



UNIVERSIDAD TECNOLÓGICA ISRAEL

MEMORIA DEL PROYECTO DE FIN DE CARRERA

**INGENIERÍA EN SISTEMAS
DE INFORMACIÓN**

**TEMA:
PROTOTIPO DE APLICACIÓN WEB-MÓVIL PARA EL REGISTRO Y CONTROL DE
LAS RUTAS ALIMENTADORAS DE LA EMPRESA PUBLICA METROPOLITANA
DE PASAJEROS DE QUITO (EPMTPQ)**

**AUTOR:
MARTINEZ LUCAS EDWIN HAROLD**

**TUTOR:
Mg. RENATO TOASA**

**QUITO, ECUADOR
2024**

UNIVERSIDAD TECNOLÓGICA ISRAEL

APROBACIÓN DEL TUTOR

En mi calidad de Tutor del Trabajo de Titulación certifico:

Que el trabajo de titulación **“PROTOTIPO DE APLICACIÓN WEB-MÓVIL PARA EL REGISTRO Y CONTROL DE LAS RUTAS ALIMENTADORAS DE LA EMPRESA PUBLICA METROPOLITANA DE PASAJEROS DE QUITO (EPMTPQ)”**, presentado por Edwin Harold Martínez Lucas, estudiante de la Carrera Ingeniería en Sistemas de Información, reúne los requisitos y méritos suficientes para ser sometido a la evaluación del Tribunal de Grado, que se designe, para su correspondiente estudio y calificación.

Quito, marzo 2024

TUTOR

Mg. Renato Toasa

TABLA DE CONTENIDOS

INTRODUCCIÓN	5
Antecedentes de la situación objeto de estudio	5
Planteamiento del problema.....	5
Justificación	5
Objetivos.....	6
General.....	6
Objetivos específicos	6
Alcance	6
CAPITULO 1. PROPUESTA	9
1.1 Diagnóstico de la situación actual	9
1.2 Recopilación de información	10
1.3 Ámbito del software.....	10
1.4 Funciones del producto	11
1.5 Características de los usuarios del sistema	14
1.6 Requisitos.....	15
1.5.1 Funcionales.	15
1.5.2 No funcionales.	17
CAPITULO 2. RESULTADOS.....	18
2.1 Metodología	18
2.2 Diseño general	19
2.2.1 Diagrama de la arquitectura del sistema	20
2.2 Diseño de interfaces.....	21
2.3 Estándares de programación utilizados.....	25

2.4	Pruebas.....	27
2.4.1	Pruebas Unitarias:	28
2.4.2	Pruebas de Estrés o Carga	29
2.5	Implementación	32
2.6	Requerimientos de hardware y software.....	32
2.6.1	Software para servidor	32
2.6.2	Software para usuario.....	32
2.6.3	Hardware para servidor	32
2.6.4	Hardware para usuario	33
CONCLUSIONES		34
RECOMENDACIONES.....		36
REFERENCIAS BIBLIOGRÁFICAS		1
ANEXOS		2

LISTA DE FIGURAS

Figura 1 <i>Diagrama de flujo de proceso manual</i>	9
Figura 2 <i>Diagrama de flujo de proceso actual</i>	10
Figura 3 <i>Base de datos del sistema</i>	20
Figura 4 <i>Diagrama de arquitectura del sistema</i>	20
Figura 5 <i>Mockup 1: Registro del usuario</i>	21
Figura 6 <i>Mockup 2: Ingreso del usuario</i>	21
Figura 7 <i>Mockup 3: Pantalla Principal</i>	22
Figura 8 <i>Mockup 4: Ingreso de registros</i>	22
Figura 9 <i>Mockup 5: Ingreso de multas</i>	22
Figura 10 <i>Mockup 6: Registro del usuario</i>	23
Figura 11 <i>Mockup 7: Ingreso del usuario</i>	23
Figura 12 <i>Mockup 8: Contraseña olvidada</i>	24
Figura 13 <i>Mockup 9: Registros Actuales</i>	24
Figura 14 <i>Mockup 10: Multas</i>	25
Figura 15 <i>Código de pruebas</i>	28
Figura 16 <i>Resultado de prueba 1</i>	29
Figura 17 <i>Prueba de stress: Usuarios recurrentes</i>	31
Figura 18 <i>Prueba de stress: Registros recurrentes</i>	31

LISTA DE TABLAS

Tabla 1	<i>Historias de Usuario Gestión de Seguridad</i>	12
Tabla 2	<i>Historias de Usuario Gestión de Cuenta</i>	12
Tabla 3	<i>Historias de Usuario Gestión de empleos</i>	12
Tabla 4	<i>Historias de Usuario Gestión de empleos</i>	13
Tabla 5	<i>Historias de Usuario Gestión de Reportes</i>	13
Tabla 6	<i>Historias de Usuario Gestión de Reportes</i>	14
Tabla 7	<i>Perfiles de usuarios para el sistema</i>	14
Tabla 8	<i>Estándares de programación en laravel</i>	25

INTRODUCCIÓN

Antecedentes de la situación objeto de estudio

La Empresa Pública De Transporte de pasajeros de Quito (EPMTPQ) actualmente es la principal empresa de movilidad de Quito movilizando cada año a 214'932.470 usuarios, el sistema cubre una distancia total de 56.5 km por todo Quito con todos sus corredores y rutas alimentadoras, además la empresa cuenta con múltiples departamentos como lo es la coordinación de recaudación que donde el promotor de este proyecto labora y es el motivo principal sobre el cual está dirigido el presente proyecto, actualmente existe un proceso en la empresa el cual conlleva registrar la serie de los boletos vendidos y está realizado de manera manual y por lo tanto existen varios errores u omisiones en el antes mencionado

Planteamiento del problema

El problema objeto de investigación se debe a que, la empresa afronta una inconsistencia en especies vendidas al que se le denomina "Evasión de pasajes" lo que conlleva a pérdida de dinero para la empresa y, por lo tanto, a la ciudadanía que usa el servicio de transporte, además se lleva de manera manual y sobre papel y en ocasiones como dueño del proceso se ha podido observar falencias en el mismo.

Por otra parte, el problema del usuario consiste en la pérdida general de recursos financieros, esto afecta de manera general a toda la ciudad, además de causar que el control sea menos eficiente, debido a que el usuario utiliza el servicio de manera fugaz y no tiene tiempo para explicar si le entregaron el boleto o no

Justificación

El prototipo propuesto es importante porque cambiará un proceso manual a uno sistematizado lo cual dará paso a muchas posibilidades de mejora y seguimiento, además de que podrá reducir la evasión de pasajes mediante el control de parte del personal involucrado, con esto se espera impulsar el proceso reduciendo recursos utilizados y tiempos de espera

El presente proyecto está orientado a realizarse con ODS 9: Industria, Innovación e infraestructura, el cual indica que intenta crear infraestructuras sostenibles, resilientes y de alta calidad para todos, fomentar la innovación, la tecnología y la investigación, y lograr un acceso a la información y al conocimiento a través de Internet. Se ha utilizado este ODS debido a que se va a automatizar un proceso y se fomenta la tecnología e investigación, lo que nos llevara a desarrollar un sistema web el cual se realizara con “Laravel” un framework de php que se ha hecho popular en los últimos años además de utilizar Bootstrap y HTML 5 para darle un diseño actual y que tendrá como gestor de base de datos MySQL 8.0

Objetivos

General

Desarrollar un Prototipo de aplicación web-móvil para el registro y control de las rutas alimentadoras de la Empresa Publica Metropolitana de pasajeros de quito (EPMTPQ)

Objetivos específicos

- Levantar los respectivos requerimientos para el desarrollo web y móvil
- Diseñar el modelo de datos del prototipo
- Codificar la solución mediante la metodología ágil KANBAN
- Validar la codificación mediante pruebas

Alcance

El presente proyecto constará de un prototipo web que tendrá un módulo de registro el cual constará de formularios de registro con validaciones por tipo de dato y cajas de selección de datos ya ingresados para ofrecer eficacia en el proceso, el módulo web también registra los datos con la fecha y hora actuales por lo que cada registro será confiable

El proyecto también tiene un prototipo móvil que tiene como propósito, el seguimiento y control, y constara de un módulo de selección de ruta y visualización de registros ordenados por cada unidad y ruta en la que se encuentre

Como valor agregado al proyecto y dueño del proceso, se implementará una columna de un estimado de boletos vendidos para aportar al seguimiento y control de las unidades además del desarrollo e interfaz de acuerdo a la usabilidad y comodidad del recaudador a cargo

Definición de componentes del Proyecto

Web

En este apartado de la aplicación se podrán registrar los datos de cada unidad con la fecha y hora actuales, lo cual ofrece veracidad al proceso de una manera confiable

Módulo de Autenticación: Este módulo se encarga de la gestión de la identidad del administrador del sistema y además de la característica de recuperación de contraseña.

Módulo de registro: En este módulo se ingresarán los datos respectivos de cada unidad con su respectiva ruta y tiquetera

Módulo de gestión de usuarios: En este módulo se gestionarán los usuarios registrados, es decir se podrá editar el nombre y la contraseña, así como los datos del mismo, es decir nombre, cedula, teléfono, edad, dirección.

Módulo de Reportes: Este módulo es un componente importante de un sistema que tiene como objetivo generar informes de las unidades que trabajaron al día filtradas por ruta

Móvil

En el apartado móvil se van a poder mostrar los datos en tablas, organizados de tal manera que cada registro lleve su respectiva unidad y los datos de la fecha y hora exactas en las que fueron registradas, así como el estimado de boletos vendidos por vuelta

Módulo de registro y autenticación: Este módulo se encarga de la gestión de la identidad de los usuarios en el sistema.

Módulo de control y seguimiento: Este módulo se encargará del control de las unidades mostrando los datos como el número de vuelta y registro de series de las tiqueteras, por lo que si hay alguna novedad se podrá enviar una notificación

Módulo de notificaciones: En este módulo se podrán recibir notificaciones de los boletos vendidos por vuelta de cada unidad, además de avisar si hay alguna inconsistencia

Módulo de control en reincidencias: En este módulo se podrá resaltar si una unidad tiene reincidencias en el control, es decir se podrá saber si la unidad tiene inconsistencias en la venta de boletos de manera repetida

Como podemos observar en el apartado móvil tendremos la posibilidad de controlar la venta de especies vendidas de una manera estructurada y con datos reales recogidos de registros anteriores para así poder notificar al personal respectivo para que proceda con la sanción correspondiente

CAPITULO 1. PROPUESTA

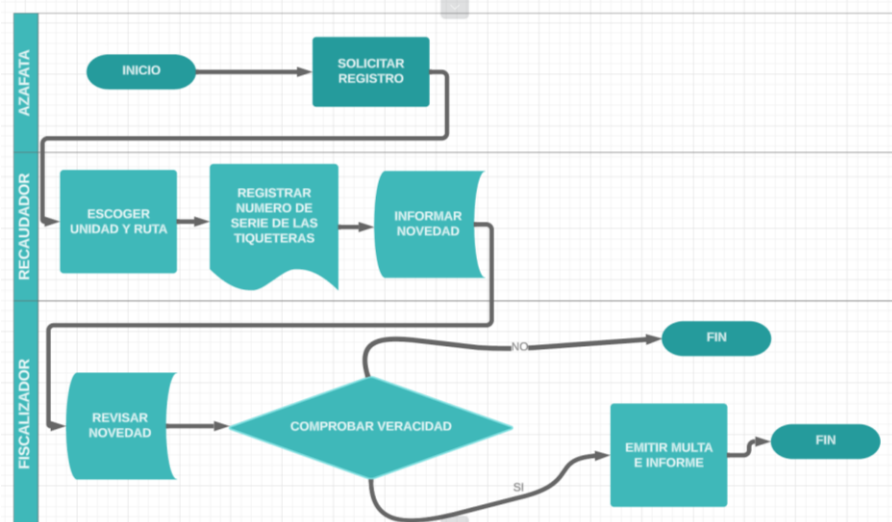
1.1 Diagnóstico de la situación actual

Actualmente el proceso se lleva de manera manual y en papel, por lo tanto, con muchas falencias y errores en el procedimiento de control y registro de especies vendidas que, de continuar llevándose como en la actualidad no presentaría cambio o beneficio alguno

La siguiente figura muestra el diagrama de cómo se lleva el proceso hasta la fecha

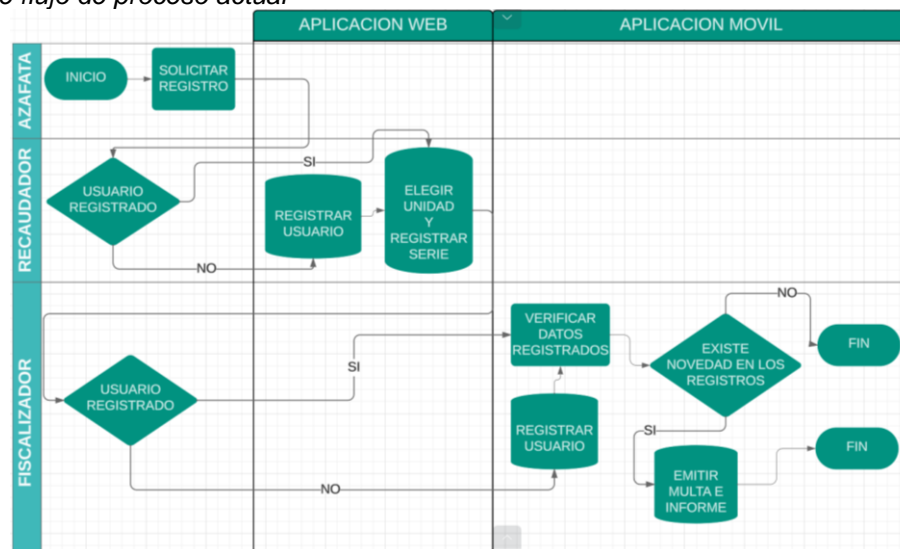
Figura 1

Diagrama de flujo de proceso manual



Nota. La figura representa el flujo del proceso, el cual se lleva de manera manual

La siguiente figura muestra el diagrama de cómo se llevará el proceso luego de la implementación del sistema

Figura 2*Diagrama de flujo de proceso actual*

Nota. La figura representa el flujo del proceso luego del despliegue de la aplicación

Por este motivo y premeditando los cambios y beneficios de la propuesta planteada se ha llegado a la conclusión de cambiar el paradigma de control que se llevaba por uno más tecnológico y actual que se adapte a las tecnologías de vanguardia

1.2 Recopilación de información

Para llevar a cabo la aplicación de los resultados de las investigaciones, se ha concluido en realizar 67 encuestas a los usuarios del sistema de transporte de la ciudad que se pueden visualizar en el anexo 1

De los resultados obtenidos se puede determinar la falta de control sobre las ventas de los boletos y la falta de un sistema que se encargue de administrar estos inconvenientes

1.3 Ámbito del software

Al sistema en general se lo denominará “Repórtalo” debido a que se utilizará para reportar alguna novedad en el registro de las unidades

Permitirá a los usuarios registrarse en el sistema con sus datos personales, permitiendo así, poder interactuar con una interfaz intuitiva y de fácil uso, además, esta

implementación proveerá de un control, así como también la generación de reportes para tener un control eficiente en los recursos utilizados, detallados de la siguiente manera

Se podrá registrar las series de las tiqueteras actuales y el usuario podrá hacer una comparación de los boletos vendidos en relación a los usuarios que desembarcan, además de que el sistema proveerá la hora y fecha actuales, esto permitirá optimizar el proceso emitiendo esto último para que el registro sea lo más rápido y eficaz posible

El usuario de la aplicación móvil, es decir, los compañeros de fiscalización podrán monitorear los datos registrados y así evitar movilizarse al lugar de registro para poder dar un fácil seguimiento y control de los registros

De haber alguna novedad en los registros, el usuario de la aplicación móvil podrá registrar la respectiva multa en el sistema debidamente comprobada para tener un listado de las unidades que han cometido alguna infracción, esto permitirá tener un control de unidades que son reincidentes para poder tomar las debidas decisiones pertinentes

Actualmente el procedimiento solo involucra el seguimiento de las unidades para impedir que las unidades se abstengan de entregar el boleto al usuario por lo que la característica de las ventas no se incluye, esto porque dicho procedimiento está asignado por otra persona ajena a lo que concierne el proceso de registro y seguimiento

1.4 Funciones del producto

La función principal de la aplicación será monitorizar los registros ingresados por medio del módulo web por el personal encargado del proceso, en este caso los fiscalizadores, así como los encargados del registro serán los recaudadores

A continuación, se muestran las historias de usuarios que se usaran para desarrollar el software

Tabla 1*Historias de Usuario Gestión de Seguridad*

HISTORIA DE USUARIO

Número: 1**Nombre:** Gestión de Seguridad**Usuario:** Recaudador y Fiscalizador**Riesgo en Desarrollo:** Alta**Prioridad en negocio:** Alta**Iteración asignada:** 2

Descripción: El usuario podrá iniciar sesión y crear su cuenta con dirección de correo electrónico y contraseña

Observación: En el caso de que la autenticación sea fallida debe mostrar un mensaje indicando que los datos ingresados son incorrectos.

Nota: La tabla muestra la historia de usuario para el ingreso al sistema.

Tabla 2*Historias de Usuario Gestión de Cuenta*

HISTORIA DE USUARIO

Número: 2**Nombre:** Gestión de Cuenta**Usuario:** Recaudador y Fiscalizador**Riesgo en Desarrollo:** Medio**Prioridad en negocio:** Alta**Iteración asignada:** 1

Descripción: El usuario podrá modificar su información personal y eliminar su cuenta si lo requiere.

Observación: El usuario podrá modificar toda su información excepto su correo electrónico

Nota: La tabla muestra los requerimientos para la gestión de cuenta del usuario.

Tabla 3*Historias de Usuario Gestión de empleos*

HISTORIA DE USUARIO

Número: 3

Nombre: Gestión de Registros

Usuario: Recaudador.

Riesgo en Desarrollo: Alta

Prioridad en negocio: Alta

Iteración asignada: 1

Descripción: El usuario podrá ingresar los datos necesarios para cada registro

Observación: La fecha y hora se registrarán de manera automática

Nota: La tabla muestra el ingreso de los datos

Tabla 4

Historias de Usuario Gestión de empleos

HISTORIA DE USUARIO

Número: 4

Nombre: Gestión de Registro

Usuario: Recaudador y Fiscalizador

Riesgo en Desarrollo: Alta

Prioridad en negocio: Alta

Iteración asignada: 1

Descripción: El usuario podrá visualizar los datos ingresados de manera organizada

Observación: Se visualizarán los datos ordenados por unidad y rutas

Nota: La tabla muestra la visualización de los datos, organizados

Tabla 5

Historias de Usuario Gestión de Reportes

HISTORIA DE USUARIO

Número: 5

Nombre: Gestión de Reportes

Usuario: Administrador

Riesgo en Desarrollo: Alta

Prioridad en negocio: Alta

Iteración asignada: 1

Descripción: El usuario podrá visualizar reportes de los usuarios del sistema, y lista de unidades que trabajaron en el día

Observación: Ninguno.

Nota: La tabla muestra los requerimientos de administración de reportes.

Tabla 6

Historias de Usuario Gestión de Reportes

HISTORIA DE USUARIO

Número: 6 **Nombre:** Gestión de Sistema

Usuario: Administrador **Riesgo en Desarrollo:** Alta

Prioridad en negocio: Alta **Iteración asignada:** 1

Descripción: El usuario puede modificar, agregar o eliminar usuarios, además que puede bloquear o eliminar cuentas de usuario.

Observación: Ninguno.

Nota: La tabla muestra los requerimientos de administración del sistema.

1.5 Características de los usuarios del sistema

Para el sistema de registro se ha empleado dos tipos de usuarios, mismos que se detallan en la siguiente tabla

Tabla 7

Perfiles de usuarios para el sistema.

Nombre de Usuario	Tipo de Usuario	Área Funcional	Actividad
-------------------	-----------------	----------------	-----------

Administrador	Administrador del Sistema	Administración	Administración de accesos Creación de reportes
Recaudador	Usuario	Registro	Ingreso de datos de rutas y unidades Registro diario Visualización de datos
Fiscalizador	Usuario	Seguimiento	Seguimiento y control de unidades Visualización de datos Emisión de multas

Nota: La tabla muestra las actividades principales de los roles de los usuarios.

1.6 Requisitos

Los requisitos funcionales y no funcionales que se solicitará para el desarrollo de la aplicación se detallan a continuación

1.5.1 Funcionales.

RF01: Solo los usuarios delegados en los dos perfiles podrán acceder al sistema

RF02: Los usuarios administradores también podrán ingresar datos al sistema

RF03: Los usuarios administradores podrán emitir reportes mensuales por ruta

RF04: Los usuarios administradores podrán tener el control de agregar o denegar el acceso a otros usuarios

RF05: El usuario administrador podrá hacer modificaciones en los registros

RF06: El usuario administrador podrá visualizar a los usuarios del sistema

RF07: El usuario recaudador podrá ingresar los datos de cada unidad, organizándola

RF08: El usuario recaudador no podrá modificar los datos de las series una vez ingresados

RF09: El usuario fiscalizador podrá ingresar a la ampliación móvil con sus credenciales ya asignadas

RF10: El usuario fiscalizador podrá visualizar los datos registrados por el usuario recaudador

RF11: El usuario fiscalizador no podrá modificar los datos de los registros

RF12: El usuario fiscalizador podrá registrar una multa si el caso lo requiere

1.5.2 No funcionales.

RNF01: El ingreso al sistema estará regulado por los usuarios administradores por lo que no cualquier persona tendrá acceso al mismo

RNF02: El sistema debe garantizar acceso y disponibilidad en todo momento

RNF03: El sistema debe garantizar fiabilidad y recuperación ante cualquier fallo

RNF04: EL sistema debe garantizar una disponibilidad mínima del 99%

RNF05: El ingreso debe contar con un procedimiento fácil de usar y ser capaz de ser aprendido en una corta capacitación

RNF06: El sistema debe garantizar la integridad de los datos almacenados

RNF07: El sistema debe escalar para manejar un aumento en la cantidad de registros sin afectar el rendimiento del mismo

RNF08: El sistema debe garantizar la usabilidad adecuada para que no tenga cargas pesadas al iniciar

CAPITULO 2. RESULTADOS

2.1 Metodología

La metodología seleccionada para la gestión y desarrollo de este proyecto fue KANBAN. KANBAN es una metodología ágil que se basa en la visualización del trabajo, permitiendo una gestión eficiente de tareas y una adaptación continua a los cambios en los requisitos del proyecto. A continuación, se describen los principales elementos y fases de la aplicación de KANBAN en el desarrollo del prototipo de aplicación web y móvil para el registro y control de rutas alimentadoras de la Empresa Pública Metropolitana de Pasajeros de Quito (EPMTPQ):

Tablero KANBAN:

Se estableció un tablero KANBAN que sirvió como herramienta central para visualizar y gestionar las tareas del proyecto. Este tablero se dividió en columnas que representaban diferentes etapas del proceso, como "To Do", "In Progress", "Review", y "Done".

Tarjetas de Tareas:

Cada tarea o funcionalidad a implementar se representó mediante tarjetas en el tablero KANBAN. Estas tarjetas contenían información detallada sobre la tarea, incluyendo descripción, prioridad, y asignación a un miembro del equipo.

Priorización y Asignación:

Las tarjetas se priorizaron según la importancia y la urgencia de la tarea. Los miembros del equipo asignaron tarjetas a sí mismos de acuerdo con su especialización y capacidad de trabajo, asegurando una distribución equitativa de tareas.

Iteraciones Continuas:

La metodología KANBAN permitió iteraciones continuas y entregas incrementales. A medida que las tarjetas se movían a través de las columnas del tablero, se

completaban tareas y se incorporaban nuevas funcionalidades, lo que permitía una adaptación constante a los cambios en los requerimientos.

Reuniones Regulares de Revisión:

Se llevaron a cabo reuniones regulares de revisión del tablero KANBAN para evaluar el progreso del proyecto, identificar posibles obstáculos y discutir ajustes necesarios en la planificación. Estas reuniones fomentaron la comunicación efectiva y la resolución rápida de problemas.

Mejora Continua:

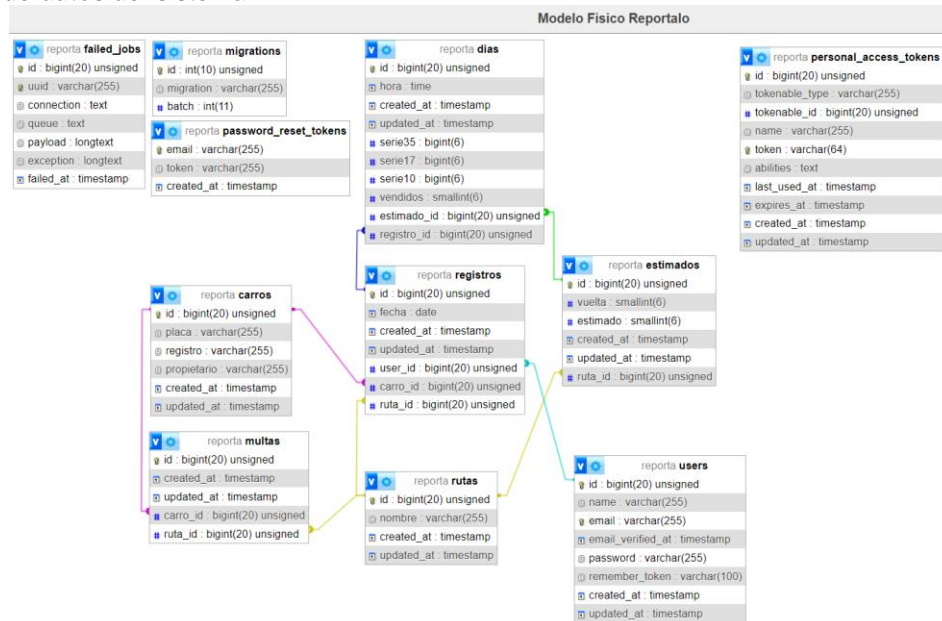
La metodología KANBAN se centró en la mejora continua. Se alentó al equipo a reflexionar sobre su desempeño, identificar áreas de oportunidad y proponer mejoras en los procesos. Esto permitió optimizar la eficiencia y la calidad del trabajo realizado.

La elección de KANBAN como metodología de desarrollo proporcionó flexibilidad y agilidad, permitiendo adaptarse a los cambios en los requerimientos y garantizando entregas incrementales y consistentes a lo largo del proyecto. Esta metodología facilitó la colaboración eficiente entre los miembros del equipo y contribuyó al éxito en la consecución de los objetivos establecidos para el desarrollo de la aplicación web-móvil.

2.2 Diseño general

Se ha tomado en cuenta las fases del proceso y requerimientos para poder diseñar el modelo físico de la base de datos detallado a continuación

Figura 3.
Base de datos del sistema

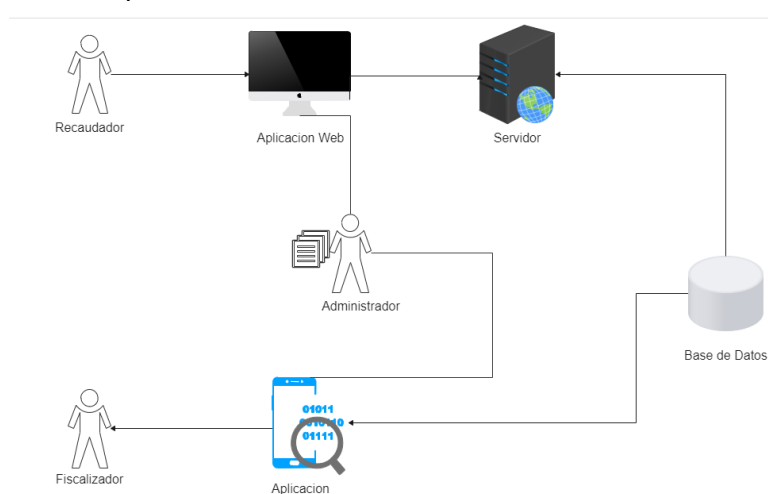


Nota. La figura representa el modelo físico del bade de datos

2.2.1 Diagrama de la arquitectura del sistema

Para poder desarrollar la solución se tomaron en cuenta varios aspectos como permisos de usuario y quien puede o no manipular los datos, esto con el motivo de garantizar la integridad de los datos, así como el procedimiento de control y se llegó a la conclusión del siguiente diagrama

Figura 4.
Diagrama de arquitectura del sistema



Nota. La figura representa el diagrama de la arquitectura del sistema

2.2 Diseño de interfaces

Para el desarrollo de la interfaz web se buscó un diseño intuitivo para poder llevar a cabo el proceso de manera que existan la menor cantidad de dificultades al momento de registrar correctamente los datos

Figura 5.

Mockup 1: Registro del usuario

Registro

Nombre

Correo

Celula

Cargo

Telefono

[Ya tienes cuenta? Ingrese aqui](#)

Nota: Solicitud de los datos al usuario

Figura 6.

Mockup 2: Ingreso del usuario

Ingreso

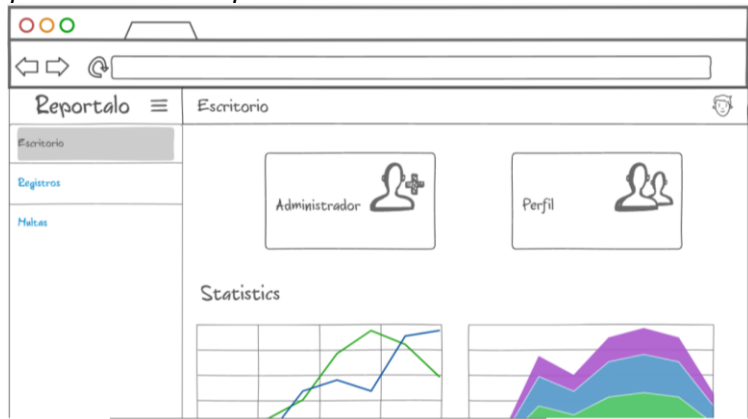
Correo

Contraseña

☒ Recordarme

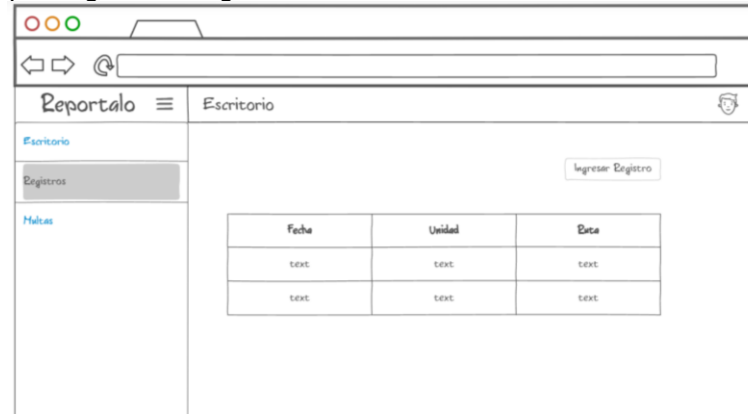
Nota: Ingreso del usuario al sistema

Figura 7.
Mockup 3: Pantalla Principal



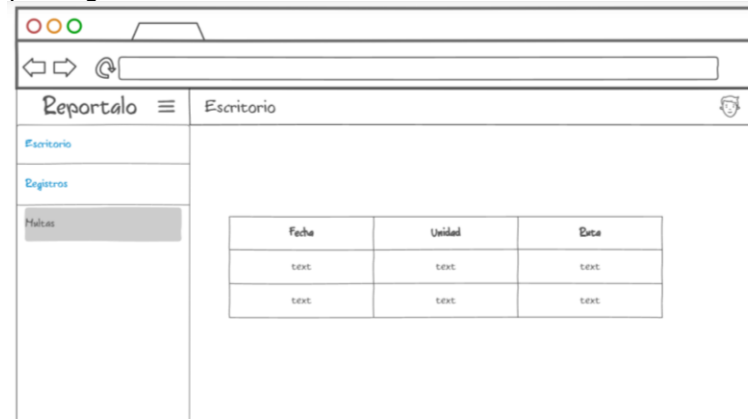
Nota: Pantalla principal

Figura 8.
Mockup 4: Ingreso de registros



Nota: Ingreso de datos

Figura 9.
Mockup 5: Ingreso de multas



Nota: Ingreso de multas

Para el desarrollo de la móvil se tomaron en cuenta los procedimientos del cargo de fiscalizador y se diseñó la interfaz de acuerdo a las actividades llevadas por el mismo

Figura 10.

Mockup 6: Registro del usuario



A mobile app mockup for a registration screen. The status bar at the top shows the time 10:21 and various icons. The screen title is 'Registro'. Below the title are three input fields: 'Name' with placeholder text 'type your username' and a person icon, 'Email' with placeholder text 'type your email' and an envelope icon, and 'Password' with placeholder text '*****' and a lock icon. Below these fields is a 'Sign Up' button. Under the button is the word 'or' followed by a blue 'Login' link.

Nota: Solicitud de los datos al usuario

Figura 11.

Mockup 7: Ingreso del usuario



A mobile app mockup for a login screen. The status bar at the top shows the time 10:21 and various icons. The screen title is 'Bienvenido a Reportalo'. Below the title are two input fields: 'Usuario' with placeholder text 'ingrese su correo' and a person icon, and 'Contraseña' with placeholder text 'ingrese su contraseña' and a lock icon. Below these fields are two blue links: 'Registro' and 'Olvide mi contraseña'. At the bottom is an 'Ingresar' button.

Nota: Ingreso del usuario al sistema

Figura 12.

Mockup 8: Contraseña olvidada



Nota: Muestra como restablecer la contraseña

Figura 13.

Mockup 9: Registros Actuales



Nota: Muestra los registros del día actual

Figura 14.
Mockup 10: Multas



Nota: Muestra las multas ingresadas

2.3 Estándares de programación utilizados

Establecer estándares de programación en Laravel es una práctica recomendada para mejorar la legibilidad del código, la consistencia y la colaboración en equipos de desarrollo.

Tabla 8
Estándares de programación en laravel

Estándares de programación	
Aspecto del Código	Estándar de programación
Convenciones de nombres	Utiliza camelCase para variables y funciones, y Pascal Case para clases y modelos. Ejemplo: \$miVariable, MiClase, miFuncion().
Rutas	Organiza las rutas de manera lógica en el archivo web.php o api.php. Utiliza el método adecuado (get, post, etc.) y elige nombres descriptivos para las rutas.
Controladores	

Modelos	Utiliza un controlador por recurso y sigue la convención de nombres. Por ejemplo, para un recurso "Post", el controlador puede llamarse PostController.
	Sigue la convención de nombres y utiliza relaciones para definir asociaciones entre modelos. Por ejemplo, utiliza hasMany, belongsTo, etc.
Migraciones	Mantén las migraciones coherentes y describe claramente los cambios en la estructura de la base de datos. Utiliza nombres descriptivos para las tablas y las columnas.
Vistas	Divide las vistas en secciones lógicas utilizando layouts y partials. Mantén la lógica de presentación fuera de las vistas tanto como sea posible.
Validaciones	Valida las entradas del usuario utilizando las herramientas proporcionadas por Laravel, como las reglas de validación y los Requests.
Comentarios	Documenta tu código con comentarios descriptivos cuando sea necesario, pero evita comentarios obvios o redundantes. Mantén el código auto explicativo siempre que sea posible.
Seguridad	Implementa medidas de seguridad, como protección contra ataques CSRF, sanitización de entrada, y utiliza las herramientas de Laravel para evitar problemas comunes de seguridad.

Nota: La tabla muestra los estándares utilizados para las buenas prácticas de programación en laravel

2.4 Pruebas

Las pruebas unitarias son un tipo de prueba en el desarrollo de software que se centra en verificar el correcto funcionamiento de unidades individuales de código, como funciones, métodos o clases. El objetivo de las pruebas unitarias es asegurarse de que cada componente o unidad del software funcione de acuerdo con las especificaciones y se comporte de acuerdo con lo esperado.

Enfoque en Componentes Individuales: Las pruebas unitarias se centran en aislar y evaluar partes específicas del código, verificando su comportamiento de manera independiente. Esto ayuda a identificar y corregir errores en las unidades individuales antes de integrarlas en el sistema completo.

Automatización: Las pruebas unitarias suelen ser automatizadas, lo que significa que se crean scripts o casos de prueba automatizados que pueden ejecutarse de manera repetida cada vez que se realizan cambios en el código. La automatización facilita la ejecución rápida y frecuente de las pruebas, lo que es crucial en entornos ágiles y de desarrollo continuo.

Rapidez y Eficiencia: Dado que las pruebas unitarias se centran en porciones pequeñas de código, su ejecución es rápida y eficiente. Esto permite a los desarrolladores recibir retroalimentación inmediata sobre la validez de su código y facilita la detección temprana de posibles problemas.

Facilitan la Identificación de Errores: Al probar unidades individuales, las pruebas unitarias facilitan la identificación y corrección temprana de errores antes de que se propaguen a otras partes del sistema.

Aseguran la Mantenibilidad: Al proporcionar un conjunto de pruebas unitarias, los desarrolladores pueden realizar cambios en el código con confianza, sabiendo que, si las pruebas siguen pasando, las unidades afectadas aún funcionan según lo esperado.

Integración Continua: Las pruebas unitarias son esenciales para implementar prácticas de integración continua, donde se integran y prueban cambios en el código de

manera constante. Esto contribuye a la detección temprana de problemas y a la entrega de software más confiable.

2.4.1 Pruebas Unitarias:

Laravel organiza las pruebas en el directorio test de tu aplicación. Las pruebas unitarias se almacenan en la carpeta Unit, las pruebas de integración en la carpeta Feature, y las pruebas de aceptación en la carpeta Browser.

Figura 15.
Código de pruebas

```
use App\Models\Registro;
use App\Models\User;
use Livewire\Livewire;
use Tests\TestCase;
use App\Filament\Resources\RegistroResource;

class PruebaRegistros extends TestCase
{
    public function testRegistro()
    {
        $resource = new RegistroResource();

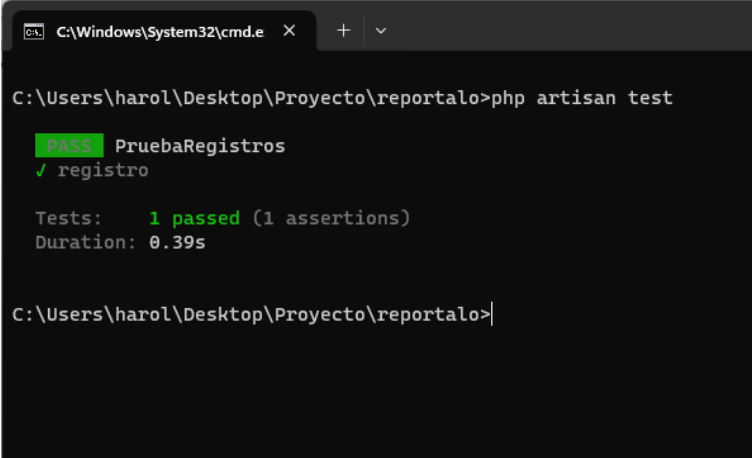
        $this->assertInstanceOf(RegistroResource::class, $resource);
    } // Otras pruebas para métodos y propiedades específicos del recurso
}
```

Nota: Muestra el código de la prueba unitaria

Ejecutar Pruebas:

Utiliza el comando `php artisan test` para ejecutar todas las pruebas. Puedes especificar un archivo o directorio específico si solo deseas ejecutar ciertas pruebas.

Figura 16.
Resultado de prueba 1



```
C:\Windows\System32\cmd.e x + v

C:\Users\harol\Desktop\Proyecto\reportalo>php artisan test

PASS PruebaRegistros
✓ registro

Tests: 1 passed (1 assertions)
Duration: 0.39s

C:\Users\harol\Desktop\Proyecto\reportalo>
```

Nota: Muestra el resultado de la prueba exitosa N°1

Al realizar el procedimiento de registro y control se mejora la integridad y disponibilidad de los mismos ya que el proceso se llevaba de manera manual y en papel con anterioridad, esto implementa un margen de mejora y escalabilidad ya que los datos están disponibles en todo momento y al poderlos visualizar en la aplicación móvil se facilita el control de más unidades en relación al tiempo de verificación en el puesto de trabajo de registro

2.4.2 Pruebas de Estrés o Carga

Las pruebas de estrés y carga fueron una parte fundamental del proceso de calidad del prototipo de la aplicación web-móvil desarrollada para el registro y control de rutas alimentadoras de la Empresa Pública Metropolitana de Pasajeros de Quito (EPMPQ). Estas pruebas se realizaron con el objetivo de evaluar el rendimiento y la capacidad de respuesta del sistema bajo condiciones extremas de carga, identificando posibles puntos de fallo y asegurando que la aplicación sea capaz de manejar la demanda esperada. A continuación, se describen los principales aspectos de las pruebas de estrés y carga:

Objetivos de las Pruebas

- Evaluar la estabilidad del sistema bajo cargas de trabajo extremas.

- Identificar cuellos de botella y posibles puntos de fallo.
- Determinar la capacidad máxima de usuarios concurrentes que el sistema puede manejar sin degradación significativa del rendimiento.

Escenarios de Carga

Se diseñaron diversos escenarios de carga que simularan situaciones de uso intensivo. Esto incluyó la simulación de múltiples usuarios concurrentes realizando diversas acciones como registro de rutas, consultas simultáneas y actualizaciones en tiempo real.

Herramientas Utilizadas

Se emplearon herramientas especializadas para la ejecución de las pruebas de carga, como Apache JMeter y la herramienta predeterminada de laravel. Estas herramientas permitieron simular patrones de tráfico realista y medir la respuesta del sistema en condiciones de carga variable.

Variables Evaluadas

Durante las pruebas, se evaluaron diversas variables, incluyendo el tiempo de respuesta del servidor, el rendimiento de la base de datos, la utilización de recursos del servidor (CPU, memoria, etc.), y la capacidad de recuperación después de picos de carga.

Figura 17.
Prueba de stress: Usuarios recurrentes

Sample #	Start Time	Thread Name	Label	Sample Time (ms)	Status	Bytes	Sent Bytes	Latency	Connect Time (ms)
41	09:53:27.463	Modulo Princip...	Ingreso de usua...	6470	Success	36540	131	6469	1
42	09:53:27.496	Modulo Princip...	Ingreso de usua...	7117	Success	36540	131	7116	1
43	09:53:27.529	Modulo Princip...	Ingreso de usua...	7771	Success	36540	131	7769	0
44	09:53:27.562	Modulo Princip...	Ingreso de usua...	8512	Success	36540	131	8510	1
45	09:53:27.595	Modulo Princip...	Ingreso de usua...	9236	Success	36540	131	9234	1
46	09:53:27.627	Modulo Princip...	Ingreso de usua...	9895	Success	36540	131	9891	0
47	09:53:27.661	Modulo Princip...	Ingreso de usua...	10640	Success	36540	131	10638	0
48	09:53:27.694	Modulo Princip...	Ingreso de usua...	11223	Success	36540	131	11222	1
49	09:53:27.726	Modulo Princip...	Ingreso de usua...	11867	Success	36540	131	11864	0
50	09:53:27.760	Modulo Princip...	Ingreso de usua...	12528	Success	36540	131	12527	1
51	09:53:27.792	Modulo Princip...	Ingreso de usua...	13158	Success	36540	131	13157	1
52	09:53:27.825	Modulo Princip...	Ingreso de usua...	13783	Success	36540	131	13782	1
53	09:53:27.859	Modulo Princip...	Ingreso de usua...	14357	Success	36540	131	14355	1
54	09:53:27.892	Modulo Princip...	Ingreso de usua...	14904	Success	36540	131	14902	0
55	09:53:27.924	Modulo Princip...	Ingreso de usua...	15453	Success	36540	131	15454	1
56	09:53:27.957	Modulo Princip...	Ingreso de usua...	15993	Success	36540	131	15991	1
57	09:53:27.991	Modulo Princip...	Ingreso de usua...	16536	Success	36540	131	16534	0
58	09:53:28.024	Modulo Princip...	Ingreso de usua...	17077	Success	36540	131	17076	1
59	09:53:28.057	Modulo Princip...	Ingreso de usua...	17625	Success	36540	131	17623	1
60	09:53:28.091	Modulo Princip...	Ingreso de usua...	18179	Success	36540	131	18178	1

Nota: Solicitud de usuarios al servidor

Figura 18.
Prueba de stress: Registros recurrentes

Sample #	Start Time	Thread Name	Label	Sample Time (ms)	Status	Bytes	Sent Bytes	Latency	Connect Time (ms)
81	10:09:39.212	Modulo Princip...	Registros diarios	104090	Success	37663	266	47211	1
82	10:09:39.222	Modulo Princip...	Registros diarios	104643	Success	37663	266	47685	1
83	10:09:39.232	Modulo Princip...	Registros diarios	105194	Success	37663	266	48159	1
84	10:09:39.243	Modulo Princip...	Registros diarios	105789	Success	37663	266	48637	1
85	10:09:39.253	Modulo Princip...	Registros diarios	106406	Success	37663	266	49112	1
86	10:09:39.263	Modulo Princip...	Registros diarios	107029	Success	37663	266	49590	1
87	10:09:39.272	Modulo Princip...	Registros diarios	107641	Success	37663	266	50101	1
88	10:09:39.282	Modulo Princip...	Registros diarios	108215	Success	37663	266	50585	1
89	10:09:39.292	Modulo Princip...	Registros diarios	108814	Success	37663	266	51086	0
90	10:09:39.302	Modulo Princip...	Registros diarios	109449	Success	37663	266	51580	1
91	10:09:39.312	Modulo Princip...	Registros diarios	110265	Success	37663	266	52224	0
92	10:09:39.323	Modulo Princip...	Registros diarios	111105	Success	37663	266	52773	1
93	10:09:39.333	Modulo Princip...	Registros diarios	111813	Success	37663	266	53362	1
94	10:09:39.343	Modulo Princip...	Registros diarios	112526	Success	37663	266	53937	0
95	10:09:39.352	Modulo Princip...	Registros diarios	113231	Success	37663	266	54516	1
96	10:09:39.362	Modulo Princip...	Registros diarios	114002	Success	37663	266	54692	1
97	10:09:39.373	Modulo Princip...	Registros diarios	114734	Success	37663	266	55169	0
98	10:09:39.382	Modulo Princip...	Registros diarios	115309	Success	37663	266	55673	1
99	10:09:39.393	Modulo Princip...	Registros diarios	115851	Success	37663	266	56142	1
100	10:09:39.403	Modulo Princip...	Registros diarios	116391	Success	37663	266	56621	1

Nota: Registros realizados

Análisis de Resultados

Los resultados de las pruebas de carga fueron analizados meticulosamente para identificar cualquier degradación del rendimiento, cuellos de botella o posibles puntos de fallo. Se prestaron especial atención a los límites de capacidad del sistema y a la capacidad de recuperación después de situaciones de carga extrema.

2.5 Implementación

Para poder implementar la solución es necesario de una computadora para el proceso de registro y un teléfono móvil ya sea Android o IOS, así como también conexión a internet y los respectivos periféricos de entrada para registrar los datos

2.6 Requerimientos de hardware y software

2.6.1 Software para servidor

- SGBD con su respectiva versión
- Servidor Web
- Servidor de aplicaciones

2.6.2 Software para usuario

- Sistema Operativo Windows 7 o superior
- 8GB RAM
- Procesador i5
- Navegador web (Chrome, Firefox, I-Explorer)
- 500gb de memoria en disco

2.6.3 Hardware para servidor

- RAM al menos de 8Gb
- Procesador Core I7
- Tarjeta de Red 1Gbps

2.6.4 Hardware para usuario

- Teléfono Android o IOS
- Conexión a internet
- Computadora

CONCLUSIONES

Ambos proyectos lograron alcanzar los objetivos específicos establecidos, proporcionando soluciones eficientes y funcionales para el registro y control de las rutas alimentadoras. La elección de Laravel para el desarrollo web y Xamarin para el desarrollo móvil demostró ser acertada, considerando la flexibilidad, escalabilidad y eficiencia en la implementación de ambos frameworks. La aplicación resultante es un prototipo integral que satisface las necesidades de la Empresa Pública Metropolitana de Pasajeros de Quito (EPMTPQ) en cuanto al seguimiento y gestión de las rutas alimentadoras.

- **Éxito en el Levantamiento de Requerimientos:**

Ambos equipos de desarrollo lograron un levantamiento completo y detallado de los requerimientos para la aplicación web y móvil. Esta fase crítica estableció una base sólida para el diseño y la implementación, garantizando que las funcionalidades esenciales fueran consideradas y abordadas de manera efectiva.

- **Integración Coherente del Modelo de Datos:**

La coherencia en el diseño del modelo de datos para la aplicación web y móvil permitió una integración fluida entre ambas plataformas. La consistencia en la estructura de datos facilitó el intercambio de información, asegurando que los registros y controles de rutas alimentadoras fueran precisos y sincronizados entre los sistemas.

- **Eficiencia en la Codificación con Metodología Ágil KANBAN:**

La implementación de la metodología ágil KANBAN demostró ser eficiente en la gestión de tareas y cambios en los requerimientos. Tanto en el desarrollo web con Laravel como en el desarrollo móvil con Xamarin, la metodología ágil facilitó la adaptación a las necesidades cambiantes del proyecto, asegurando un progreso constante y entregas oportunas.

- **Aprovechamiento de Fortalezas de Laravel y Xamarin:**

La elección de Laravel para el desarrollo web y Xamarin para el desarrollo móvil resultó acertada. Laravel proporcionó un marco sólido, estructurado y seguro para la aplicación web, mientras que Xamarin ofreció una solución eficiente para el desarrollo móvil multiplataforma. Ambos frameworks contribuyeron a la creación de un prototipo integral, cumpliendo con los estándares de calidad y rendimiento requeridos por la EPMT PQ.

RECOMENDACIONES

Se recomienda realizar una investigación detallada en el área de optimización de rutas y eficiencia operativa dentro del contexto de la Empresa Pública Metropolitana de Pasajeros de Quito (EPMTPQ). Este tipo de investigación podría abordar estrategias avanzadas de planificación de rutas, utilizando algoritmos específicos para minimizar tiempos de viaje, reducir costos operativos y mejorar la experiencia del usuario. Un enfoque en la eficiencia operativa puede resultar en beneficios significativos para la empresa y los usuarios del sistema de transporte público.

- **Monitoreo Proactivo de Cambios en Requerimientos:**

Dado el éxito en el levantamiento inicial de requerimientos, se sugiere establecer un sistema proactivo de monitoreo de cambios en las necesidades de la EPMTPQ. Esto puede incluir reuniones regulares con los usuarios finales y la implementación de herramientas de seguimiento de requisitos.

- **Enfoque Incremental en la Seguridad:**

Aunque se implementaron medidas de seguridad efectivas, se recomienda adoptar un enfoque incremental, evaluando y mejorando constantemente las prácticas de seguridad. La ciberseguridad es un campo en constante evolución, y mantenerse actualizado es esencial para garantizar la protección de la aplicación.

- **Implementación de un Programa de Capacitación Continua:**

Con el objetivo de asegurar un uso eficiente de la aplicación, se recomienda implementar un programa de capacitación continua para el personal de la EPMTPQ. Esto podría incluir sesiones de formación periódicas y material de referencia actualizado.

- **Exploración de Tecnologías Emergentes:**

Aprovechando el éxito con Laravel y Xamarin, se sugiere mantener un ojo en tecnologías emergentes que puedan mejorar aún más el rendimiento, la eficiencia o la experiencia del usuario. La exploración constante de nuevas opciones garantizará que el proyecto se beneficie de las últimas innovaciones tecnológicas

REFERENCIAS BIBLIOGRÁFICAS

Blog Redacción. (19 de mayo de 2022). ¿Cómo Realizar el Cálculo del Tamaño de la Muestra Online Para tu Investigación de Mercado? Obtenido de <https://www.rochiconsulting.com/blog/calculo-del-tamano-de-la-muestra-online/>

Cadavid, A. N., Martínez, J. D. F., & Vélez, J. M. (2013). Revisión de metodologías ágiles para el desarrollo de software. *Prospectiva*, 11(2), 30-39.

Cíceri, M. (2019). *Introducción a Laravel: Aplicaciones robustas ya gran escala*. RedUsers.

Cobo, Á. (2005). *PHP y MySQL: Tecnología para el desarrollo de aplicaciones web*. Ediciones Díaz de Santos.

Gauchat, J. D. (2012). *El gran libro de HTML5, CSS3 y JavaScript*. Marcombo.

Gil, C. G. (2018). Objetivos de Desarrollo Sostenible (ODS): una revisión crítica. *Papeles de relaciones eco sociales y cambio global*, 140(1), 107-118.

Hinojosa, M. A. (2003). Diagrama de Gantt. *Producción, procesos y operaciones*, 48.

Rendón-Macías, M. E., Villasís-Keever, M. Á., & Miranda-Novales, M. G. (2016). Estadística descriptiva. *Revista Alergia México*, 63(4), 397-407.

Risso, A. (2002). Metodología de investigación de las www. *Metodología de las Ciencias del comportamiento*, 487-491.

Santillán, L. A. C., Ginestà, M. G., & Mora, Ó. P. (2014). Bases de datos en MySQL. *Universitat oberta de Catalunya*.

ANEXOS

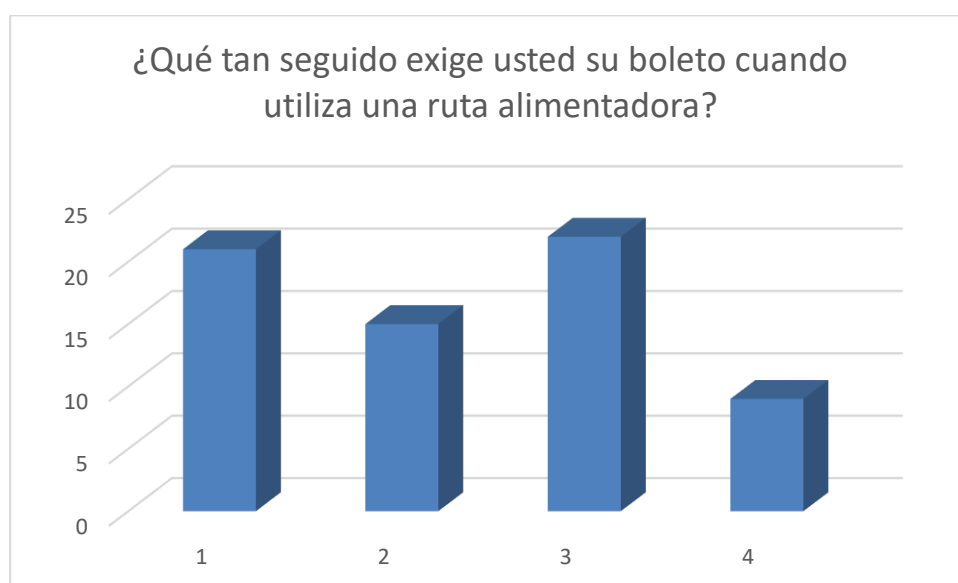
¿Qué tan seguido exige usted su boleto cuando utiliza una ruta alimentadora?

1-Nunca

2-Pocas veces

3-Seguido

4-Siempre



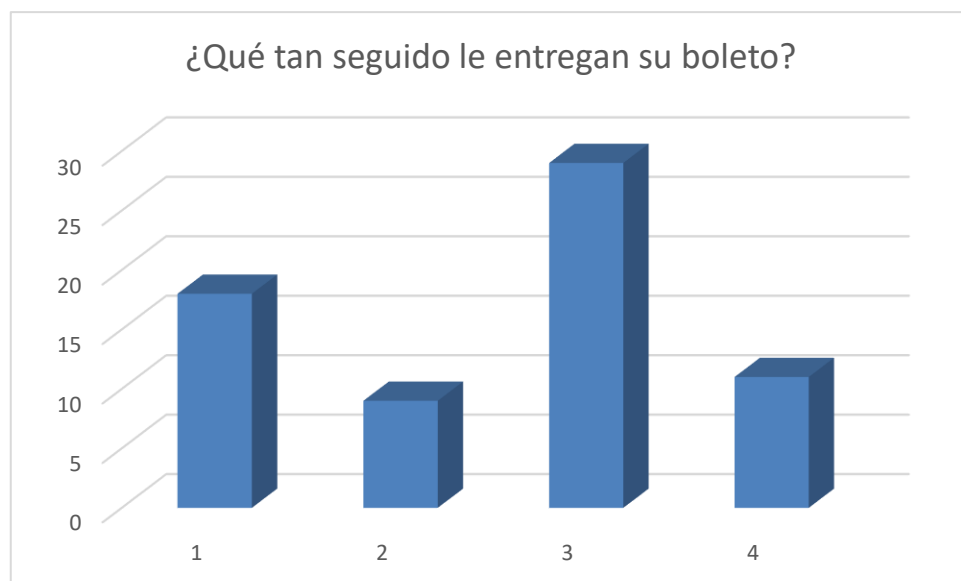
¿Qué tan seguido le entregan su boleto?

1-Nunca

2-Pocas veces

3-Seguido

4-Siempre



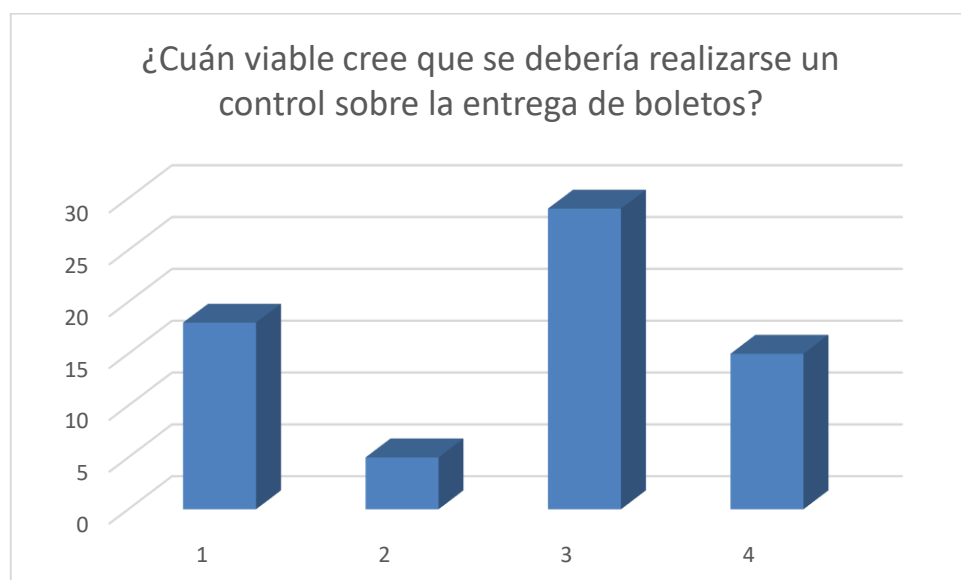
¿Cuán viable cree que se debería realizarse un control sobre la entrega de boletos?

1-Nunca

2-Pocas veces

3-Seguido

4-Siempre



¿Cuándo no le entregan su boleto usted acude a denunciarlo?

1-Nunca

2-Pocas veces

3-Seguido

4-Siempre



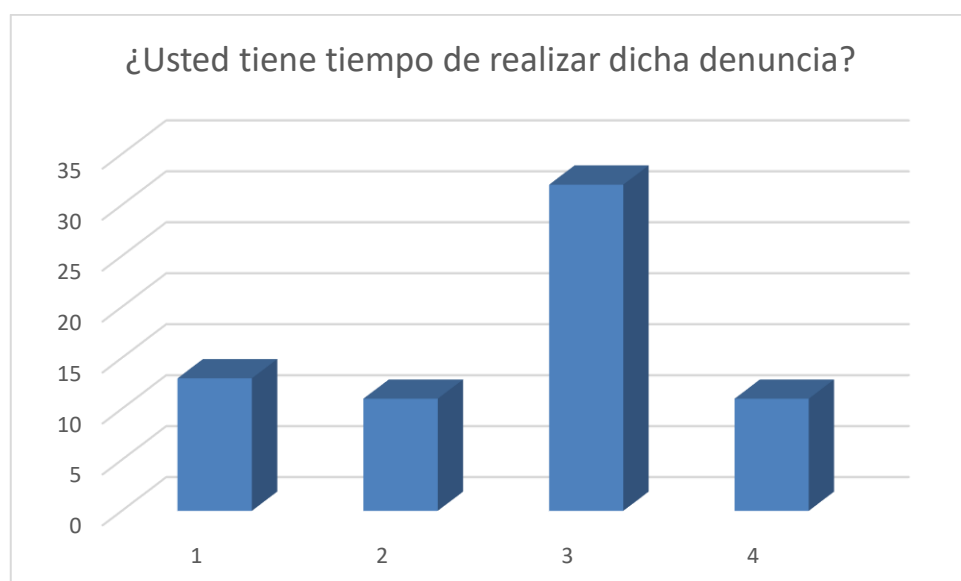
¿Usted tiene tiempo de realizar dicha denuncia?

1-Nunca

2-Pocas veces

3-Seguido

4-Siempre



Manual de Usuario

Objetivo del Manual

El manual de usuario tiene por objetivo minimizar el impacto del cambio de proceso para que el procedimiento se haga lo más fácil posible, además de controlar los componentes de las aplicaciones

Alcance del Programa

Las aplicaciones web y móvil tienen como fin automatizar un proceso que antes se lo llevaba de forma manual, el sistema web solo servirá para ingresar datos en el usuario de Recaudador y sacar reportes y emitir multas en usuario Administrador, así como la aplicación móvil tiene como fin el monitoreo y control

Audiencia

El presente programa va dirigido a los empleados de la empresa de transporte EPMT PQ que tienen relación con el proceso de control de ventas de boletos en las rutas alimentadoras

Instalación

Descarga del Programa

La descarga de la parte web será innecesaria debido a que basta con tener un navegador funcional en la computadora a utilizar

Por otra parte, el apartado móvil se instalará mediante archivo apk, el cual se deberá ejecutar en un teléfono inteligente

Inicio de Sesión

Creación de Cuenta

Ambos sistemas contarán con la característica de creación de cuenta, el cual solicitará correo y datos personales para llevar un control dentro del usuario del sistema

Ingreso al Sistema

El ingreso a ambos sistemas se lo llevará de manera típica con correo como usuario y la contraseña prevista previamente

Recuperación de Contraseña

La característica de recuperación de contraseña estará disponible en ambos sistemas, en el caso del sistema web se lo llevará mediante un servidor de correo para contribuir con la seguridad del sistema

Interfaz de Usuario

Navegación

Las interfaces de ambos sistemas están diseñadas para proveer una experiencia de fácil uso a los usuarios

Funcionalidades Principales

Función de Ingreso de datos

Esta función estará disponible en el sistema web que llevarán los recaudadores y tiene el fin de simplemente registrar los datos actuales de las series de las tiqueteras de las rutas alimentadoras

Pasos para Utilizarla

Para utilizar esta función bastará con ingresar la unidad y ruta en la que se encuentra ese día y luego las series de las tiqueteras, ya que la fecha y hora se registran automáticamente, esto optimizara el tiempo de registro con respecto a cómo se lo llevaba de forma manual

Consejos y Mejores Prácticas

Para poder utilizar de mejor manera esta característica se debe llevar un control de los usuarios para que tengan celeridad al momento de registrar los datos

Función de control y emisión de multas

Esta función está disponible para el apartado móvil en el que el usuario fiscalizador podrá llevar un control más personal y específico sobre las unidades registradas en el día, así como también se podrán emitir multas para optimizar el procedimiento

Pasos para Utilizarla

Para utilizar esta característica el usuario deberá registrarse en la aplicación web para poder comenzar a utilizarla

Consejos y Mejores Prácticas

Para llevar un mejor control en la aplicación web el usuario fiscalizador deberá tener un previo conocimiento en el proceso de control

Martinez

INFORME DE ORIGINALIDAD

9%	8%	0%	9%
INDICE DE SIMILITUD	FUENTES DE INTERNET	PUBLICACIONES	TRABAJOS DEL ESTUDIANTE

FUENTES PRIMARIAS

1	Submitted to Universidad Tecnológica Israel Trabajo del estudiante	8%
2	Submitted to Corporación Universitaria Minuto de Dios, UNIMINUTO Trabajo del estudiante	1%
3	repositorio.uisrael.edu.ec Fuente de Internet	1%

Excluir citas	Activo	Excluir coincidencias	< 1%
Excluir bibliografía	Activo		