) (Ingeniería). Universidad Nacional de Loja. Loja-Ecuador. 2015. Págs. 2. Disponible en: <https://dspace.unl.edu.ec/jspui/handle/123456789/10494>
6. **CAMPOS, Joaquin.** Planificación y gestión activa de proyectos en gnu/linux con redmine. [En línea]. (Memoria final). Universitat Oberta de Catalunya. 2013. págs. 7. [Consultado 6 julio 2023]. Disponible en: <https://openaccess.uoc.edu/bitstream/10609/19274/6/jcamposITFC0113memoria.pdf>
7. **DEVS, Quality.** “Qué es Flutter y por qué utilizarlo en la creación de tus apps”. *Quality Devs*. [En línea], 2019. [Consultado: 6 julio 2023]. Disponible en: <https://www.qualitydevs.com/2019/07/05/que-es-flutter/>

8. **FERNÁNDEZ MONTESINOS, Jorge.** Implantación de un sistema de gestión de incidencias [En línea]. (Trabajo de titulación) (Ingeniería). Universitat Politècnica de Valencia. 2014. págs. 13. [Consultado 6 julio 2023]. Disponible en: <https://riunet.upv.es/handle/10251/40469>
9. **GATA, G. & GATA, W.** *El uso del sistema de aplicación basado en la web: la configuración de Redmine en la gestión de proyectos.* En: Actas del Seminario Internacional de Tecnología de la Información. Jakarta Selatan, 2010. págs. 142–147.
10. **GONZALES, Renzo.** Implementación de un sistema de service desk mediante el uso de una herramienta libre para mejorar la gestión de incidencias utilizando ITIL en una empresa pública. [En línea]. (Trabajo de titulación) (Ingeniería) Universidad Tecnológica del Perú, Facultad de Ingeniería de Sistemas e Informática. 2021. págs. 5. [Consulta: 07 de julio 2023]. Disponible en: <https://hdl.handle.net/20.500.12867/4512>
11. **GOOGLE.** "Dart". *Revista electrónica* [En línea], 2023, [Consulta: 6 de julio 2023]. Disponible en: <https://dart.dev/>
12. **LIZÁRRAGA, Kevin.** Implementación de Flutter para el desarrollo de aplicaciones móviles nativas en iOS y Android. [En línea]. (Trabajo de titulación) (Ingeniería). Universidad Politécnica de Sinaloa, Programa Académico de Ingeniería en Informática. Mazatlán, Sinaloa. 2020. págs. 30. [Consulta: 7 de julio 2023]. Disponible en: <http://repositorio.upsin.edu.mx/formatos/A031LIZARRAGAOSUNAKEVINANTONI O6608.pdf>
13. **PALOMO PASTOR, Francisco.** Desarrollo de un sistema de gestión de incidencias. [En línea]. (Trabajo de titulación) (Licenciatura en Informática). Universidad Politécnica de Madrid. 2009. págs. 7. [Consultado 6 julio 2023]. Disponible en: https://oa.upm.es/1709/1/PFC_FRANCISCO_PALOMO_PASTOR.pdf
14. **POZO, Gálvez y JAVIER, Santiago.** Desarrollo de un sistema para la administración de requerimientos e incidencias mediante tickets en una empresa de desarrollo de software. [En línea]. (Trabajo de titulación). PUCE. 2017. págs. 23. [Consultado 6 julio 2023]. Disponible en: <http://repositorio.puce.edu.ec/handle/22000/13133>

15. **PUETATE, Galo; IBARRA, José Luis.** *APLICACIONES MÓVILES HÍBRIDAS* [en línea]. PRIMERA EDICIÓN. Ibarra-Ecuador: Centro de publicaciones PUCE, [s.f.]. [Consulta: 6 de julio de 2023]. Disponible en: <https://www.pucesi.edu.ec/webs2/wp-content/uploads/2021/02/Aplicaciones-M%C3%B3viles-H%C3%ADbridas-2020.pdf>
16. **RAMÍREZ, M.; et al.** *Metodología SCRUM y desarrollo de Repositorio Digital*. 2018. Revista Ibérica De Sistemas e Tecnologias De Informação, 1062-1072. Recuperado de: <https://login.bdbiblioteca.universidadean.edu.co/login?url=https://search-proquestcom.bdbiblioteca.universidadean.edu.co/docview/2195127128?accountid=34925>
17. **SERVER IZQUIERDO, Ricardo & NAVARRO, Amparo.** Evaluación de la eficiencia de las entidades financieras en las secciones de crédito de las cooperativas [en línea]. Mazatlán, México: Dpto. de Economía y Ciencias Sociales, 2001. [Consulta: 10 de julio]. Disponible en: <https://dialnet.unirioja.es/metricas/documentos/ARTREV/112303>
18. **TAKEUCHI, Hirotaka; et al.** *The New New Product Development Game*. Lugar: Editorial, 1986. ISBN (opcional), págs. 137–146.
19. **TASHILDAR, Aakanksha.** Application Development Using Flutter. *International Research Journal of Modernization in Engineering, Technology and Science*, vol. 02, no. 08, 2020. [Consultado: 15 de julio 2023]. Disponible en: https://www.irjmets.com/uploadedfiles/paper/volume2/issue_8_august_2020/3180/1628083124.pdf
20. **VÁSQUEZ, Chávez & ELY, Imer.** Propuesta de un sistema de help desk para la gestión de incidencias de sistemas de información [En línea]. Universidad Peruana de Las Américas, Lima Perú 2022. págs. 6 - 8. [Consulta: 20 julio 2023]. Disponible en: <http://repositorio.ulasamericas.edu.pe/handle/upa/2314>



ANEXOS

ANEXO A: CARTA DE RECEPCIÓN Y ACEPTACIÓN DEL PRODUCTO DE SOFTWARE.



Riobamba, 23 de enero de 2024


Para quien corresponda,

La presente tiene como propósito formalizar la recepción y aceptación del producto de software, "SISTEMA MÓVIL CON SEGURIDAD BIOMÉTRICA PARA GESTIONAR TICKETS DE INCIDENCIAS TI EN LAS SUCURSALES DE RIOBAMBA DE LA COOPERATIVA DE AHORRO Y CRÉDITO FERNANDO DAQUILEMA", entregado por el Sr. DENNYS MAURICIO MEJA BRONCANO con número de cédula 0605830025 a la "COOPERATIVA DE AHORRO Y CRÉDITO FERNANDO DAQUILEMA LTDA."

Dicho producto, es resultado de mi proceso de titulación como estudiante de la carrera de Ingeniería de Software en la ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO, comprende diversos módulos que han sido previamente solicitados, revisados y aprobados por la empresa antes de la formalización de este documento.

Ambas partes, en señal de conformidad, procedemos a la recepción y aceptación formal de este software mediante la firma de la presente carta.


Fredy Janela
C.I. 1723634931
Coordinador de Tecnologías de la Información
COAC. Fernando Daquilema


Denny Meja
C.I. 0605830025
Estudiante
ESPOCH



COOPERATIVA DE AHORRO Y CRÉDITO FERNANDO DAQUILEMA LTDA. | RUC: 090100000000000000 | TEL: 0605830025 | WWW.COOPERATIVA-DAQUILEMA.COM

RIOBAMBA - QUITO - GUAYAS - SANTIAGO DE CALERITA - GUAYAS - LALAMA - MOYLA - CANA - AMBATO - LINDERA - GUARANDA - CUNTA - SPOC - BOLSON

ANEXO B: HISTORIAS DE USUARIO

Identificación de la Historia de Usuario:	HC-001
Nombre del Requerimiento :	Iniciar Sesión como Cliente
Usuarios:	Cientes
Descripción:	Permitir a los clientes iniciar sesión en el sistema.
Iteración Asignada:	Sprint 1
Prioridad del requerimiento:	Alta

Identificación de la Historia de Usuario:	HC-001
Nombre del Requerimiento :	Iniciar Sesión como Técnico
Usuarios:	Técnicos
Descripción:	Facilitar el inicio de sesión para los técnicos en el sistema.
Iteración Asignada:	Sprint 1
Prioridad del requerimiento:	Alta

Identificación de la Historia de Usuario:	HC-005
Nombre del Requerimiento :	Registrar Dispositivo como Cliente
Usuarios:	Cientes
Descripción:	Permitir a los clientes registrar sus dispositivos en el sistema.
Iteración Asignada:	Sprint 1
Prioridad del requerimiento:	Media

Identificación de la Historia de Usuario:	HC-002
Nombre del Requerimiento:	Gestionar Tickets como Cliente

Usuarios:	Clientes
Descripción:	Facilitar a los clientes la gestión de sus tickets en el sistema.
Iteración Asignada:	Sprint 2
Prioridad del requerimiento: Alta	

Identificación de la Historia de Usuario:	HT-002
Nombre del Requerimiento:	Gestionar Tickets como Técnico
Usuarios:	Técnicos
Descripción:	Mejorar la gestión de tickets para los técnicos en el sistema.
Iteración Asignada:	Sprint 2
Prioridad del requerimiento: Alta	

Identificación de la Historia de Usuario:	HT-004
Nombre del Requerimiento:	Cerrar Ticket como Técnico
Usuarios:	Técnicos
Descripción:	Permitir a los técnicos cerrar tickets después de resolverlos.
Iteración Asignada:	Sprint 2
Prioridad del requerimiento: Alta	

Identificación de la Historia de Usuario:	HC-007
Nombre del Requerimiento:	Filtrar Tickets por Estado
Usuarios:	Clientes
Descripción:	Brindar a los clientes la capacidad de filtrar tickets según su estado.
Iteración Asignada:	Sprint 2
Prioridad del requerimiento: Media	

Identificación de la Historia de Usuario:	HC-006
Nombre del	Consultar Estado de Tickets

Requerimiento:	
Usuarios:	Clientes
Descripción:	Permitir a los clientes verificar el estado actual de sus tickets.
Iteración Asignada:	Sprint 2
Prioridad del requerimiento: Media	

Identificación de la Historia de Usuario:	HC-003
Nombre del Requerimiento:	Ver Perfil como Cliente
Usuarios:	Clientes
Descripción:	Brindar a los clientes la capacidad de ver su perfil en el sistema.
Iteración Asignada:	Sprint 3
Prioridad del requerimiento: Media	

Identificación de la Historia de Usuario:	HT-003
Nombre del Requerimiento:	Enviar a Revisión como Técnico
Usuarios:	Técnicos
Descripción:	Permitir a los técnicos enviar tickets a revisión después de resolverlos.
Iteración Asignada:	Sprint 3
Prioridad del requerimiento: Alta	

Identificación de la Historia de Usuario:	HC-008
Nombre del Requerimiento:	Evaluar Experiencia de Usuario
Usuarios:	Clientes
Descripción:	Permitir a los clientes evaluar su experiencia con el sistema.
Iteración Asignada:	Sprint 3
Prioridad del requerimiento: Media	

Identificación de la Historia de Usuario:	HC-009
Nombre del Requerimiento:	Acceder a Historial de Tickets
Usuarios:	Clientes
Descripción:	Facilitar a los clientes el acceso a su historial de tickets en el sistema.
Iteración Asignada:	Sprint 4
Prioridad del requerimiento: Baja	

Identificación de la Historia de Usuario:	HT-006
Nombre del Requerimiento:	Modificar Estado de Tickets Asignados
Usuarios:	Técnicos
Descripción:	Permitir a los técnicos modificar el estado de los tickets asignados.
Iteración Asignada:	Sprint 4
Prioridad del requerimiento: Alta	

Identificación de la Historia de Usuario:	HT-007
Nombre del Requerimiento:	Visualizar Historial de Tickets Asignados
Usuarios:	Técnicos
Descripción:	Facilitar a los técnicos la visualización de su historial de tickets asignados.
Iteración Asignada:	Sprint 4
Prioridad del requerimiento: Media	

Identificación de la Historia de Usuario:	HC-001
Nombre del Requerimiento:	Iniciar Sesión como Cliente
Usuarios:	Clientes
Descripción:	Permitir a los clientes iniciar sesión en el sistema.
Iteración Asignada:	Sprint 1
Prioridad del requerimiento: Alta	

ANEXO C: ENCUESTA GOOGLE FORMS

Información

demográfica

Encuesta de eficiencia del sistema Daquinet

Comparte tu experiencia sobre la eficiencia al utilizar el sistema Daquinet para gestionar incidencias. Tu opinión es clave para mejorar nuestro servicio. ¡Gracias por participar!

dennysmejia27@gmail.com [Cambiar de cuenta](#)



 No compartido

* Indica que la pregunta es obligatoria

Información Demográfica

Rol en la cooperativa

- Personal Técnico
- Personal Administrativo

Experiencia con el Sistema

¿Utilizó el sistema de incidencias anterior? *

- Si
- No

Sistema nuevo Daquinet

Sistema nuevo "Daquinet"

¿Cómo calificaría la velocidad de respuesta general del sistema nuevo "Daquinet" al abrir, cerrar o actualizar tickets?

- Muy rápida
- Rápida
- Moderada
- Lenta

¿Cuánto tiempo, en promedio, tarda desde que se le presentaba un problema hasta que termina de crear un ticket utilizando el sistema nuevo Daquinet?

- Menos de 3 minutos
- 3-5 minutos
- mas de 5 minutos

¿Cuánto tiempo, en promedio, tarda desde que usted crea un ticket hasta que es aceptado por un técnico utilizando el sistema nuevo Daquinet?

- Menos de 15 minutos
- 15-30 minutos
- 30 minutos - 1 hora
- Más de 1 hora

Sistema anterior

Sistema anterior

¿Cómo calificaría la velocidad de respuesta general del sistema anterior al abrir, cerrar o actualizar tickets?

- Muy rápida
- Rápida
- Moderada
- Lenta

¿Cuánto tiempo, en promedio, tardaba desde que se le presentaba un problema hasta que terminaba de crear un ticket utilizando el sistema anterior?

- Menos de 3 minutos
- 3-5 minutos
- mas de 5 minutos

¿Cuánto tiempo, en promedio, tardaba desde que usted creaba un ticket hasta que era aceptado por un técnico utilizando el sistema anterior?

- Menos de 15 minutos
- 15-30 minutos
- 30 minutos - 1 hora
- Más de 1 hora

Preguntas de comparativa

¿En su experiencia, siente que la atención es más eficiente en términos de facilidad y velocidad al momento de crear un ticket en el sistema nuevo Daquinet?

- Sí, ha mejorado significativamente
- Sí, ha mejorado moderadamente
- No ha mejorado, ha empeorado

¿En su experiencia, siente que el tiempo de la atención de su ticket es más eficiente en el sistema nuevo Daquinet?

- Sí, ha mejorado significativamente
- Sí, ha mejorado moderadamente
- No ha mejorado, ha empeorado

ANEXO D: RESULTADO DE LA ENCUESTA

Pregunta 1	Pregunta 2	Pregunta 3	Pregunta 4	Pregunta 5	Pregunta 6	Pregunta 7	Pregunta 8	Pregunta 9	Pregunta 10
Personal Administrativo	Si	más de 5 minutos	30 minutos - 1 hora	Menos de 3 minutos	Menos de 15 minutos	Moderada	Muy rápida	Sí, ha mejorado significativamente	Sí, ha mejorado significativamente
Personal Administrativo	Si	más de 5 minutos	Más de 1 hora	Menos de 3 minutos	Menos de 15 minutos	Rápida	Muy rápida	Sí, ha mejorado significativamente	Sí, ha mejorado significativamente
Personal Técnico	Si	más de 5 minutos	30 minutos - 1 hora	3-5 minutos	Menos de 15 minutos	Moderada	Muy rápida	Sí, ha mejorado significativamente	Sí, ha mejorado significativamente
Personal Administrativo	Si	más de 5 minutos	Más de 1 hora	Menos de 3 minutos	Menos de 15 minutos	Lenta	Rápida	Sí, ha mejorado significativamente	Sí, ha mejorado significativamente
Personal Administrativo	Si	más de 5 minutos	Más de 1 hora	Menos de 3 minutos	15-30 minutos	Moderada	Muy rápida	Sí, ha mejorado moderadamente	Sí, ha mejorado moderadamente
Personal Técnico	Si	más de 5 minutos	Más de 1 hora	Menos de 3 minutos	15-30 minutos	Moderada	Muy rápida	Sí, ha mejorado significativamente	Sí, ha mejorado significativamente
Personal Administrativo	Si	más de 5 minutos	30 minutos - 1 hora	Menos de 3 minutos	Menos de 15 minutos	Moderada	Rápida	Sí, ha mejorado moderadamente	Sí, ha mejorado significativamente
Personal Administrativo	Si	más de 5 minutos	30 minutos - 1 hora	3-5 minutos	Menos de 15 minutos	Moderada	Rápida	Sí, ha mejorado significativamente	Sí, ha mejorado significativamente
Personal Administrativo	Si	más de 5 minutos	Más de 1 hora	Menos de 3 minutos	Menos de 15 minutos	Moderada	Muy rápida	Sí, ha mejorado significativamente	Sí, ha mejorado significativamente
Personal Administrativo	Si	más de 5 minutos	30 minutos - 1 hora	Menos de 3 minutos	15-30 minutos	Moderada	Muy rápida	Sí, ha mejorado moderadamente	Sí, ha mejorado moderadamente
Personal Administrativo	Si	3-5 minutos	Más de 1 hora	Menos de 3 minutos	15-30 minutos	Moderada	Muy rápida	Sí, ha mejorado	Sí, ha mejorado

ativo			hora	minutos	os			significati vamente	significati vamente
Personal Administr ativo	Si	más de 5 minutos	Más de 1 hora	3-5 minutos	Menos de 15 minut os	Lenta	Rápid a	Sí, ha mejorado significati vamente	Sí, ha mejorado moderada mente
Personal Administr ativo	Si	más de 5 minutos	30 minut os - 1 hora	Menos de 3 minutos	Menos de 15 minut os	Modera da	Muy rápida	Sí, ha mejorado moderada mente	Sí, ha mejorado significati vamente
Personal Administr ativo	Si	más de 5 minutos	Más de 1 hora	Menos de 3 minutos	15-30 minut os	Lenta	Muy rápida	Sí, ha mejorado significati vamente	Sí, ha mejorado significati vamente
Personal Técnico	Si	más de 5 minutos	30 minut os - 1 hora	Menos de 3 minutos	Menos de 15 minut os	Modera da	Muy rápida	Sí, ha mejorado moderada mente	Sí, ha mejorado moderada mente
Personal Administr ativo	Si	más de 5 minutos	30 minut os - 1 hora	3-5 minutos	Menos de 15 minut os	Modera da	Rápid a	Sí, ha mejorado significati vamente	Sí, ha mejorado moderada mente
Personal Administr ativo	Si	más de 5 minutos	30 minut os - 1 hora	3-5 minutos	Menos de 15 minut os	Modera da	Rápid a	Sí, ha mejorado significati vamente	Sí, ha mejorado significati vamente
Personal Administr ativo	Si	más de 5 minutos	Más de 1 hora	3-5 minutos	Menos de 15 minut os	Lenta	Muy rápida	Sí, ha mejorado moderada mente	Sí, ha mejorado moderada mente
Personal Técnico	Si	más de 5 minutos	Más de 1 hora	3-5 minutos	15-30 minut os	Lenta	Rápid a	Sí, ha mejorado moderada mente	Sí, ha mejorado moderada mente
Personal Administr ativo	Si	más de 5 minutos	30 minut os - 1 hora	Menos de 3 minutos	15-30 minut os	Modera da	Rápid a	Sí, ha mejorado moderada mente	Sí, ha mejorado moderada mente
Personal Administr ativo	Si	más de 5 minutos	Más de 1 hora	3-5 minutos	15-30 minut os	Modera da	Rápid a	Sí, ha mejorado moderada mente	Sí, ha mejorado significati vamente
Personal Administr ativo	Si	más de 5 minutos	Más de 1 hora	Menos de 3 minutos	15-30 minut os	Modera da	Muy rápida	Sí, ha mejorado significati vamente	Sí, ha mejorado significati vamente
Personal Administr ativo	Si	más de 5 minutos	30 minut os - 1 hora	Menos de 3 minutos	Menos de 15 minut os	Modera da	Muy rápida	Sí, ha mejorado moderada mente	Sí, ha mejorado moderada mente

Personal Administrativo	Si	más de 5 minutos	30 minutos - 1 hora	Menos de 3 minutos	Menos de 15 minutos	Moderada	Muy rápida	Sí, ha mejorado significativamente	Sí, ha mejorado moderadamente
Personal Administrativo	Si	más de 5 minutos	Más de 1 hora	3-5 minutos	Menos de 15 minutos	Lenta	Rápida	Sí, ha mejorado significativamente	Sí, ha mejorado significativamente
Personal Administrativo	Si	más de 5 minutos	30 minutos - 1 hora	3-5 minutos	15-30 minutos	Moderada	Muy rápida	Sí, ha mejorado moderadamente	Sí, ha mejorado significativamente
Personal Administrativo	Si	más de 5 minutos	Más de 1 hora	3-5 minutos	15-30 minutos	Moderada	Muy rápida	Sí, ha mejorado significativamente	Sí, ha mejorado significativamente
Personal Administrativo	Si	más de 5 minutos	30 minutos - 1 hora	Menos de 3 minutos	Menos de 15 minutos	Moderada	Muy rápida	Sí, ha mejorado significativamente	Sí, ha mejorado moderadamente
Personal Técnico	Si	más de 5 minutos	Más de 1 hora	Menos de 3 minutos	Menos de 15 minutos	Lenta	Muy rápida	Sí, ha mejorado significativamente	Sí, ha mejorado significativamente
Personal Administrativo	Si	más de 5 minutos	30 minutos - 1 hora	Menos de 3 minutos	15-30 minutos	Lenta	Muy rápida	Sí, ha mejorado significativamente	Sí, ha mejorado significativamente
Personal Administrativo	Si	más de 5 minutos	30 minutos - 1 hora	3-5 minutos	15-30 minutos	Lenta	Rápida	Sí, ha mejorado moderadamente	Sí, ha mejorado significativamente
Personal Administrativo	Si	más de 5 minutos	Más de 1 hora	3-5 minutos	Menos de 15 minutos	Moderada	Muy rápida	Sí, ha mejorado significativamente	Sí, ha mejorado significativamente

ANEXO E: INTERFACES DE USUARIO



Enrollamiento

Descripción: Tiene como función verificar si el dispositivo esta registrado en la base de datos determinando si es un dispositivo de confianza o no.

Caso positivo: Si el dispositivo es de confianza entonces se redirigirá a la pantalla de opciones de logeo.

Caso negativo: Si el dispositivo no es de confianza entonces se redirigirá al registro de dispositivo.



Registro de Dispositivo

Descripción: Tiene como función registrar el dispositivo del usuario mediante sus credenciales e información del dispositivo.

Caso positivo: Si los datos son correctos se registrará el dispositivo y se redirigirá hacia la pantalla de opciones de logeo.

Caso negativo: Si los datos no son correctos se mostrará una alerta indicando el error.



Logeo por usuario y contraseña

Descripción: Tiene como función ingresar a la aplicación mediante las credenciales del usuario pasando por un proceso de verificación.

Caso positivo: Si las credenciales son las correctas se redirigirá hacia la pantalla principal de la aplicación.

Caso negativo: Si las credenciales no son correctas se mostrará una alerta indicando el error.



Inicio / Usuario normal

Descripción: Tiene como función mostrar todas las tareas que están en desarrollo o Nuevas del usuario, mostrando la información base de las tareas. Además, se muestra las diferentes secciones de navegación disponibles y un menú en la parte posterior izquierda.

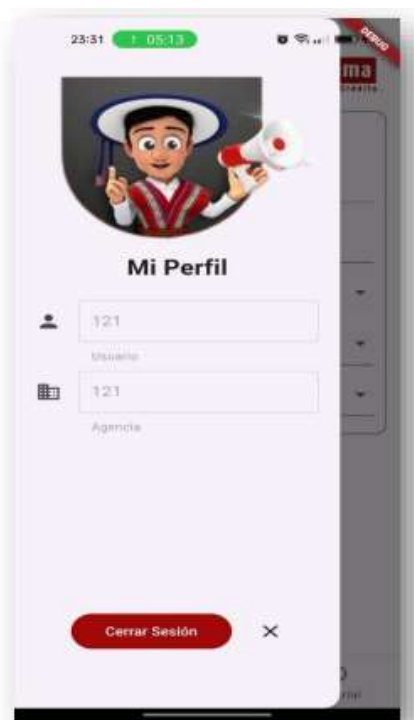
Interacción: El usuario podrá visualizar toda la información del ticket dando click en la tarea.



Inicio / Técnico

Descripción: Tiene como función mostrar todas las tareas que están en desarrollo o Nueva del usuario, mostrando la información base de las tareas. Además, se muestra las diferentes secciones de navegación disponibles y un menú en la parte posterior izquierda.

Interacción: El técnico podrá visualizar toda la información del ticket dando click en la tarea.



Mi perfil

Descripción: Un submenú del inicio teniendo como función mostrar la información base del usuario ingresado en la aplicación. Además, brindando opciones de cerrar sesión y cerrar el submenú.

Interacción: Si el usuario selecciona la opción de cerrar sesión se mostrará una alerta que debe ser confirmada, redirigiéndola hacia la pantalla de enrolamiento.



Ver / Actualizar ticket

Descripción: Tiene 2 funciones base las cuales consiste en mostrar la información completa del ticket, además, permitiendo actualizar cierta información del ticket.

Caso positivo: Si el ticket ha sido actualizado correctamente se mostrará una notificación confirmando la acción.

Caso negativo: Si el ticket no pudo ser actualizado correctamente, se mostrará una notificación informando sobre el estado.

Aclaración: Solo se podrá actualizar información del ticket que estén en estado de:

- Nueva
- EN DESARROLLO

Aquellos tickets que estén finalizados no podrán ser actualizados.



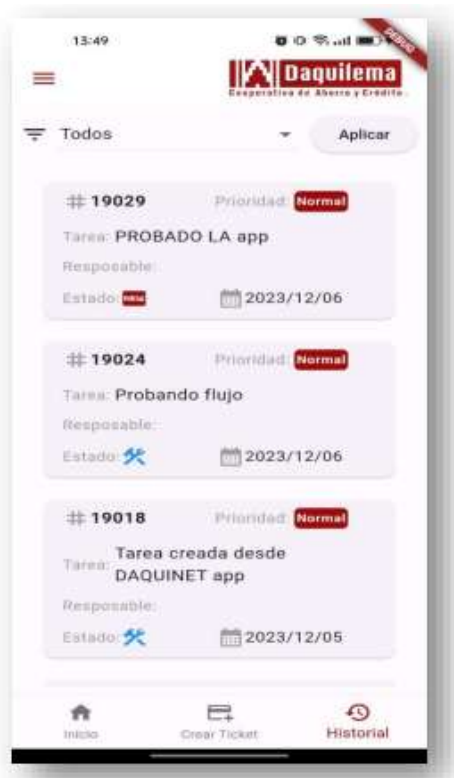
Mensajería

Descripción: Tiene como función comunicar mediante mensajes entre el técnico y el usuario.

Caso positivo: Si el mensaje ha sido enviado correctamente se mostrará una alerta confirmando el envío.

Caso negativo: Si el mensaje no pudo ser enviado se mostrará una alerta indicando que no se pudo realizar el envío.

Aclaración: Solo se envían mensajes si el ticket no está finalizado.



Historial

Descripción: Tiene como función mostrar todas las tareas que han sido creadas por el usuario y visualizarlas por su estado siendo filtradas.



Tareas Pendientes / Técnico

Descripción: Tiene como función mostrar todas las tareas que han sido creadas (Nuevas) y siguen sin ser tomadas por algún técnico.

Interacción: Para acceder a la información completa del ticket se debe seleccionar en el botón 'Ver detalles'



Ver ticket Pendiente / Técnico

Descripción: Tiene como función mostrar toda la información de una tarea pendiente (Nueva) donde el técnico puede realizar acciones sobre el ticket.

Aceptar ticket:

- Caso exitoso: Si el usuario toma el ticket y el proceso se haya realizado de manera exitosa, entonces se mostrará una alerta de éxito.

- Caso negativo: Si el usuario toma el ticket, pero el proceso no se pudo realizar de manera exitoso, se mostrará una alerta indicando sobre el proceso fallido.



Ver tickets Asignados / Técnico

Descripción: Tiene como función mostrar todos los tickets que han sido asignados a un técnico en específico, tickets que están bajo su cargo.

Aclaración: Se mostrará todos los tickets históricamente y el técnico podrá acceder a su información presionando sobre el ticket.



Ver un ticket Asignado / Técnico

Descripción: Tiene como función mostrar la información completa del ticket e interactuar con el usuario solicitante siempre y cuando el ticket no está finalizado.

ANEXO F: TEST DE SHAPIRO WILK PARA LAS RESPUESTAS DESDE LA PREGUNTA 4 HASTA LA PREGUNTA 10.

Shapiro Wilk test de las respuestas de la pregunta 4

Población: 32

H0: $X_i = N(\mu, \sigma^2)$

H1: $X_i \neq N(\mu, \sigma^2)$

Nivel de significancia: 0,05

i	X_i	$(X_i - \text{MED})^2$	A_i	$X_i \text{ INV}$	Dif ($X_i - X_i \text{ INV}$)
1	2	0,35253906	0,4188	3	-1
2	2	0,35253906	0,2898	3	-1
3	2	0,35253906	0,2463	3	-1
4	2	0,35253906	0,2141	3	-1
5	2	0,35253906	0,1878	3	-1
6	2	0,35253906	0,1651	3	-1
7	2	0,35253906	0,1449	3	-1
8	2	0,35253906	0,1265	3	-1
9	2	0,35253906	0,1093	3	-1
10	2	0,35253906	0,0931	3	-1
11	2	0,35253906	0,0777	3	-1
12	2	0,35253906	0,0629	3	-1
13	2	0,35253906	0,0485	3	-1
14	3	0,16503906	0,0344	3	0
15	3	0,16503906	0,0206	3	0
16	3	0,16503906	0,0068	3	0
17	3	0,16503906		3	
18	3	0,16503906		3	
19	3	0,16503906		3	
20	3	0,16503906		2	
21	3	0,16503906		2	
22	3	0,16503906		2	
23	3	0,16503906		2	
24	3	0,16503906		2	

25	3	0,16503906		2	
26	3	0,16503906		2	
27	3	0,16503906		2	
28	3	0,16503906		2	
29	3	0,16503906		2	
30	3	0,16503906		2	
31	3	0,16503906		2	
32	3	0,16503906		2	

Fuente: Mejia, D. 2024

Realizado por: Mejia, D. 2024

MEDIA Xi	2,59375
$\Sigma (Xi-MED)^2$	7,71875
SUMA-PRODUCTO (Ai * Dif)	-2,1848
SHAPIRO WILK correlación (P)	0,61840985
SHAPIRO WILK porcentaje mínimo	0,930

Fuente: Mejia, D. 2024

Realizado por: Mejia, D. 2024

Una vez obtenido la correlación $P = 0,61840985$ y tomando en cuenta que el porcentaje mínimo que nos indica Shapiro Wilk con nuestros datos que es de 0.930 podemos rechazar la hipótesis nula y aceptar la hipótesis alternativa que nos indica que los valores obtenidos en la encuesta **NO** pertenecen a una distribución normal.

Shapiro Wilk test de las respuestas de la pregunta 5

Población: 32

H0: $X_i = N(\mu, \sigma^2)$

H1: $X_i \neq N(\mu, \sigma^2)$

Nivel de significancia: 0,05

i	Xi	(Xi - MED) ²	Ai	Xi INV	Dif (Xi-Xi INV)
1	3	0,35253906	0,4188	4	-1
2	3	0,35253906	0,2898	4	-1
3	3	0,35253906	0,2463	4	-1
4	3	0,35253906	0,2141	4	-1
5	3	0,35253906	0,1878	4	-1
6	3	0,35253906	0,1651	4	-1
7	3	0,35253906	0,1449	4	-1
8	3	0,35253906	0,1265	4	-1
9	3	0,35253906	0,1093	4	-1
10	3	0,35253906	0,0931	4	-1
11	3	0,35253906	0,0777	4	-1
12	3	0,35253906	0,0629	4	-1
13	3	0,35253906	0,0485	4	-1
14	4	0,16503906	0,0344	4	0
15	4	0,16503906	0,0206	4	0
16	4	0,16503906	0,0068	4	0
17	4	0,16503906		4	
18	4	0,16503906		4	
19	4	0,16503906		4	
20	4	0,16503906		3	
21	4	0,16503906		3	
22	4	0,16503906		3	
23	4	0,16503906		3	
24	4	0,16503906		3	
25	4	0,16503906		3	
26	4	0,16503906		3	
27	4	0,16503906		3	
28	4	0,16503906		3	
29	4	0,16503906		3	
30	4	0,16503906		3	
31	4	0,16503906		3	
32	4	0,16503906		3	

Fuente: Mejia, D. 2024

Realizado por: Mejia, D. 2024

MEDIA Xi	3,59375
$\Sigma (Xi-MED)^2$	7,71875
SUMA-PRODUCTO (Ai * Dif)	-2,1848
SHAPIRO WILK correlación (P)	0,61840985
SHAPIRO WILK porcentaje mínimo	0,930

Fuente: Mejia, D. 2024

Realizado por: Mejia, D. 2024

Una vez obtenido la correlación P 0,61840985 y tomando en cuenta que el porcentaje mínimo que nos indica Shapiro Wilk con nuestros datos que es de 0.930 podemos rechazar la hipótesis nula y aceptar la hipótesis alternativa que nos indica que los valores obtenidos en la encuesta **NO** pertenecen a una distribución normal.

Shapiro Wilk test de las respuestas de la pregunta 6

Población: 32

H0: $X_i = N(\mu, \sigma^2)$

H1: $X_i \neq N(\mu, \sigma^2)$

Nivel de significancia: 0,05

i	X_i	$(X_i - MED)^2$	A_i	X_i INV	Dif ($X_i - X_i$ INV)
1	1	0,5625	0,4188	3	-2
2	1	0,5625	0,2898	2	-1
3	1	0,5625	0,2463	2	-1
4	1	0,5625	0,2141	2	-1
5	1	0,5625	0,1878	2	-1
6	1	0,5625	0,1651	2	-1
7	1	0,5625	0,1449	2	-1
8	1	0,5625	0,1265	2	-1
9	1	0,5625	0,1093	2	-1
10	2	0,0625	0,0931	2	0

11	2	0,0625	0,0777	2	0
12	2	0,0625	0,0629	2	0
13	2	0,0625	0,0485	2	0
14	2	0,0625	0,0344	2	0
15	2	0,0625	0,0206	2	0
16	2	0,0625	0,0068	2	0
17	2	0,0625		2	
18	2	0,0625		2	
19	2	0,0625		2	
20	2	0,0625		2	
21	2	0,0625		2	
22	2	0,0625		2	
23	2	0,0625		2	
24	2	0,0625		1	
25	2	0,0625		1	
26	2	0,0625		1	
27	2	0,0625		1	
28	2	0,0625		1	
29	2	0,0625		1	
30	2	0,0625		1	
31	2	0,0625		1	
32	3	1,5625		1	

Fuente: Mejia, D. 2024

Realizado por: Mejia, D. 2024

MEDIA Xi	1,75
$\Sigma (Xi-MED)^2$	8
SUMA-PRODUCTO (Ai * Dif)	-2,3214
SHAPIRO WILK correlación (P)	0,67361225
SHAPIRO WILK porcentaje mínimo	0,930

Fuente: Mejia, D. 2024

Realizado por: Mejia, D. 2024

Una vez obtenido la correlación $P = 0,67361225$ y tomando en cuenta que el porcentaje mínimo que nos indica Shapiro Wilk con nuestros datos que es de 0.930 podemos rechazar la hipótesis nula y aceptar la hipótesis alternativa que nos indica que los valores obtenidos en la encuesta **NO** pertenecen a una distribución normal.

Shapiro Wilk test de las respuestas de la pregunta 7

Población: 32

H0: $X_i \sim N(\mu, \sigma^2)$

H1: $X_i \not\sim N(\mu, \sigma^2)$

Nivel de significancia: 0,05

i	X_i	$(X_i - \text{MED})^2$	A_i	$X_i \text{ INV}$	Dif ($X_i - X_i \text{ INV}$)
1	1	1,80566406	0,4188	3	-2
2	1	1,80566406	0,2898	3	-2
3	1	1,80566406	0,2463	3	-2
4	1	1,80566406	0,2141	3	-2
5	1	1,80566406	0,1878	3	-2
6	1	1,80566406	0,1651	3	-2
7	1	1,80566406	0,1449	3	-2
8	1	1,80566406	0,1265	3	-2
9	1	1,80566406	0,1093	3	-2
10	1	1,80566406	0,0931	3	-2
11	2	0,11816406	0,0777	3	-1
12	3	0,43066406	0,0629	3	0
13	3	0,43066406	0,0485	3	0
14	3	0,43066406	0,0344	3	0
15	3	0,43066406	0,0206	3	0
16	3	0,43066406	0,0068	3	0
17	3	0,43066406		3	
18	3	0,43066406		3	
19	3	0,43066406		3	
20	3	0,43066406		3	
21	3	0,43066406		3	

22	3	0,43066406		2	
23	3	0,43066406		1	
24	3	0,43066406		1	
25	3	0,43066406		1	
26	3	0,43066406		1	
27	3	0,43066406		1	
28	3	0,43066406		1	
29	3	0,43066406		1	
30	3	0,43066406		1	
31	3	0,43066406		1	
32	3	0,43066406		1	

Fuente: Mejia, D. 2024

Realizado por: Mejia, D. 2024

MEDIA Xi	2,34375
$\Sigma (Xi-MED)^2$	27,21875
SUMA-PRODUCTO (Ai * Dif)	-4,0691
SHAPIRO WILK correlación (P)	0,60831503
SHAPIRO WILK porcentaje mínimo	0,930

Fuente: Mejia, D. 2024

Realizado por: Mejia, D. 2024

Una vez obtenido la correlación $P = 0,60831503$ y tomando en cuenta que el porcentaje mínimo que nos indica Shapiro Wilk con nuestros datos que es de 0.930 podemos rechazar la hipótesis nula y aceptar la hipótesis alternativa que nos indica que los valores obtenidos en la encuesta **NO** pertenecen a una distribución normal.

Shapiro Wilk test de las respuestas de la pregunta 8

Población: 32

H0: $Xi = N(\mu, \sigma^2)$

H1: $X_i \neq N(\mu, \sigma^2)$

Nivel de significancia: 0,05

i	X_i	$(X_i - \text{MED})^2$	A_i	$X_i \text{ INV}$	Dif ($X_i - X_i \text{ INV}$)
1	1	0,25	0,4188	2	-1
2	1	0,25	0,2898	2	-1
3	1	0,25	0,2463	2	-1
4	1	0,25	0,2141	2	-1
5	1	0,25	0,1878	2	-1
6	1	0,25	0,1651	2	-1
7	1	0,25	0,1449	2	-1
8	1	0,25	0,1265	2	-1
9	1	0,25	0,1093	2	-1
10	1	0,25	0,0931	2	-1
11	1	0,25	0,0777	2	-1
12	1	0,25	0,0629	2	-1
13	1	0,25	0,0485	2	-1
14	1	0,25	0,0344	2	-1
15	1	0,25	0,0206	2	-1
16	1	0,25	0,0068	2	-1
17	2	0,25		1	
18	2	0,25		1	
19	2	0,25		1	
20	2	0,25		1	
21	2	0,25		1	
22	2	0,25		1	
23	2	0,25		1	
24	2	0,25		1	
25	2	0,25		1	
26	2	0,25		1	
27	2	0,25		1	
28	2	0,25		1	
29	2	0,25		1	
30	2	0,25		1	
31	2	0,25		1	

32	2			1	
			0,25		

Fuente: Mejia, D. 2024

Realizado por: Mejia, D. 2024

MEDIA Xi	1,5
$\Sigma (Xi-MED)^2$	8
SUMA-PRODUCTO (Ai * Dif)	-2,2466
SHAPIRO WILK correlación (P)	0,63090145
SHAPIRO WILK porcentaje mínimo	0,930

Fuente: Mejia, D. 2024

Realizado por: Mejia, D. 2024

Una vez obtenido la correlación $P = 0,63090145$ y tomando en cuenta que el porcentaje mínimo que nos indica Shapiro Wilk con nuestros datos que es de 0.930 podemos rechazar la hipótesis nula y aceptar la hipótesis alternativa que nos indica que los valores obtenidos en la encuesta **NO** pertenecen a una distribución normal.

Shapiro Wilk test de las respuestas de la pregunta 9

Población: 32

H0: $Xi = N(\mu, \sigma^2)$

H1: $Xi \neq N(\mu, \sigma^2)$

Nivel de significancia: 0,05

i	Xi	$(Xi - MED)^2$	Ai	Xi INV	Dif (Xi-Xi INV)
1	2	0,390625	0,4188	3	-1
2	2	0,390625	0,2898	3	-1
3	2	0,390625	0,2463	3	-1
4	2	0,390625	0,2141	3	-1
5	2	0,390625	0,1878	3	-1
6	2	0,390625	0,1651	3	-1

7	2	0,390625	0,1449	3	-1
8	2	0,390625	0,1265	3	-1
9	2	0,390625	0,1093	3	-1
10	2	0,390625	0,0931	3	-1
11	2	0,390625	0,0777	3	-1
12	2	0,390625	0,0629	3	-1
13	3	0,140625	0,0485	3	0
14	3	0,140625	0,0344	3	0
15	3	0,140625	0,0206	3	0
16	3	0,140625	0,0068	3	0
17	3	0,140625		3	
18	3	0,140625		3	
19	3	0,140625		3	
20	3	0,140625		3	
21	3	0,140625		2	
22	3	0,140625		2	
23	3	0,140625		2	
24	3	0,140625		2	
25	3	0,140625		2	
26	3	0,140625		2	
27	3	0,140625		2	
28	3	0,140625		2	
29	3	0,140625		2	
30	3	0,140625		2	
31	3	0,140625		2	
32	3	0,140625		2	

Fuente: Mejia, D. 2024

Realizado por: Mejia, D. 2024

MEDIA Xi	2,625
$\Sigma (Xi-MED)^2$	7,5
SUMA-PRODUCTO (Ai * Dif)	-2,1363

SHAPIRO WILK correlación (P)	0,60850369
SHAPIRO WILK porcentaje mínimo	0,930

Fuente: Mejia, D. 2024

Realizado por: Mejia, D. 2024

Una vez obtenido la correlación $P = 0,60850369$ y tomando en cuenta que el porcentaje mínimo que nos indica Shapiro Wilk con nuestros datos que es de 0.930 podemos rechazar la hipótesis nula y aceptar la hipótesis alternativa que nos indica que los valores obtenidos en la encuesta **NO** pertenecen a una distribución normal.

Shapiro Wilk test de las respuestas de la pregunta 10

Población: 32

H0: $X_i = N(\mu, \sigma^2)$

H1: $X_i \neq N(\mu, \sigma^2)$

Nivel de significancia: 0,05

i	X_i	$(X_i - \text{MED})^2$	A_i	$X_i \text{ INV}$	Dif ($X_i - X_i \text{ INV}$)
1	2	0,43066406	0,4188	3	-1
2	2	0,43066406	0,2898	3	-1
3	2	0,43066406	0,2463	3	-1
4	2	0,43066406	0,2141	3	-1
5	2	0,43066406	0,1878	3	-1
6	2	0,43066406	0,1651	3	-1
7	2	0,43066406	0,1449	3	-1
8	2	0,43066406	0,1265	3	-1
9	2	0,43066406	0,1093	3	-1
10	2	0,43066406	0,0931	3	-1
11	2	0,43066406	0,0777	3	-1
12	3	0,11816406	0,0629	3	0
13	3	0,11816406	0,0485	3	0
14	3	0,11816406	0,0344	3	0
15	3	0,11816406	0,0206	3	0
16	3	0,11816406	0,0068	3	0
17	3	0,11816406		3	

18	3	0,11816406		3	
19	3	0,11816406		3	
20	3	0,11816406		3	
21	3	0,11816406		3	
22	3	0,11816406		2	
23	3	0,11816406		2	
24	3	0,11816406		2	
25	3	0,11816406		2	
26	3	0,11816406		2	
27	3	0,11816406		2	
28	3	0,11816406		2	
29	3	0,11816406		2	
30	3	0,11816406		2	
31	3	0,11816406		2	
32	3	0,11816406		2	

Fuente: Mejia, D. 2024

Realizado por: Mejia, D. 2024

MEDIA Xi	2,65625
$\Sigma (Xi-MED)^2$	7,21875
SUMA-PRODUCTO (Ai * Dif)	-2,0734
SHAPIRO WILK correlación (P)	0,59553074
SHAPIRO WILK porcentaje mínimo	0,930

Fuente: Mejia, D. 2024

Realizado por: Mejia, D. 2024

Una vez obtenido la correlación $P = 0,59553074$ y tomando en cuenta que el porcentaje mínimo que nos indica Shapiro Wilk con nuestros datos que es de 0.930 podemos rechazar la hipótesis nula y aceptar la hipótesis alternativa que nos indica que los valores obtenidos en la encuesta **NO** pertenecen a una distribución normal.

ANEXO

G:

MANUAL

TÉCNICO



Título del Proyecto: DAQUINET

Equipo de Desarrollo:

5.2.1.1 *Dennys Mauricio Mejia Broncano*

Stakeholders:

- *Empleados de la cooperativa Fernando Daquilema.*
- *Desarrolladores*
- *QA*
- *Producción*
- *Otros interesados (Dirección administrativa etc.)*

Fecha de Creación del Documento: 09/10/2023

Fecha de la Última Actualización: 15/11/2023

Versión del Documento: VR06

Responsables: Fredy Janeta, William Morocho

Riobamba – Ecuador

Propósito del Proyecto:

Facilitar a los empleados el acceso rápido y sencillo a la asistencia técnica o soporte a través de sus dispositivos móviles, permitiendo una respuesta y resolución más ágil de los problemas técnicos.

Alcance del Proyecto

5.2.1.2 *Proyecto dirigido a los empleados de la cooperativa excluyendo socios, personas y entidades ajenas.*

Únicamente se realizan solicitudes de carácter técnico.

Visión General del Sistema:

El aplicativo móvil de tickets para la asistencia a empleados es una solución tecnológica diseñada para facilitar y agilizar la gestión de solicitudes de soporte. Esta aplicación permite a los empleados solicitar fácilmente asistencia técnica y recibir atención en el menor tiempo posible.

Número de versión: VR02

Fecha de revisión: Día y fecha de revisión

Responsable: Ing. Fredy Jane

Introducción

5.2.1.3 *El presente documento corresponde al proyecto DAQUINET desarrollado por el equipo Llamings para la cooperativa Fernando Daquilema. Este proyecto busca facilitar a los empleados el acceso a soporte técnico a través de una aplicación móvil.*

El documento presenta una descripción general del sistema, definiendo su propósito, alcance, stakeholders y visión general. También se detallan los actores involucrados, los requisitos funcionales y no funcionales, casos de uso, diagramas de flujo y modelo de datos.

Para la implementación se ha conformado una suite de herramientas basadas en tecnologías ampliamente adoptadas como Java, Spring Boot, MySQL, Redmine y Bitbucket. Esto permite un desarrollo ágil y eficiente, con colaboración entre el equipo. El presente documento busca guiar el desarrollo del proyecto DAQUINET hacia los objetivos planteados inicialmente.

1 DESCRIPCIÓN GENERAL DEL SISTEMA

Tabla 1: Descripción del Usuario – Cliente

Actor	Usuario(cliente)
Rol	Cliente (empleado de la empresa)
Descripción	Este es el actor específico que asume el rol de “Cliente o Empleado de la Empresa” en el sistema. Cada empleado que accede y utiliza la aplicación móvil para interactuar con la funcionalidad de asistencia representa este actor. Los usuarios, generan, cancelan, visualizan las solicitudes de asistencia a través de la creación de tickets.
Privilegios	3 mínimos
Requerimientos asociados	<ul style="list-style-type: none">➤ RF01➤ RF02➤ RF03➤ RF04

Tabla 2: Descripción del Usuario – Técnico

Actor	Usuario (técnico)
Rol	Técnico (empleado de la empresa específicamente del departamento de Soporte (TI))
Descripción	Este es el actor específico que asume el rol de “Técnico de Soporte” en el sistema. Cada técnico que interactúa con la aplicación para revisar y responder a las solicitudes de asistencia representa este actor. Los técnicos se encargan de atender las solicitudes asignadas y proporcionar soluciones o respuestas apropiadas.
Privilegios	2 medianos
Requerimientos asociados	<ul style="list-style-type: none"> ➤ RF01 ➤ RF02 ➤ RF03 ➤ RF04 ➤ RF05

2 REQUISITOS

2.1 Requisitos funcionales

ID	RF01
Nombre	Inicio de sesión
Prioridad	Alta
Descripción	Permite a los usuarios autenticarse en el sistema proporcionando sus credenciales para acceder a funcionalidades restringidas.
Actores	Usuarios - Cliente Técnico
Dependencias	Ninguna
Requerimientos No funcionales	RNF01.
Comentarios	Se debe brindar las funcionalidad y permisos según el rol que se tenga el usuario.

ID	RF02
Nombre	Registro de dispositivo
Prioridad	Alta
Descripción	Permite a los usuarios registrar su dispositivo, proporcionando la información de su dispositivo móvil para registrarse.
Actores	<ul style="list-style-type: none"> - Técnico - Cliente
Dependencias	Ninguna.
Requerimientos No funcionales	RNF01.
Comentarios	<p>Para crear una cuenta, primeramente, se debe validar las credenciales mismas que deben constar en el registro (BD) de la cooperativa.</p> <p>El rol y la prioridad y más detalles se determinará según el tipo de cargo que este ocupe dentro de la cooperativa.</p>

ID	RF03
Nombre	Gestionar ticket
Prioridad	Alta
Descripción	Permite a los usuarios autenticados gestionar tickets.
Actores	<ul style="list-style-type: none"> - Cliente - Técnico
Dependencias	RF01 (para realizar esta acción es necesario que el usuario este logeado)
Requerimientos No funcionales	RNF01.

Comentarios	<p>El sistema debe permitir el cambio de estado según los estados disponibles:</p> <ul style="list-style-type: none"> - Nueva - En curso - Finalizada con éxito - Finalizado sin éxito. <p>Dentro de las operaciones disponibles se encuentran:</p> <ul style="list-style-type: none"> - Crear ticket - Modificar ticket - Ver ticket
--------------------	--

ID	RF04
Nombre	Ver perfil
Prioridad	Media
Descripción	Permite a los usuarios ver el perfil, misma que debe constar con su información básica.
Actores	<ul style="list-style-type: none"> - Técnico - Cliente
Dependencias	RF01 (para realizar esta acción es necesario que el usuario este logeado)

Requerimientos No funcionales	RNF01.
Comentarios	Ninguna.

ID	RF05
Nombre	Cerrar sesión
Prioridad	Alta
Descripción	Permite a los usuarios cerrar su sesión actual, desconectándose del sistema y finalizando su acceso a las funcionalidades restringidas.
Actores	<ul style="list-style-type: none"> - Técnico - Cliente - Administrador

Dependencias	RF01 (para realizar esta acción es necesario que el usuario este logeado)
Requerimientos No funcionales	RNF01.
Comentarios	Ninguna.

2.2 Extensión de requisitos

SUBPROC ESO	
Id de Requerimiento Padre	RF03
Id de Subproceso	SP01
Nombre	Aceptar ticket (Gestionar ticket)
Prioridad	Alta
Descripción	Permite a un técnico aceptar una solicitud de servicio (ticket) que se le ha sido asignado.
Actores	- Técnico
Comentarios	Es una extensión del requerimiento de GESTIONAR TICKET

SUBPROC ESO	
Id de	RF03
Requerimiento Padre	
Id de Subproceso	SP02
Nombre	Modificar ticker (Gestionar ticket)
Prioridad	Alta
Descripción	Permite a un técnico modificar el estado de un ticker de acuerdo los diferentes estados que se manejen.
Actores	- Técnico
	- Cliente
Comentarios	Es una extensión del requerimiento de GESTIONAR TICKET

--

Subproceso	
Id de RequerimientoPadre	RF03
Id de Subproceso	SP03
Nombre	Cerrar ticker (Gestionar ticket)
Prioridad	Alta
Descripción	Permite a un cliente que haya solicitado asistencia cerrar el proceso de atención donde: <ul style="list-style-type: none"> 1. Únicamente puede modificar el estado a: <ul style="list-style-type: none"> - Finalizar con éxito - Finalizar sin éxito
Actores	- Cliente
Comentarios	Es una extensión del requerimiento de GESTIONAR TICKET

2.3 Requisitos No funcionales

ID	RNF01
Categoría	Seguridad
Prioridad	Alta
Descripción	El sistema debe garantizar la confidencialidad y protección de los datos de los usuarios almacenados en la base de datos.
Objetivo	Asegurar que los datos de los usuarios no sean accesibles ni manipulables por personas no autorizadas.
Dependencias	Requisitos funcionales *
Restricciones	- El sistema debe brindar acceso únicamente a los usuarios debidamente autenticados.
Comentarios	- Incluir JWT para el manejo de tokens

2.4 Casos de uso

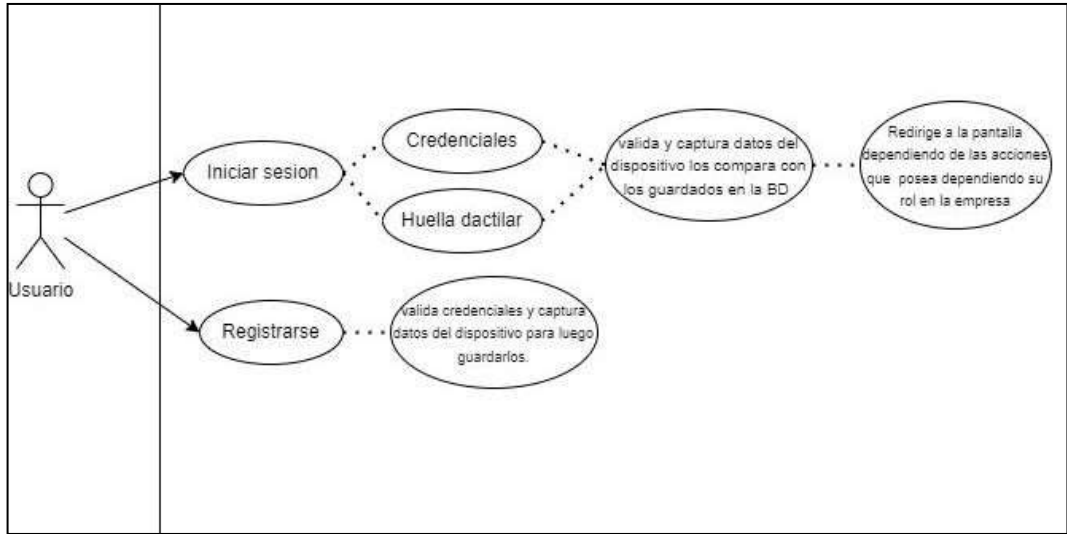


Ilustración 1: Casos de uso – usuarios

3 DISEÑO

3.1 Diagramas de flujos.

➤ *Flujo General de autenticación*

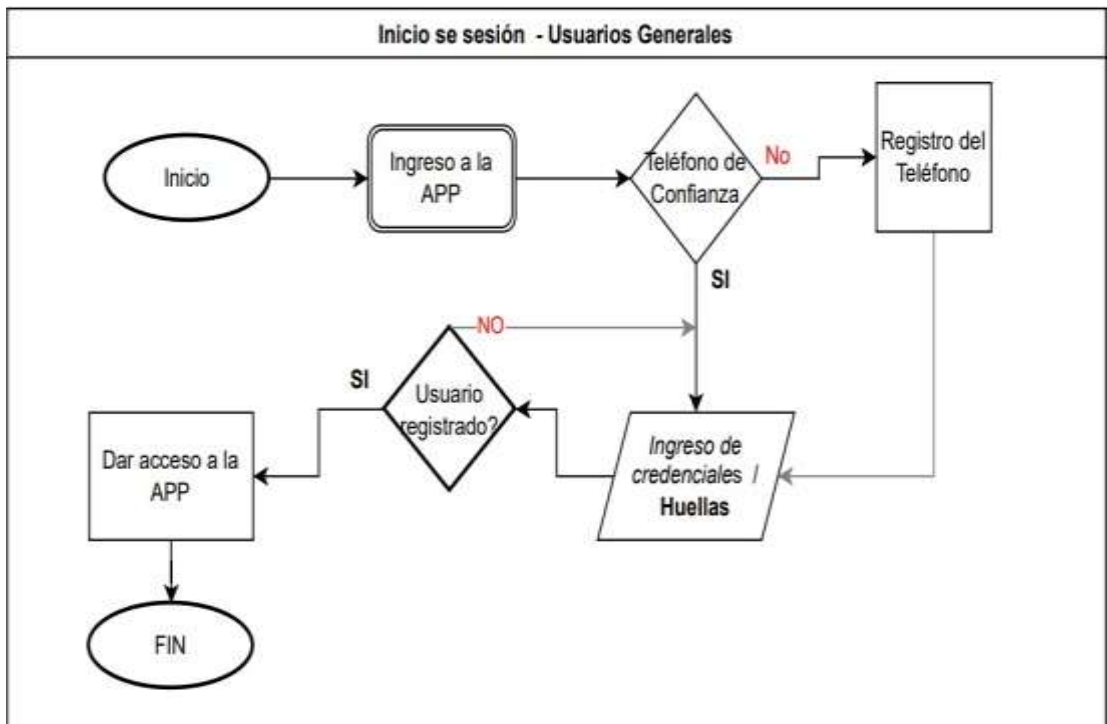


Diagrama de flujo - Acceso a la APP

➤ **Flujo de Clientes en la aplicación**

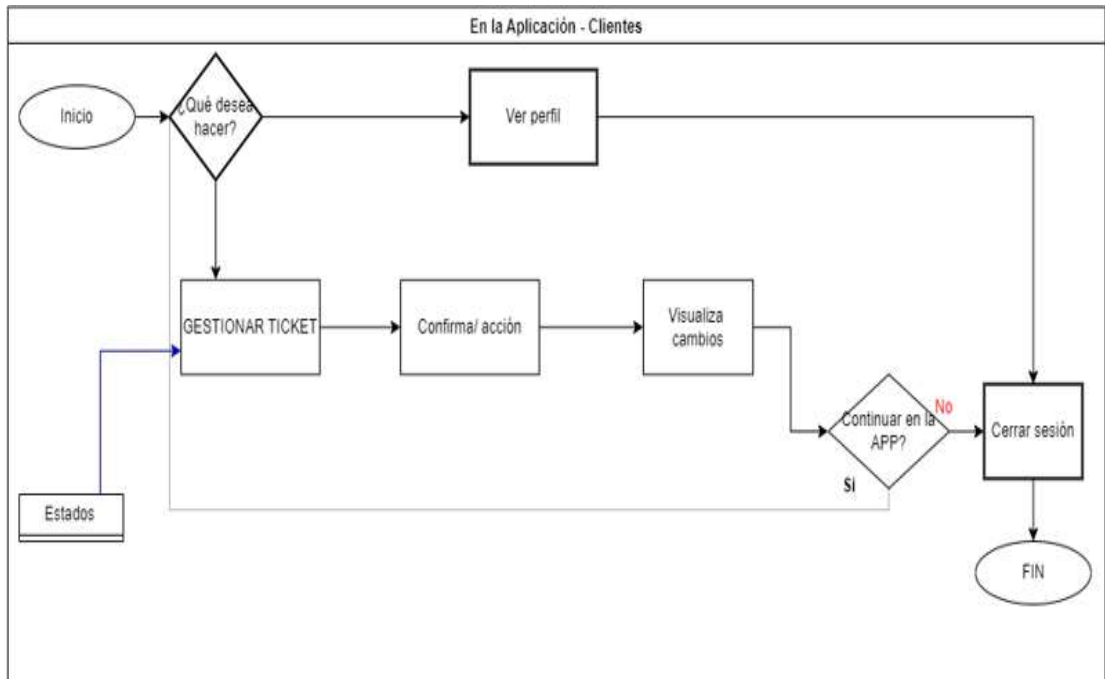
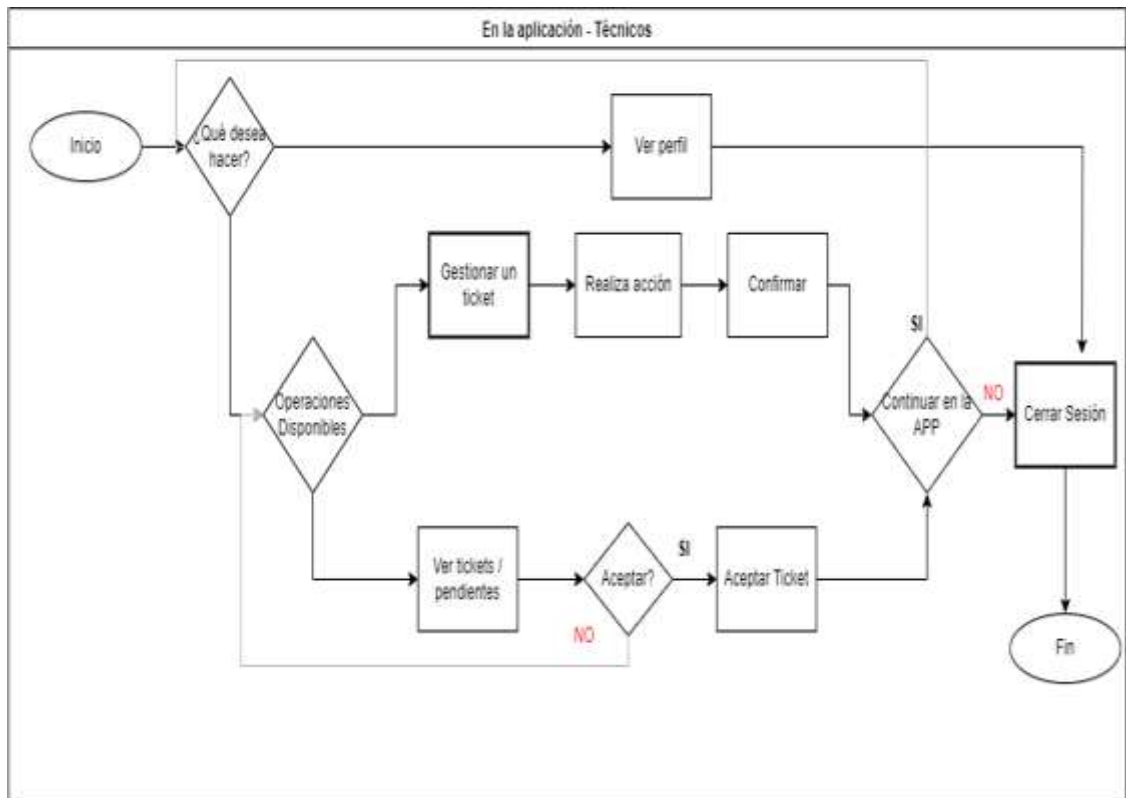
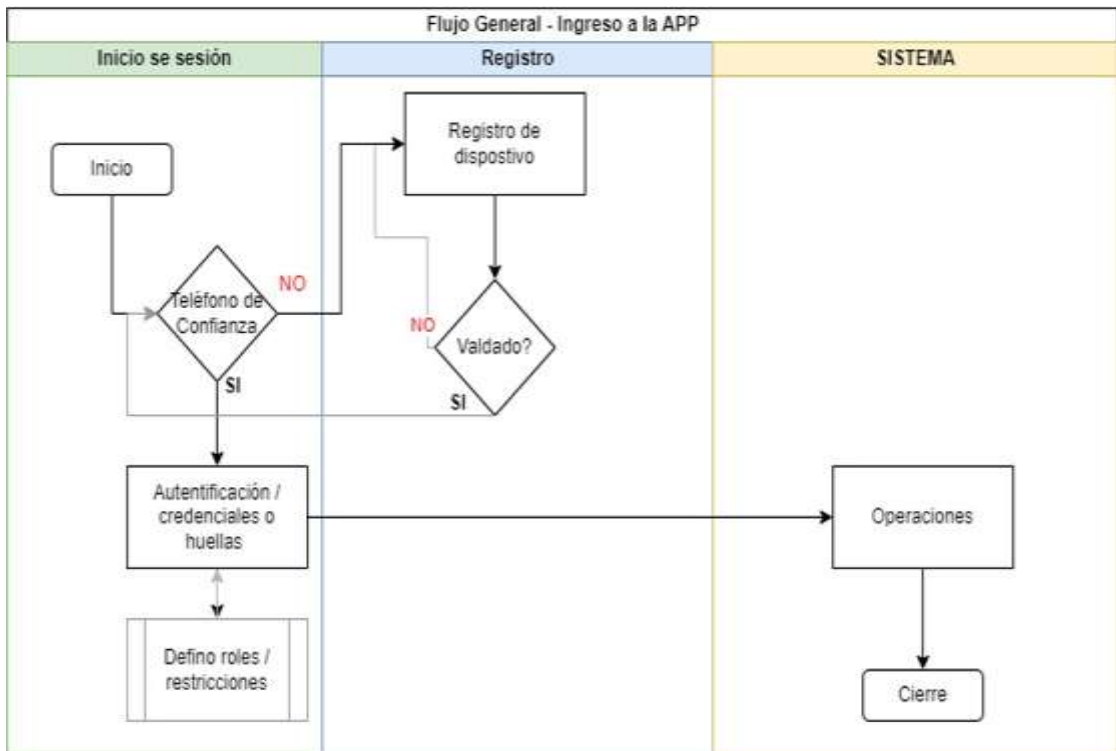


Diagrama de flujo - Gestión de Usuarios

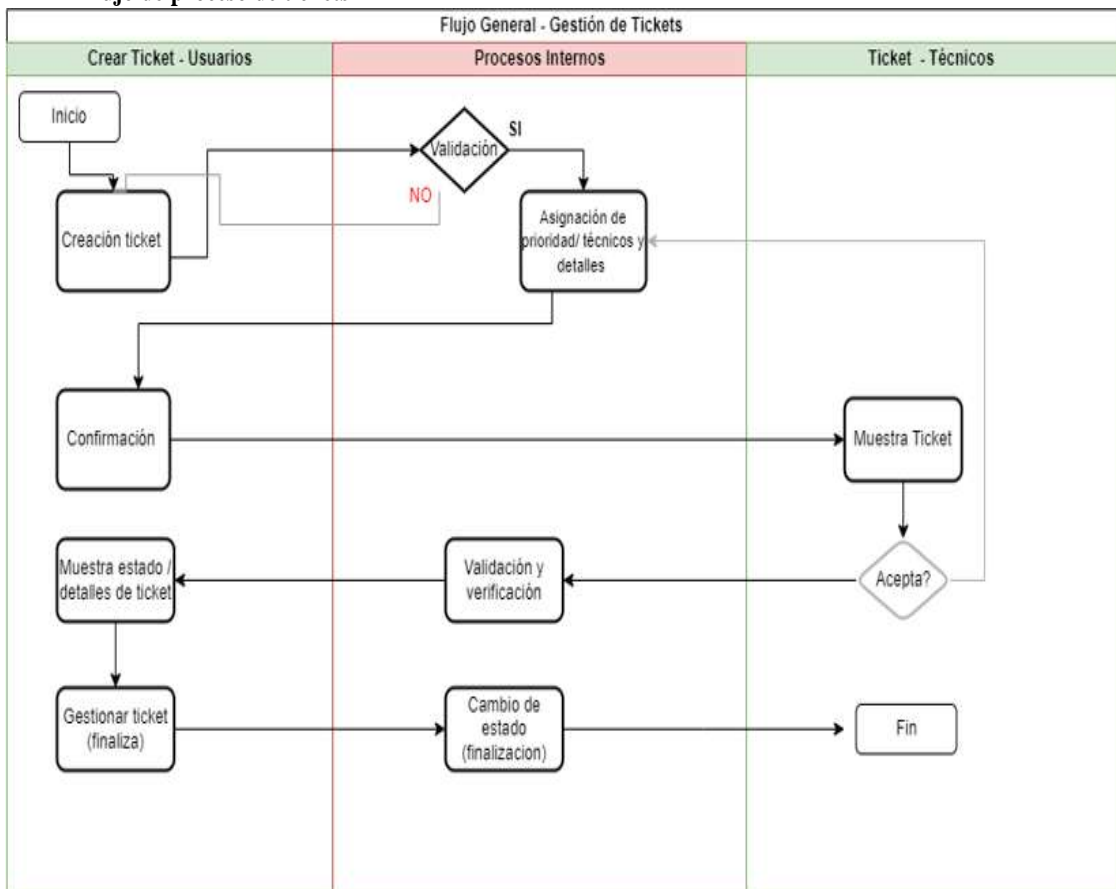
➤ **Flujo Técnicos en la aplicación**



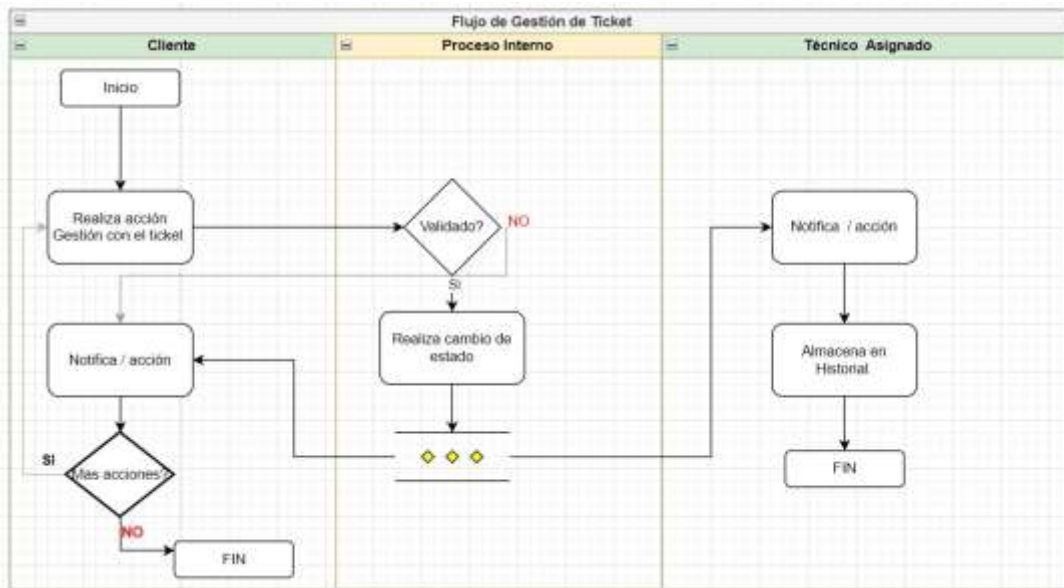
➤ Flujo de Inicio de sesión



➤ Flujo de proceso de tickets



➤ Flujo de Gestionar ticket



3.2 Modelado de datos

3.3 Diagrama entidad-relación

Entidades: TUSUARIO, TDISPOSITIVO, TDISPOSITIVOHIST, TPERMISOS

Relaciones:

- *TDISPOSITIVO* tiene FK a *TUSUARIO*
- *TDISPOSITIVOHIST* no tiene FK
- *TUSUARIOHIST* no tiene FK
- *TPERMISOS* no tiene FK

3.3.1 Diccionario de datos

- **TPERMISOS:** tabla de estados

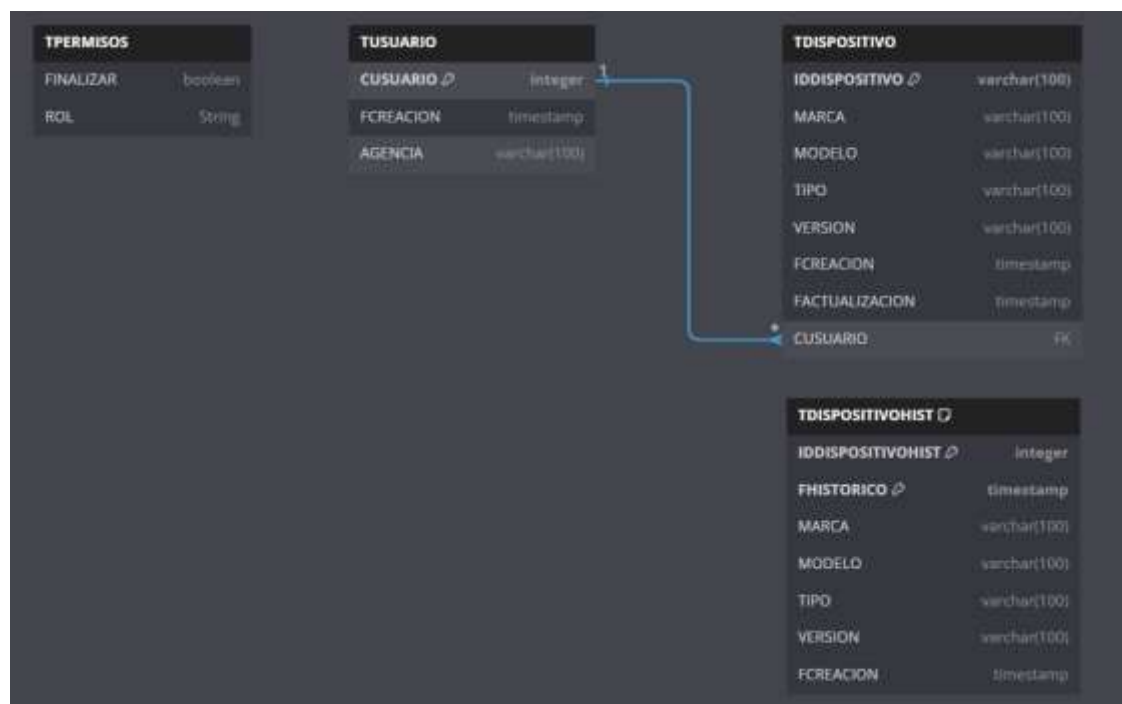
FINALIZAR: Esta columna almacena valores booleanos que indican si los roles tienen permiso para finalizar ciertas acciones o tareas. Puede ser verdadero (true) o falso (false).

ROL: Almacena el nombre o identificador del rol al que se le asigna o retira el permiso de finalización.

TUSUARIO: tabla de usuarios

CUSUARIO: llave primaria, identificación única del Usuario

FCREACION: fecha creación registro
 TDISPOSITIVO: tabla de dispositivos
 IDDISPOSITIVO: llave primaria
 MARCA: marca del dispositivo
 MODELO: modelo del dispositivo
 TIPO: tipo de dispositivo
 VERSION: versión del dispositivo
 FCREACION: fecha creación registro
 FACTUALIZACION: fecha actualización registro
 CUSUARIO: FK a TUSUARIO
 TDISPOSITIVOHIST: tabla histórica de dispositivos
 IDDISPOSITIVOHIST: llave primaria
 FHISTORICO: fecha del registro histórico, llave primaria
 MARCA: marca del dispositivo
 MODELO: modelo del dispositivo
 TIPO: tipo de dispositivo
 VERSION: versión del dispositivo
 FCREACION: fecha creación registro



4 DISEÑO DE LA ARQUITECTURA

Daquinet Tickets es una API REST desarrollada con Spring Boot 3 que implementa una arquitectura hexagonal. En los siguientes puntos se detallarán los aspectos más importantes que reflejen la aplicación de la arquitectura en este proyecto entre otros aspectos.

4.2 Arquitectura

Para el presente proyecto denominado Daquinet tickets se ha contemplado el uso de la arquitectura MVCI, lo cual nos permite desacoplar entre la lógica de negocio central de una aplicación y los detalles de infraestructura como la Diseño de la interfaz. Esta arquitectura promueve desacoplar el dominio de negocio del código de infraestructura. Las entradas y salidas (inputs y outputs) estarán a cargo de objetos de transporte de datos (DTO) siendo este el único medio de comunicación externa.

4.3 Organización de carpetas

Daquinet permite crear tareas en Redmine de asistencia técnica desde mi cliente usando a mi API de SPRING BOOT como puente.

Tengo mi organización de carpetas siendo de la siguiente manera:

Controladores: Contiene las clases que exponen los endpoints HTTP de la API REST, actuando como adaptadores hacia el exterior. Los controladores reciben las solicitudes HTTP y llaman a los servicios de dominio correspondientes para realizar las operaciones requeridas (Service). Implementan la capa de entrada de la arquitectura, encapsulando la complejidad de recibir una request HTTP y orquestar la funcionalidad del sistema. Cuentan con filtros para validar los datos entrantes desde el cliente antes de pasarlos al dominio. Esto provee una capa de seguridad adicional. Los endpoints definidos en los controladores proveen el punto de acceso de los clientes a las capacidades del sistema. Los nombres de URLs siguen convenciones REST para ser intuitivos y fáciles de consumir.

INFRA: Esta carpeta está compuesta por 3 subcarpetas y tiene como propósito contener todo lo relacionado con Configuración, Seguridad y errores. Siendo precisamente un paquete de paquetes.

JWT: esta carpeta contiene las clases que nos permiten realizar todo lo relacionado a la seguridad usando la librería de JWT permitiendo crear y validar tokens.

Config: Dentro de esta carpeta contiene clases que proporcionan la configuración del manejo de la documentación usando SWAGGER, así como la definición de rutas protegidas y rutas públicas.

Errores: Dentro de esta carpeta está contenida las clases que permiten el manejo de excepciones en SPRING BOOT.

Modelos: Dentro de esta carpeta está contenida las modelos (clases) de mi entidad misma que serán mapeadas en la base de datos mediante Hibernate ORM. Además, esta carpeta (aparte de sus clases) contiene 3 subcarpetas que son usados como puente Modelo de comunicación a REDMINE, DTOs, Auxiliares. A continuación, se detallan el propósito de cada una de las carpetas.

Auxiliares: Dentro de esta carpeta se cuenta con clases que permiten tener un modelo de clases (pero que no son mapeados en la DB). Tienen como propósito ser usadas como modelo de comunicación al cliente u en otros casos son clases embebidas que conforman las Entidades de la DB. Modelos auxiliares para comunicación con cliente y composición de entidades.

DTOs: Son clases de tipo RECORD que nos permiten transferir datos entre capas. Los DTOs funcionan como puertos de entrada/salida del dominio.

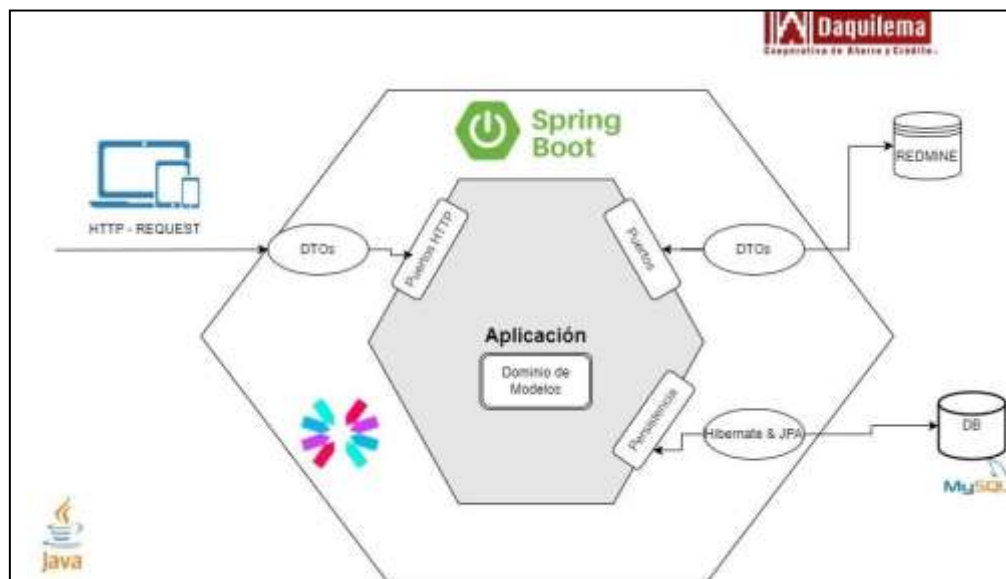
Auxiliares REDMINE: Son clases modelos propiamente usadas para tener una interacción con REDMINE. Sirven como puente de transporte de datos, pero específicamente a REDMINE.

Repositorios: Esta carpeta contiene Interfaces que implementan el acceso a datos como adaptadores de persistencia, aislando los detalles de la DB del dominio. Esto aísla completamente al dominio de cómo se realiza el acceso a datos, facilitando el cambio de tecnologías con un bajo impacto.

Service: Dentro de carpeta se encuentran las clases que realizan las operaciones usando a los repositorios manejándose aquí el dominio y la lógica de negocio en conjunto con todas las operaciones disponibles en la API. Finalmente, para hacer uso de ellas por medio de los controladores.

Utils: Conjunto de clases que permiten realizar operaciones ajenas a la DB o entidades. Contienen funciones reutilizables por cualquier clase.

A continuación, se presenta la imagen referencial a la arquitectura empleada:



Arquitectura de DAQUINET

4.4 Definición de las clases de entidades y sus atributos

Para el desarrollo de la app DAQUIENT se contempló el uso de 4 tablas que conforman la base de datos siendo:

Usuario: Es la entidad que representa al usuario que está registrado en la base de datos y teniendo como atributos a:

CUSUARIO: Es el identificador del usuario provista por el CORE de la institución, misma que funciona como clave primaria para identificarlo entre los registros.

FCREACION: Hace referencia a la fecha de creación de la cuenta en la DB

Permisos: La entidad Permisos es quien brinda información sobre si un usuario tiene los permisos necesarios para realizar ciertas acciones.

Dispositivo: Es la entidad que guarda la información del dispositivo que está vinculado a un usuario.

Histórico del dispositivo: Guarda todo el accionar que existe entre el usuario y su dispositivo creándose un registro por cada acción que se realice entre el usuario y el dispositivo.

La definición de las entidades se encuentra descritas en la carpeta de “Modelos”

Auxiliares	31/10/2023 10:39	Carpeta de arc
DTO	3/11/2023 12:42	Carpeta de arc
TDISPOSITIVO.java	1/11/2023 11:55	IntelliJ IDEA Co
TDISPOSITIVOHIST.java	26/10/2023 15:01	IntelliJ IDEA Co
TPERMISOS.java	31/10/2023 14:08	IntelliJ IDEA Co
TUSUARIO.java	31/10/2023 14:28	IntelliJ IDEA Co

Ilustración 3: Entidades

A continuación, se representa de manera ilustrativa la estructura de carpetas en el gráfico.

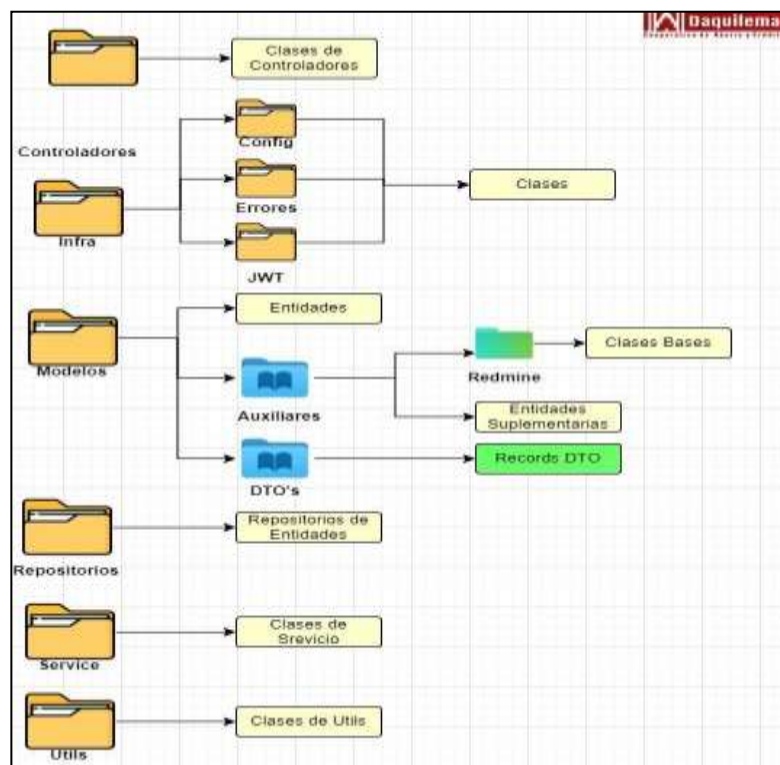


Ilustración 4: Organización de carpetas

5 IMPLEMENTACIÓN – BACK END

5.1 Herramientas

El desarrollo de aplicaciones modernas requiere una amplia gama de herramientas que cubran las diversas necesidades del ciclo de vida del software. Para el proyecto actual se ha conformado una suite de herramientas que permiten maximizar la productividad del equipo de desarrollo, la calidad de la aplicación y la colaboración

entre los miembros del equipo.

Las herramientas escogidas están basadas en tecnologías ampliamente adoptadas por la industria y con gran soporte de la comunidad. Esto provee una base sólida para construir el proyecto y resolver problemas que se presenten durante el proceso.

Adicionalmente, se ha puesto especial énfasis en la integración entre herramientas, permitiendo flujos de trabajo eficientes. Por ejemplo, la integración entre el sistema de control de versiones y la herramienta de gestión de proyectos.

La selección de herramientas también ha considerado la curva de aprendizaje, optando por aquellas que el equipo ya tiene experiencia usando, lo que se traduce en una mayor productividad desde el inicio.

A continuación, se presenta la descripción de cada herramienta seleccionada y los criterios que se consideraron para su elección. Esta suite de herramientas nos permite desarrollar el proyecto de manera ágil, eficiente y trabajando de forma coordinada entre todo el equipo.

HERRAMIENTA	RAZON DE USAR.	VERSION
Eclipse IDE	Entorno de desarrollo elegido por su amplia adopción en proyectos Java, gran cantidad de plugins disponibles y soporte extensivo para aplicaciones Spring Boot. Permite un rápido desarrollo y depuración del código.	Eclipse IDE 2023-09
Java	Uno de los lenguajes de programación más populares, con una gran comunidad de desarrolladores que provee abundante documentación y soporte. Es altamente portable, permitiendo que la aplicación se ejecute en múltiples plataformas	
MySQL	MySQL: Base de datos relacional open source, altamente escalable y con buen performance. Por su popularidad existe abundante documentación y soporte. Ideal para almacenar los datos de los tickets, usuarios, dispositivos, etc.	10.4.24- MariaDB

Spring Boot	Por su rápida curva de aprendizaje, incorpora herramientas de alto nivel que aceleran el desarrollo. Por su configuración automática y contener un servidor web (Tomcat) permite crear entregables autocontenidos.	3.14
Spring Security	Provee funciones de seguridad esenciales para una app como autenticación, autorización y cifrado. Se integra fácilmente en apps Spring Boot.	N/A
Postman	Imprescindible para probar y documentar los endpoints REST desarrollados. Permite validar rápidamente el funcionamiento y comportamiento de la API.	10.19.6
Redmine:	Herramienta colaborativa para gestión de proyectos, asignación y seguimiento de tareas entre los miembros del equipo. Fundamental para coordinación.	5.0.3
Bitbucket	Permite trabajo colaborativo entre los desarrolladores, seguimiento de cambios y versionamiento del código fuente. Integraciones disponibles con otras herramientas.	N/A
Hibernate & JPA	Liberan al desarrollador de la complejidad del mapeo objeto-relacional. Automatiza el acceso y persistencia de datos, incrementando productividad.	N/A
Flutter	Flutter es un toolkit de desarrollo de interfaces de usuario creado por Google. Siguiendo la línea de desarrollo manejada por la cooperativa, se ha tomado a esta herramienta como parte del desarrollo.	3.13.7

6 CODIFICACIÓN

La etapa de codificación se ha realizado utilizando el framework Spring Boot para crear una API REST que se comunica con la instancia de Redmine.

Se han creado los siguientes módulos:

Módulo de seguridad: Implementa la autenticación y autorización mediante JWT para proteger los endpoints privados.

Módulo de acceso a datos: Provee el acceso a la base de datos MYSQL para almacenar información de sesión y otros datos interesados.

Módulo de integración con Redmine: Contiene los servicios REST que se comunican con la API de Redmine para crear tareas, consultar proyectos, etc. Se utiliza la librería RestTemplate.

Módulo de API: Expone los endpoints REST documentados con OpenAPI para ser consumidos por el cliente móvil. Mapea las peticiones entrantes a los servicios de integración con Redmine.

La aplicación se despliega como un único artefacto JAR ejecutable. La documentación de la API se genera automáticamente a partir de las anotaciones en el código fuente.

6.1 Pruebas

Las siguientes pruebas se han desarrollado con el fin de asegurar el correcto funcionamiento de la API REST usando como herramienta a POSTMAN. Las pruebas realizadas se detallan a continuación en la siguiente tabla:

Requisito	Tipo de prueba	Descripción	Resultado esperado	Estado	Nota
RF01	Funcionalidad	Ingreso al sistema mediante el logeo con algún campo vacío	codeError: 500 Mensaje del campo vacío	Aprobado	Todos los campos son obligatorios
	Seguridad	Ingreso al sistema con alguna información errónea del dispositivo	codeError: 500 No se puede procesar la solicitud. Inténtalo luego	Aprobado	
	Seguridad	Ingreso mediante usuario correcto, pero contraseña inválida (viceversa)	codeError: 500 No se puede procesar la solicitud. Inténtalo luego	Aprobado	
RF02	Funcionalidad	Registro del dispositivo ya con algún campo inválido.	codeError: 500 Verifique que todos los campos estén completos	Aprobado	

	Seguridad	Registro de un dispositivo ya existente con diferente	codeError: 500 No se puede realizar esta acción.	PENDIENTE Falta tener otro usuario ...	
--	-----------	---	---	--	--

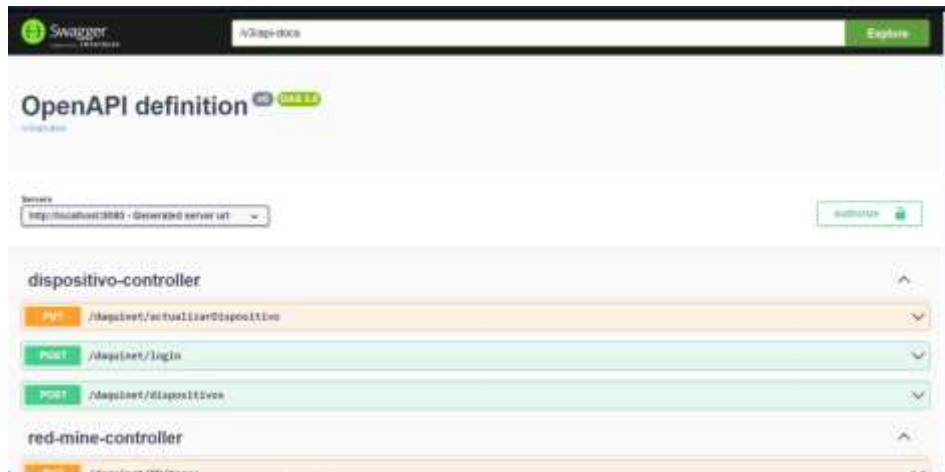
Requisito	Tipo de prueba	Descripción	Resultado esperado	Estado	Nota
	Seguridad	Registro de un dispositivo diferente con un usuario existente	codeError: 200 Dispositivo actualizado correctamente	Aprobado	
RN01	Seguridad	Acceso a un recurso o endPoint privado sin el token de JWT	403	Aprobado	
RF03	Funcionalidad	Crear una tarea con algún campo vacío	codeError: 500 Mensaje	Aprobado	Revisar que campos se consideran obligatorios y opcionales
	Funcionalidad	Actualizar una tarea ya finalizada	codeError: 500 Mensaje	Aprobado	
	Funcionalidad	Actualizar una tarea con un usuario (técnico) inexistente en REDMINE	codeError: 500 Mensaje	PENDIENTE	No existe de momento otro usuario vinculado

Funcionalidad	Crear una tarea con usuario (personal que realiza la solicitud) que no existe en la base de datos.	codeError: 500 Mensaje	Aprobado	
Funcionalidad	Actualizar tarea con algún estado que no esta considerado parte del proyecto	codeError: 500 Mensaje	Aprobado	
Funcionalidad	Actualizar tarea con una categoría no considerado parte del proyecto.	codeError: 500 Mensaje	PENDIENTE	No se tiene los permisos para obtener las categorías.
Funcionalidad	Eliminar una tarea que no esta dentro del proyecto DAQUITICKET	codeError: 500 Mensaje	Aprobado	
Funcionalidad	Ver la información de los usuarios no incluidos en el proyecto	codeError: 500 Mensaje	Aprobado	
Funcionalidad	Filtrar tareas con estados no considerados u otros caracteres	codeError: 200 Mensaje...	Aprobado	Si algún campo no esta considerado se asigna uno por default
Seguridad	Acceder a una tarea que no está dentro de DAQUITICKET	codeError: 500 No se puede procesar la solicitud. Inténtalo luego	Aprobado	

Requisito	Tipo de prueba	Descripción	Resultado esperado	Estado	Nota
	Seguridad	Acceder a la información de un usuario X no incluido en el proyecto	codeError: No se ha podido llevar a cabo esta acción, Inténtelo más luego	Aprobado	
	Seguridad	Modificar una tarea por parte de un usuario / técnico que no está registrado en la tarea	codeError: 500 Mensaje	Parcialmente completado	Se busca que un usuario y técnico puedan modificar la tarea donde están presentes. Dado que no es posible que un usuario X modifique una tarea de un usuario Y

6.2 Documentación

La documentación respectiva del proyecto se ha realizado utilizando Swagger, misma que cuenta con toda la información necesaria para realizar cualquier acción y puede ser accedida.



7 IMPLEMENTACIÓN

Daquinet App ha sido desarrollado con Flutter implementando la arquitectura limpia cual permitetener una organización de código optima, mantenible y escalable. Al definir una estructura de capas con responsabilidades bien definidas y desacopladas, se promueve la separación de conceptos, facilitando el futuro crecimiento.

7.1 Arquitectura

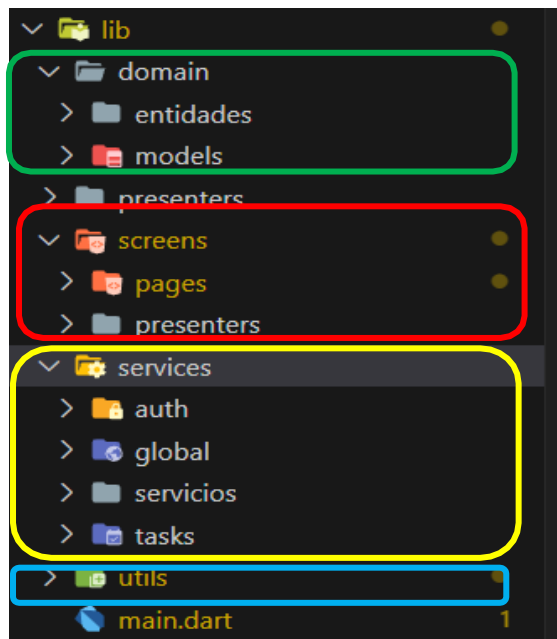
Capa de presentación (UI): Desarrollada con widgets de Flutter, contiene toda la interfaz de usuario y los elementos visuales que interactúan con el usuario.

Capa de lógica de negocio: Concentra los casos de uso referentes al flujo de tickets y las reglas de negocio. Independiente de la UI.

Capa de datos: Proporciona una interfaz común para acceder al repositorio de datos, aislando la lógica de negocio de cómo se obtienen o persisten los datos interna o externamente.

Otros: Proporciona funciones, métodos reutilizables según el contexto.

A continuación, se presenta gráficamente la representación de la arquitectura.



Capa Lógica

Capa de Presentación

Capa de datos

Otros

7.2 Diseño

7.2.1 MockUps

Los mockups o prototipos de alta fidelidad para Daquinet App se desarrollaron con la herramienta Balsamiq y Figma buscando representar de forma muy cercana el aspecto real de las interfaces antes de codificarlas.

Se generaron mockups específicos para smartphone considerando que la aplicación tiene como objetivo principal su uso en dispositivos móviles. En total se diseñaron más de 9 pantallas presentadas a continuación:

Enrollamiento



Registro de Usuario



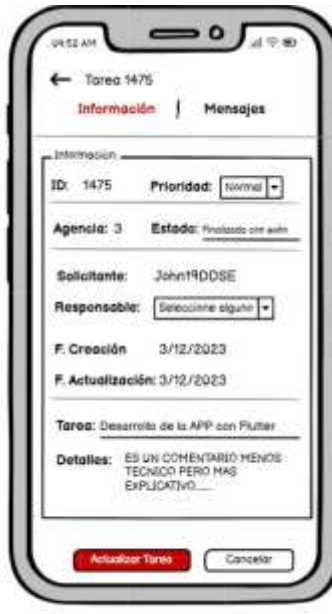
Opciones de Logeo



Inicio



Actualizar



Mensajería



Crear tarea



Historial



Implementación

Se he llevado a cabo la implementación en Flutter tomando como base los diseños de los MockUps realizados en el punto anterior. A continuación, se presenta todas las Pantallas que forman parte de Daquinet App en conjunto con su descripción y detalles sobre su función entre otras.



Enrollamiento

Descripción: Tiene como función verificar si el dispositivo esta registrado en la base de datos determinando si es un dispositivo de confianza ono.

Caso positivo: Si el dispositivo es de confianza entonces se redirigirá a la pantalla de opciones de logeo.

Caso negativo: Si el dispositivo no es de confianza entonces se redirigirá al registro de dispositivo.



Registro de Dispositivo

Descripción: Tiene como función registrar el dispositivo del usuario mediante sus credenciales e información del dispositivo.

Caso positivo: Si los datos son correctos se registrará el dispositivo y se redirigirá hacia la pantalla de opciones de logeo.

Caso negativo: Si los datos no son correctos se mostrará una alerta indicando el error.



Opciones de Logeo

Descripción: Tiene como función mostrar las diferentes opciones de logeo a la aplicación siendo:

- Usuario y contraseña
- Huella

El usuario deberá seleccionar cualquiera de las dos opciones disponibles.



Logeo por usuario y contraseña

Descripción: Tiene como función ingresar a la aplicación mediante las credenciales del usuario pasando por un proceso de verificación.

Caso positivo: Si las credenciales son las correctas se redirigirá hacia la pantalla principal de la aplicación.

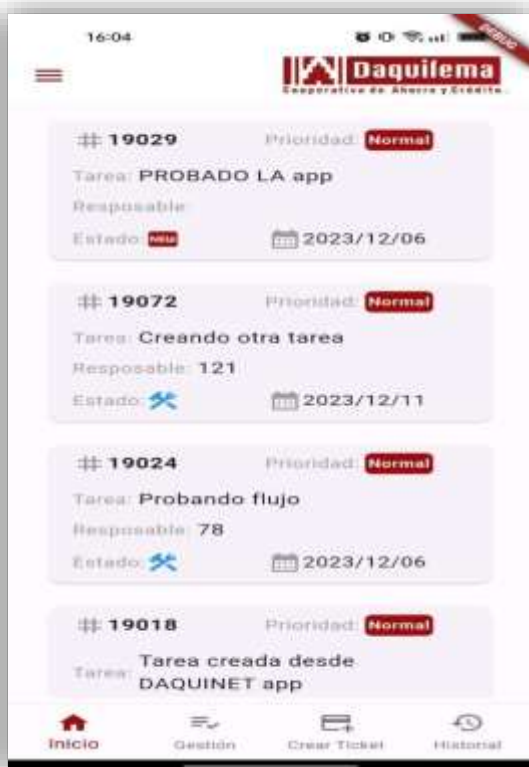
Caso negativo: Si las credenciales no son correctas se mostrará una alerta indicando el error.



Inicio / Usuario Cliente

Descripción: Tiene como función mostrar todas las tareas que están en desarrollo o Nuevas del usuario, mostrando la información base de las tareas. Además, se muestra las diferentes secciones de navegación disponibles y un menú en la parte posterior izquierda.

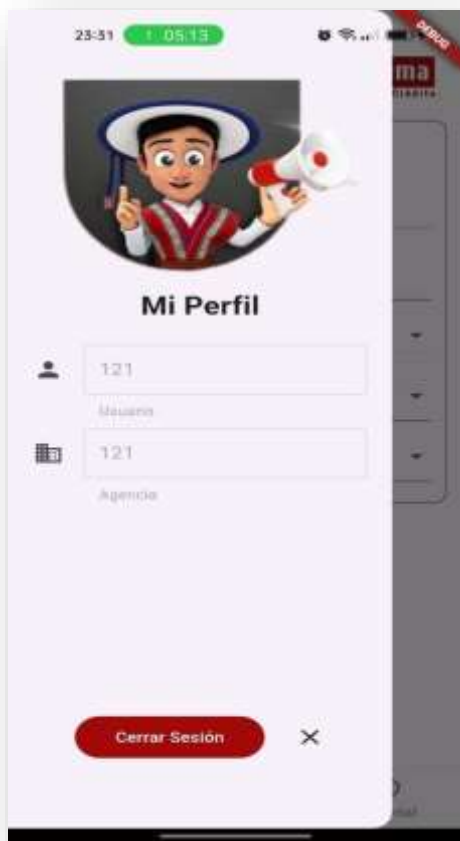
Interacción: El usuario podrá visualizar toda la información del ticket dando click en la tarea.



Inicio / Técnico

Descripción: Tiene como función mostrar todas las tareas que están EN DESARROLLO o Nueva del usuario, mostrando la información base de las tareas. Además, se muestra las diferentes secciones de navegación disponibles y un menú en la parte posterior izquierda.

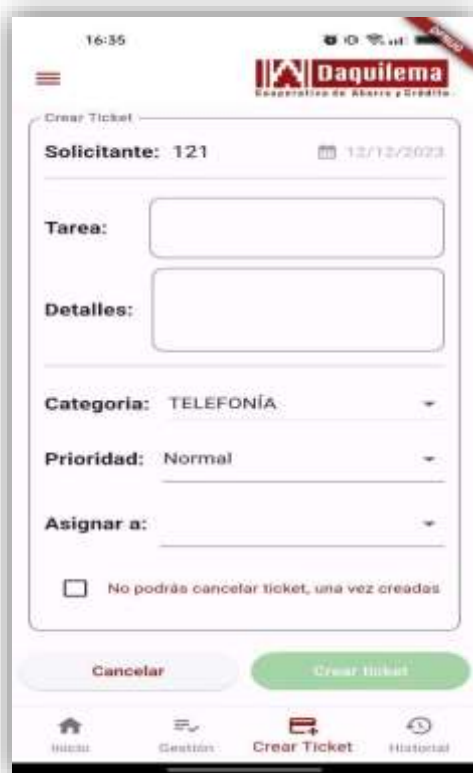
Interacción: El técnico podrá visualizar toda la información del ticket dando click en la tarea.



Mi perfil

Descripción: Un submenú del inicio teniendo como función mostrar la información base del usuario ingresado en la aplicación. Además, brindando opciones de cerrar sesión y cerrar el submenú.

Interacción: Si el usuario selecciona la opción de cerrar sesión se mostrará una alerta que debe ser confirmada, redirigiéndola hacia la pantalla de enrolamiento.



Crear ticket

Descripción: Tiene como función crear un ticket para lo cual deberá ingresar los datos del ticket obligatorio tales como:

- Tarea
- Descripción
- Categoría
- Prioridad
- Asignar (Opcional)

Caso positivo: Si el ticket ha sido registrado correctamente se mostrará una notificación confirmando la acción.

Caso negativo: Si el ticket no pudo ser registrado correctamente, se mostrará una notificación de error informando que no se pudo crear el ticket.



Ver / Actualizar ticket

Descripción: Tiene 2 funciones base las cuales consiste en mostrar la información completa del ticket, además, permitiendo actualizar cierta información del ticket.

Caso positivo: Si el ticket ha sido actualizado correctamente se mostrará una notificación confirmando la acción.

Caso negativo: Si el ticket no pudo ser actualizado correctamente, se mostrará una notificación informando sobre el estado.

Aclaración: Solo se podrá actualizar información del ticket que estén en estado de:

- Nueva
- EN DESARROLLO

Aquellos tickets que estén finalizados no podrán ser actualizados



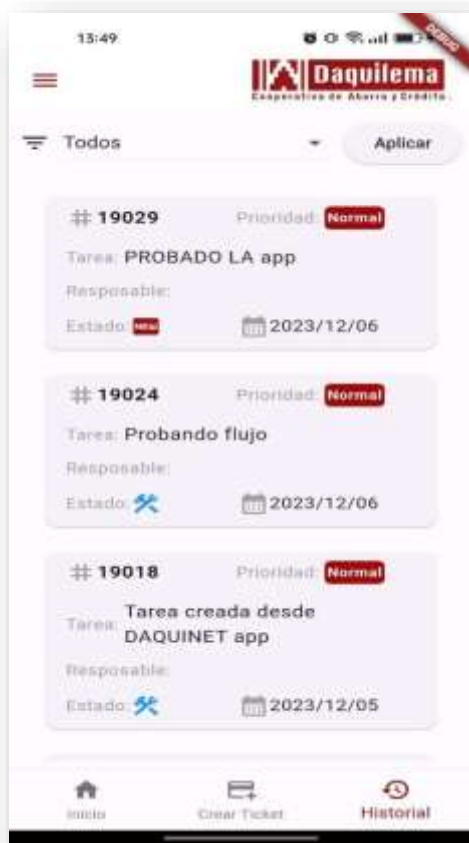
Mensajería

Descripción: Tiene como función comunicar mediante mensajes entre el técnico y el usuario.

Caso positivo: Si el mensaje ha sido enviado correctamente se mostrará una alerta confirmando el envío.

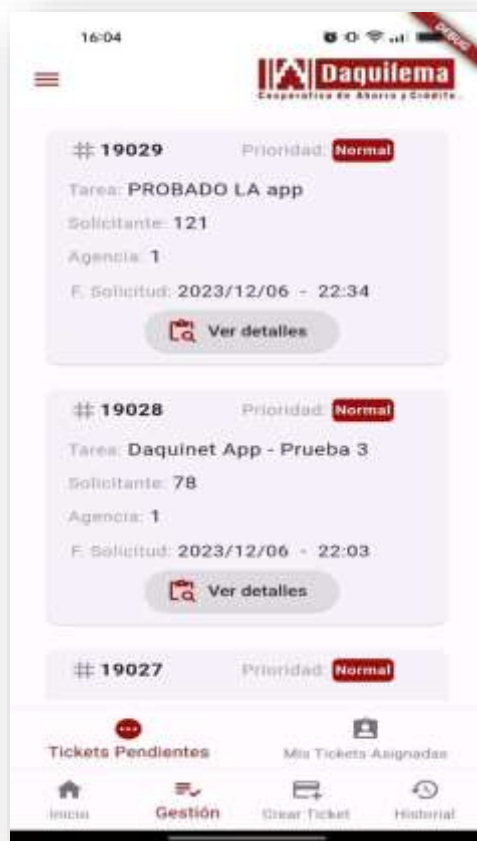
Caso negativo: Si el mensaje no pudo ser enviado se mostrará una alerta indicando que no se pudo realizar el envío.

Aclaración: Solo se envían mensajes si el ticket no finaliza.



Historial

Descripción: Tiene como función mostrar todas las tareas que han sido creadas por el usuario y visualizarlas por su estado siendo filtradas.



Tareas Pendientes / Técnico

Descripción: Tiene como función mostrar todas las tareas que han sido creadas (Nuevas) y siguen sin ser tomadas por algún técnico.

Interacción: Para acceder a la información completa del ticket se debe seleccionar en el botón 'Ver detalles'.



Ticket Pendiente / Técnico

Descripción: Tiene como función mostrar toda la información de una tarea pendiente (Nueva) donde el técnico puede realizar acciones sobre el ticket.

Aceptar ticket:

- Caso exitoso: Si el usuario toma el ticket y el proceso se haya realizado de manera exitosa, entonces se mostrará una alerta indicando éxito.
- Caso negativo: Si el usuario toma el ticket, pero el proceso no se pudo realizar de manera exitosa, se mostrará una alerta indicando sobre el proceso fallido.



Ver tickets Asignados / Técnico

Descripción: Tiene como función mostrar todos los tickets que han sido asignados a un técnico en específico, tickets que están bajo su cargo.

Aclaración: Se mostrará todos los tickets históricamente y el técnico podrá acceder a su información presionando sobre el ticket.



Ver un ticket Asignado / Técnico
Descripción: Tiene como función mostrar la información completa del ticket e interactuar con el usuario solicitante siempre y cuando el ticket no está finalizado.

8 Detalle de la API REST

En el presente apartado se presenta los detalles ya en funcionamiento de la API REST tanto de respuesta como de salida de datos entre otros.

9 ACCIONES POSIBLES DESDE LA API (INTERACCIÓN ENTRE EL CLIENTE Y REDMINE Y DB)

ID	Nombre	Acciones	Estado
1	Registrar Dispositivo	DAQUINET API debe proporcionar un registro de dispositivos vinculado a un usuario	COMPLETADO
2	Actualizar dispositivo	DAQUINET API <i>Debe permitir actualizar cualquier información (interno) de su dispositivo</i>	COMPLETADO

		DAQUINET API Debe <i>permitir actualizar la información total de un dispositivo</i>	COMPLETADO
3	Traer tareas	DAQUINET API debe permitir traer todas las tareas que estén dentro del proyecto.	COMPLETADO
		DAQUINET API debe permitir traer una tarea en específico mediante su identificador	COMPLETADO

ID	Nombre	Acciones	Estado
4	Modificar tareas	DAQUINET API debe permitir modificar información de la tarea siendo únicamente: <ul style="list-style-type: none"> ▪ A quien se le asigno ▪ El estado de la tarea ▪ La descripción ▪ Todo lo relacionado a las fechas 	COMPLETADO
5	Eliminar tarea	DAQUINET API debe permitir que se elimine una tarea en especifica	COMPLETADO
6	Mostrar miembros	DAQUINET API debe permitir mostrar los miembros del Proyecto <input type="checkbox"/> en estos casos serian aquellos a los que se puede asignar tareas (Técnicos)	COMPLETADO
7	Buscar un miembro	DAQUINET API debe permitir traer toda la información de un usuario desde la propia BASE DE DATOS	COMPLETADO
8	Devolución de Servicios	DAQUINET API debe proveer información al cliente sobre: <ul style="list-style-type: none"> ▪ Las categorías disponibles ▪ Los estados disponibles ▪ Información básica del proyecto ▪ Debe proveer información sobre las prioridades 	COMPLETADO
9	Registro de historial de dispositivo	Se debe llevar un historial de todo lo que pase con el usuario y un dispositivo	COMPLETADO

10	Crear Tarea	<p>DAQUINET API debe asegurarse que se creen las tareas con los siguientes parámetros recibidas desde el Cliente:</p> <p>Parámetros necesarios desde del cliente</p> <ul style="list-style-type: none"> ▪ CUSUARIO <input type="checkbox"/> Quien solicita ▪ A quien se lo asigna (OPCIONAL) ▪ ID SUCURSAL <ul style="list-style-type: none"> ▪ La tarea con sus parámetros • Nombre de la tarea <ul style="list-style-type: none"> • Descripción (Opcional) <p>Parámetros Complementarios</p> <ul style="list-style-type: none"> ▪ Fecha de creación manejo interno ▪ Estado ▪ Prioridad ▪ Id del Proyecto ▪ Categoría a la que pertenece 	PARCIALMENTE COMPLETADO
11	Buscar Usuario de la DB	DAQUINET API debe permitir buscar la información propia de la REDMINE	COMPLETADO

ID	Nombre	Acciones	Estado
12	Inicio de sesión mediante USUARIO Y CONTRASEÑA	DAQUINET API debe permitir iniciar sesión mediante usuario y contraseña misma que será validada por el CORE.	COMPLETADO
12	Cambio de estado	<p>DAQUINET API debe permitir el cambio de estado de una tarea dado las siguientes restricciones:</p> <ol style="list-style-type: none"> 1. No se podrá cambiar el estado de una tarea cuando este tiene ya un estado cerrado. 2. Se debe verificar un usuario este registrado en la DB y en el mismo campo tarea (CUSUARIO). 3. Verificar los permisos necesarios si 	COMPLETADO

		este trata de eliminar una tarea.	
--	--	-----------------------------------	--

a. Entrada de datos

Para el ingreso de datos se ha contemplado el uso únicamente del formato JSON como estándar en el intercambio de información a través de las APIs REST.

Las peticiones POST y PUT y GET enviadas a los endpoints deben incluir un *body (cuerpo)* con la estructura JSON esperada por cada API. Los campos obligatorios están debidamente documentados en cada especificación de endpoint utilizando **SWAGGER**.

Todos los EndPoints responden también en formato JSON estandarizado con códigos de respuesta HTTP que indican el resultado además de un cuerpo de datos (en algunos casos es nulo).

Para ejemplificar, se presenta un formato JSON de un método **POST**:

```
1  {
2    "IDDISPOSITIVO" : "A55py723S07457P45E4552" ,
3    "MODELO" : "SAMSUNG S9" ,
4    "TIPO" : "SMARTPHONE" ,
5    "MARCA" : "SAMSUNG" ,
6    "VERSION" : "12.0"
7  }
```

b. Salida de datos

Para la salida de los datos se utiliza el formato JSON como estándar de respuesta mismas que vienen acompañadas con un body con un formato establecido de respuesta siendo la siguiente:

codeError: Número entero que determina el tipo de respuesta en la acción.

message: Mensaje que indica la descripción de la acción.

content: Cuerpo de información de la acción. Aquí es donde se representa toda la información necesaria devuelta.

```
{
  "codeError": 200,
  "message": "Autenticación exitosa",
  "content": {
    "autenticacionExitosa": true,
    "IdDispositivo": "A55py723S07457P45E4552",
    "cUsuario": "121",
    "token": "eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJBNTI1IFBuhez0A3N4nn0x05Qs0GNk387EYKi1n1nnUDwLG"
  }
}
```

c. Interpretación de respuesta

Se ha planteado el uso de dos únicos códigos de respuesta para la interpretación de una acción siendo detallados a continuación:

Código 200: Código usado para indicar que el procedimiento o acción ha sido procesada de manera correcta.

```
{
  "codeError": 200,
  "message": "Autenticación exitosa",
  "content": {
    "autenticacionExitosa": true,
    "IdDispositivo": "A55py723S07457P45E4552",
    "cUsuario": "121",
    "token": "eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJBNTI1IFBuhez0A3N4nn0x05Qs0GNk387EYKi1n1nnUDwLG"
  }
}
```

Código 500: Código usado para indicar que un procedimiento no pudo ser procesada.

```
1 {
2   "codeError": 500,
3   "message": "Verifique que todos los campos esten completos",
4   "content": [
5     "El campo IDDISPOSITIVO no puede ser nulo"
6   ]
7 }
```

d. Gestión de Logs

Con el fin de llevar un registro detallado de eventos y facilitar la depuración y monitoreo, la aplicación implementa una sólida gestión de logs.

Los logs son generados automáticamente por la aplicación por cada acción que se realice y registran información como:

- Fecha y hora exacta del evento
- Nivel de severidad (INFO, WARN, ERROR)
- Origen y contexto (clase, método, línea de código)
- Mensaje descriptivo de la operación realizada
- Identificadores únicos (usuario, sesión, transacción)

Además, por cada nivel de severidad se tiene un formato mismo que se detalla a continuación:

- INFO: muestra cada interacción al inicio y fin de cada proceso indicando un mensaje representativo de la acción junto a la dirección IP del solicitante tal y como se muestra a continuación:

```
17:01:22.941 [http-nio-8080-exec-1] INFO c.o.Api.Service.UsuarioService - Se ha solicitado la información de la base de datos de usuario 121 desde 0.0.0.0:0.0.0.1
Hibernate:
  select
    t1_0.cusuario,
    t1_0.fcreacion
  from
    tusuario t1_0
  where
    t1_0.cusuario=1
17:01:22.946 [http-nio-8080-exec-1] INFO c.o.Api.Service.UsuarioService - Se ha encontrado al usuario: 121
```

- ERROR: Se muestra siempre y cuando algo salió con su mensaje previo y mostrando el error en consola mediante un manejo de errores tal y como se muestra a continuación:
- WARM: Se muestra el mensaje que indica situaciones anormales que no son parte de los errores con un mensaje previo y mostrando la causa del error:

```
17:05:35.265 [http-nio-8080-exec-10] ERROR c.Daquilema.Api.Utils.UtilRequest - ERROR al realizar la solicitud:
org.springframework.web.client.HttpClientErrorException$UnprocessableEntity: 422 Unprocessable Entity: [{"errors":["Asignado a no es válido"]}
  at org.springframework.web.client.HttpClientErrorException.create(HttpClientErrorException.java:133)
  at org.springframework.web.client.DefaultResponseErrorHandler.handleError(DefaultResponseErrorHandler.java:183)
  at org.springframework.web.client.DefaultResponseErrorHandler.handleError(DefaultResponseErrorHandler.java:117)
  at org.springframework.web.client.ResponseErrorHandler.handleError(ResponseErrorHandler.java:63)
  at org.springframework.web.client.RestTemplate.handleResponse(RestTemplate.java:513)
  at org.springframework.web.client.RestTemplate.doExecute(RestTemplate.java:864)
  at org.springframework.web.client.RestTemplate.execute(RestTemplate.java:764)
  at org.springframework.web.client.RestTemplate.exchange(RestTemplate.java:546)
  at coac.Daquilema.Api.Utils.UtilRequest.requestHttp(UtilRequest.java:58)
```



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
CERTIFICADO DE CUMPLIMIENTO DE LA GUÍA PARA
NORMALIZACIÓN DE TRABAJO DE FIN DE GRADO

Fecha de entrega: 12/04/2024

INFORMACIÓN DEL AUTOR

Nombres – Apellidos: Dennys Mauricio Mejía Broncano

INFORMACIÓN INSTITUCIONAL

Facultad: Informática y Electrónica

Carrera: Ingeniería de Software

Título a optar: Ingeniero de Software


Ing. Raúl Hernán Rosero Miranda
DIRECTORA DEL TRABAJO
DE INTEGRACIÓN CURRICULAR


Ing. María Belén Paredes Regalado
ASESOR DEL TRABAJO
DE INTEGRACIÓN CURRICULAR