



Universidad
Europea MADRID

PROYECTO INTEGRADOR

1DAM

RutaSinBaches



Rutas Sin Baches

Ignacio Bermejo Villar

Henry Raul Illescas Eras

Arpad Kiss

Mario Seoane García

Índice

Resumen.....	2
1. Introducción	4
2. Objetivos	6
3. Tecnologías utilizadas.....	8
4. Desarrollo e implementación	11
5. Metodología	18
6. Resultados y conclusiones.....	23
7. Trabajos futuros	25
Anexos	27
Anexo I – Listado de requisitos de la aplicación	27
Anexo II – Guía de uso de la aplicación	30

Resumen

Palabras clave: Colaboración; Equipo; Esfuerzo; Java; Herramientas.

El proyecto "Rutas Sin Baches" representa un esfuerzo colaborativo para abordar un problema crítico en el contexto urbano. Desarrollado por un equipo de cuatro integrantes, esta aplicación Java ofrece una solución accesible y eficiente para identificar y gestionar baches en las vías urbanas. Su creación involucró una combinación de metodologías ágiles, atención a la experiencia del usuario y un enfoque centrado en la fiabilidad del sistema.

Desde el inicio, el equipo se comprometió con el uso de **metodologías** ágiles para una colaboración efectiva y una entrega incremental de funcionalidades. Cada miembro del equipo desempeñó roles específicos, abarcando desde el desarrollo del backend y frontend hasta la gestión de la base de datos y la integración de servicios en la nube. Esta distribución de responsabilidades permitió un avance constante y una gestión eficaz del proyecto.

La aplicación permite a los usuarios reportar baches de manera intuitiva, contribuyendo así a una base de datos dinámica. La transparencia y la confianza se promovieron mediante la posibilidad de que los usuarios visualizaran el estado de las reparaciones y recibieran notificaciones sobre las acciones tomadas en las áreas reportadas.

El proyecto incluyó pruebas rigurosas para garantizar la estabilidad y seguridad de la aplicación. Se llevaron a cabo pruebas unitarias, de integración y de aceptación del usuario, junto con medidas de seguridad para proteger los datos personales y la integridad de la base de datos.

El desarrollo tecnológico se apoyó en una variedad de herramientas, desde IDEs como IntelliJ IDEA y Eclipse hasta plataformas de gestión de código como GitHub. La gestión de la base de datos se realizó con Oracle Database y posteriormente con phpMyAdmin en AWS, lo que proporcionó alta disponibilidad y escalabilidad. Se implementó el patrón de diseño Modelo Vista Controlador (MVC) para una estructura clara y mantenible del código.

En resumen, "Rutas Sin Baches" no solo representa un esfuerzo técnico para resolver problemas cotidianos, sino también un ejemplo de colaboración multidisciplinaria y aplicación efectiva de la tecnología para mejorar la calidad de vida en las ciudades.

1. Introducción

Para este apartado se recomienda una extensión de 1 a 2 páginas. Presentación del proyecto, contexto y detalles introductorios.

En el contexto actual de urbanización y crecimiento de las ciudades, la calidad de las infraestructuras viales es un factor crítico que afecta directamente la calidad de vida de los ciudadanos. Con el objetivo de solucionar estos problemas, se desarrolló la aplicación "Rutas Sin Baches", un proyecto integrador llevado a cabo por un equipo de cuatro integrantes.

"Rutas Sin Baches" fue concebida para ofrecer una solución eficiente y accesible a la identificación y gestión de baches en las vías urbanas. La aplicación, desarrollada en Java, permitía a los usuarios reportar baches mediante una interfaz intuitiva y fácil de usar. Al enviar un reporte, los usuarios contribuían a la creación de una base de datos dinámica que facilitaba a las autoridades locales y a las empresas de mantenimiento vial la priorización y reparación de las áreas más afectadas.

El desarrollo de esta aplicación involucró varios aspectos técnicos y organizativos. Desde el inicio del proyecto, el equipo se comprometió a aplicar metodologías ágiles para garantizar una colaboración efectiva y una entrega incremental de funcionalidades. Cada miembro del equipo asumió roles específicos, abarcando desde el desarrollo del backend y frontend hasta la gestión de la base de datos y la integración de servicios en la nube.

Para gestionar los datos de manera eficaz y segura, se optó por implementar la base de datos en phpMyAdmin alojada en Amazon Web Services (AWS). Esta decisión se fundamentó en los múltiples beneficios que ofrece AWS, como la alta disponibilidad, la escalabilidad y las avanzadas características de seguridad. Además, phpMyAdmin proporcionó una interfaz gráfica intuitiva para la administración de la base de datos, lo que facilitó el trabajo de los desarrolladores en la gestión de datos y en la optimización de consultas SQL.

Durante el desarrollo de "Rutas sin Baches", se puso especial énfasis en la experiencia del usuario y la fiabilidad del sistema. La aplicación permitía a los usuarios no solo reportar baches, sino también visualizar el estado de las reparaciones y recibir notificaciones sobre las acciones tomadas en las áreas reportadas. Esta funcionalidad aumentó la transparencia y la confianza de los usuarios en el sistema.

El proyecto incluyó pruebas rigurosas para garantizar la estabilidad y seguridad de la aplicación. Se llevaron a cabo pruebas unitarias, de integración y de aceptación del usuario para identificar y corregir posibles fallos. Además, se implementaron medidas de seguridad para proteger los datos personales de los usuarios y asegurar la integridad de la base de datos.

Este proyecto representó un esfuerzo conjunto y multidisciplinario que integró conocimientos técnicos, metodológicos y de gestión de proyectos. La aplicación proporcionó una herramienta valiosa para identificar y gestionar denuncias, y sirvió como un caso práctico de desarrollo colaborativo en el que cada miembro del equipo aportó sus habilidades y conocimientos para lograr un objetivo común. Este proyecto integrador demostró cómo la tecnología puede ser aplicada eficazmente para resolver problemas cotidianos y mejorar la calidad de vida en las ciudades.

2. Objetivos

1. Objetivos generales del proyecto:

- Mediante este trabajo se pretende que los alumnos consigan:
- Gestionar y realizar el trabajo del área asignada, en relación con las cargas de trabajo propias del Ciclo Formativo (planificación del tiempo).
- Realizar consultas a la persona adecuada y saber respetar la autonomía de los compañeros.
- Mantener el espíritu de innovación y actualización en el ámbito de su trabajo, buscando habilidades para la sinergia de equipos y liderazgo.
- Gestionar eficazmente los conflictos que se puedan producir, mediando y contribuyendo al establecimiento de un ambiente de trabajo agradable y actuando sinceramente, respetuosa y tolerante.
- Resolver problemas y tomar decisiones individuales, siguiendo las normas y procedimientos establecidos, definidos dentro del ámbito de su competencia.

2. Objetivos específicos del Proyecto (vinculados a capacidades de los módulos):

Bases de datos

- Crear bases de datos definiendo su estructura y las características de sus elementos según el modelo relacional.
- Diseñar modelos lógicos normalizados interpretando diagramas entidad/relación.
- Realizar el diseño físico de bases de datos utilizando asistentes, herramientas gráficas y el lenguaje de definición de datos.
- Realizar las operaciones CRUD de la base de datos.

- Consultar y modificar la información almacenada utilizando asistentes, herramientas gráficas y el lenguaje de manipulación de datos.

Programación

- Realizar el diseño de clases necesarias para seguir el patrón MVC.
- Implementar cada una de las clases para lograr su objetivo: clases de interfaz gráfica pertenecientes a la vista, clases que representen el modelo y clases encargadas de la lógica de la aplicación pertenecientes al control.
- Desarrollar una aplicación gestione información almacenada en bases de datos relacionales identificando y utilizando mecanismos de conexión.
- Esta tarea permitirá alcanzar la competencia completa para desarrollar componentes software en lenguajes de programación orientados a objetos.

Entornos de Desarrollo

- Realizar el análisis y el diseño de cualquier aplicación empleando técnicas UML.
- Documentar aplicaciones.
- Gestionar las diferentes versiones de un software y el trabajo colaborativo.
- Realizar pruebas de testeo sobre los programas.
- Seguir metodologías ágiles para el desarrollo y planificación de programas.

3. Tecnologías utilizadas

Para el desarrollo del proyecto se ha utilizado diversas tecnologías y herramientas para lograr los objetivos de manera eficiente y eficaz. A continuación, se detallan los más destacados:

Entorno de Desarrollo Integrado:

- IntelliJ IDEA ULTIMATE: Para el desarrollo de la aplicación de Ruta Sin Baches, se decidió utilizar como IDE (Integrated Development Environment en inglés) IntelliJ de JetBrains. Esta herramienta proporcionó muchas características para escribir, depurar y desplegar código de manera eficiente y rápida. Además, con “Code With Me”, una funcionalidad de IntelliJ permitió que el equipo pudiese trabajar en el mismo código a la vez y aumentó la productividad pudiendo terminar tareas en menos tiempo de lo esperado.
- Eclipse IDE: Para el desarrollo de la aplicación en su parte más gráfica se utilizó Eclipse, instalando el plugin de WindowBuilder. Este plugin permitió al equipo trabajar con la interfaz gráfica de una manera más sencilla e intuitiva. Además de la interfaz gráfica se usó este IDE para el back-end para evitar posibles conflictos en la estructura de proyecto al cambiar de IDE.

Lenguajes de programación:

- Java: Se usó Java como lenguaje de desarrollo principal del proyecto por su robustez, portabilidad, amplio soporte de la comunidad y toda la documentación que existe por Internet, oficial o no oficial, y es el lenguaje más conocido por el equipo.
- PL/SQL, o Procedural Language/Structured Query Language: Aunque haya un debate en si este lenguaje es de programación o no, se ha decidido incluirlo junto a Java, ya que tiene características propias de un lenguaje de programación, como variables, funciones, etc. PL/SQL sirvió para desarrollar procedimientos almacenados, funciones y otros objetos de base de datos necesarios para interactuar con la base de datos de Oracle.

- MySQL: Este lenguaje fue utilizado para desarrollar la base de datos de la aplicación (CRUD, Create Replace Update Delete) sustituyendo a Oracle cuando cambiamos a phpMyAdmin en AWS.

Herramientas de gestión de código:

- GitHub: Una plataforma en la que se hospedó el repositorio Git de la aplicación. Esta web sirvió para colaborar de manera efectiva, realizar cambios y seguir esos cambios, revisar código y gestionar problemas de manera centralizada.
- GitKraken: Se utilizó esta interfaz gráfica de git para facilitar ciertas acciones (como por ejemplo, el “undo commit” o la resolución de conflictos).
- Git: Se utilizó esta herramienta para tener un SCV (sistema de control de versiones) para gestionar el código del proyecto y se complementó con GitHub.

Herramientas organizativas:

- Trello: Para la gestión y organización de tareas, se siguió la metodología ágil Kanban utilizando Trello. Esta plataforma permitió crear tableros personalizados, listar tareas pendientes, en progreso y completadas, asignar responsabilidades y dar seguimiento al progreso del proyecto de manera colaborativa. Además, se conectó el repositorio de GitHub y se añadió la documentación de Java para un acceso más fácil a estas.
- WhatsApp: Se utilizó como herramienta de comunicación instantánea entre los miembros del equipo. Esta plataforma permitió mantener al equipo de desarrollo conectados en tiempo real, compartir actualizaciones rápidas, resolver dudas y coordinar tareas de manera ágil.
- Discord: Esta herramienta ha sido utilizada para una buena comunicación con los demás integrantes mientras se desarrollaba la aplicación. Se ha optado por una dinámica de programación llamada Pair Programming, explicada en Metodología.

Herramientas de apoyo a la programación:

Todas estas herramientas de IA generativa han sido usadas con responsabilidad y con pleno control por parte del equipo de desarrollo, observando, evaluando y discriminando código generado por las siguientes inteligencias artificiales:

- ChatGPT: Una herramienta de inteligencia artificial, para obtener asistencia en la generación de texto, responder preguntas y obtener información relevante durante el desarrollo del proyecto, así cómo resolver problemas de lógica de programación
- Gemini: Una adicional para el desarrollo. Gemini es otra inteligencia artificial que nos apoyó a la hora de buscar información en Google de una manera más eficaz que un simple motor de búsqueda.
- GitHub Copilot: Esta ha sido la IA más utilizada, ya que te sugiere código mientras vas programando en el propio IDE, además utiliza tu últimas líneas de código para copiar tu forma de programar y personalizar el código. También ha servido para resolver dudas de programación rápidamente al poder tener un chat integrado en el IDE. Sin duda, se ha incrementado la productividad, se ha reducido muchísimas horas de programación y se ha eliminado tareas repetitivas, permitiendo al equipo centrarse en temas más importantes como la accesibilidad y la usabilidad para mejorar la experiencia del usuario.

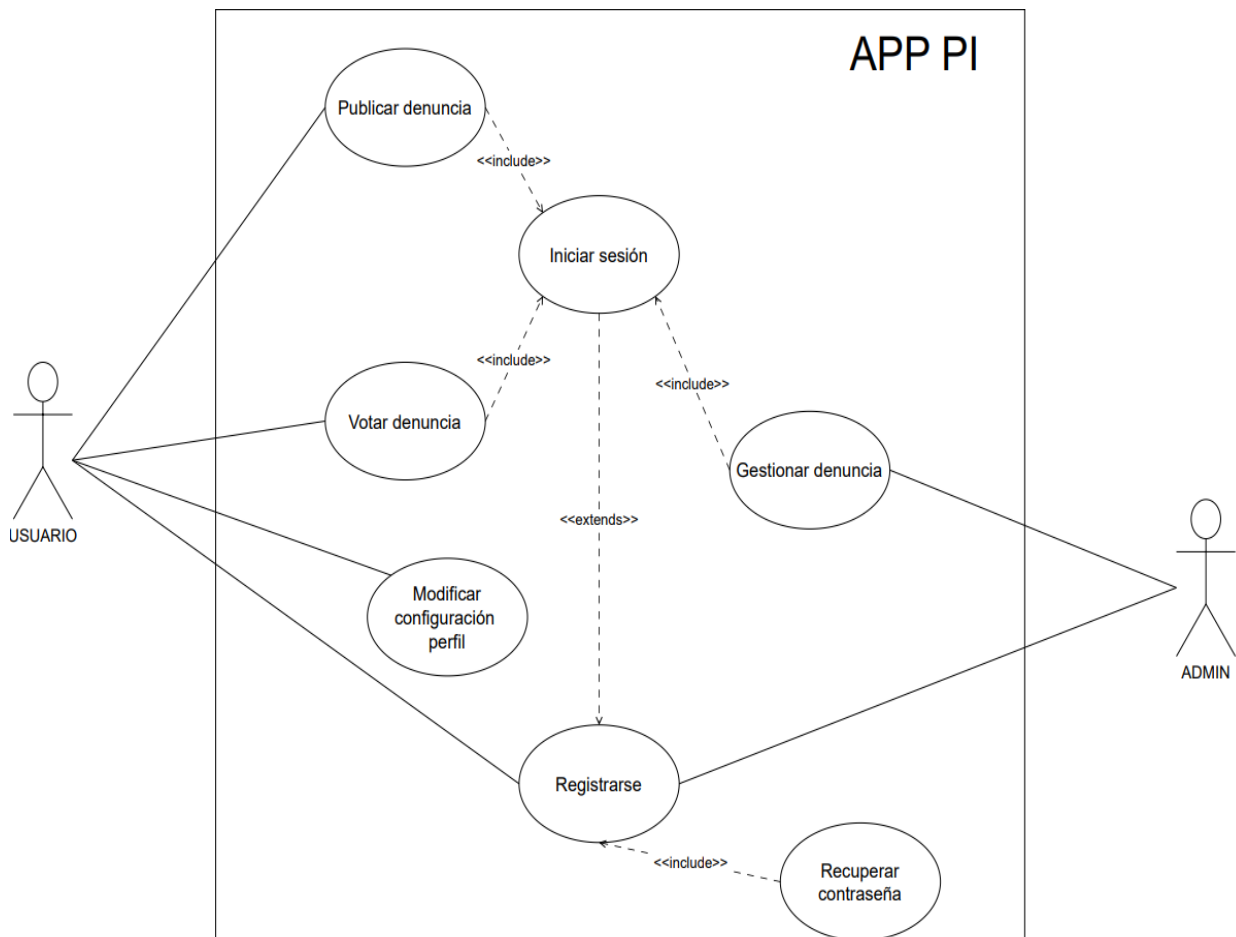
Herramientas de gestión de bases de datos:

- Oracle Database: El sistema de gestión de base de datos secundario para almacenar, recuperar y administrar los datos del proyecto. Oracle Database es una solución robusta y escalable que proporciona capacidades avanzadas de gestión de datos, también con su modalidad en la web llamada Apex Oracle, con las mismas funcionalidades que la versión de forma local.
- phpMyAdmin en AWS: El grupo decidió trabajar con MySQL creando la base de datos en Amazon Web Services. Ya que es mucho más sencillo almacenar la información de una manera conjunta (a diferencia que en localhost) siendo almacenada en la web. Además, la infraestructura en la nube de AWS proporciona alta disponibilidad y escalabilidad para nuestros datos. Esto ayudó a los integrantes a aprender esta tecnología de Amazon que tiene tantas ventajas para el desarrollo profesional del grupo.

4. Desarrollo e implementación

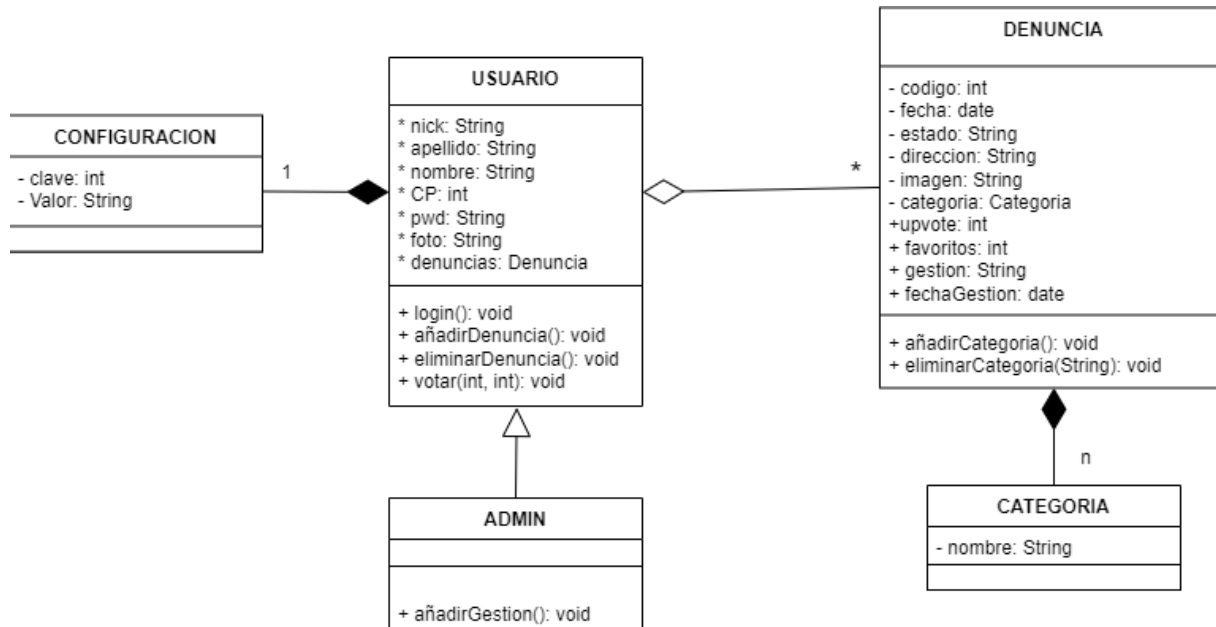
El desarrollo de la aplicación fue un trabajo difícil en general (y muy intensa), antes de comenzar a desarrollar se realizaron los pasos previos de diseño de diagramas, gracias a estos los pasos del desarrollo fueron más claros.

Comenzamos con la creación de varios diagramas fundamentales para la comprensión y construcción del sistema. El diagrama de casos de uso identifica las interacciones entre los usuarios (actores) y la aplicación, abarcando funcionalidades como el registro de usuario, inicio de sesión, actualización de perfil y visualización de datos para los usuarios registrados, así como la gestión de usuarios, generación de reportes y mantenimiento de la base de datos para los administradores. Este boceto básico aclaró en una primera instancia a el grupo de trabajo el funcionamiento (de una manera muy resumida) de la aplicación.



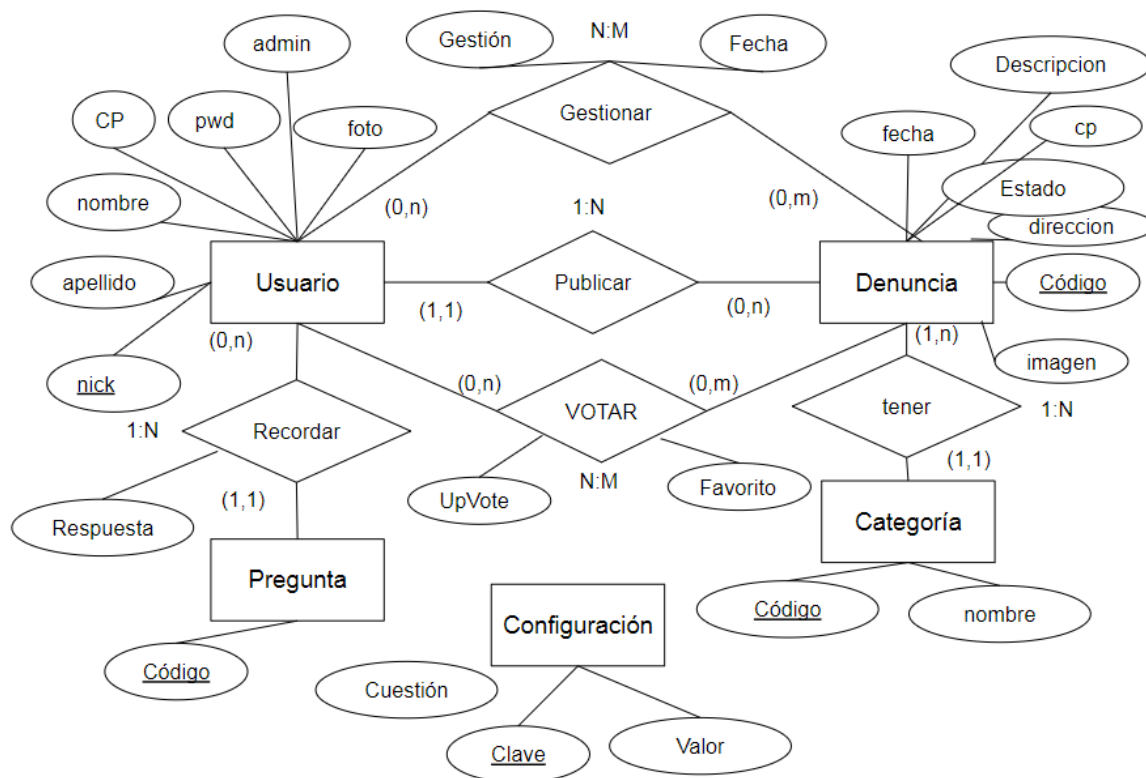
El diagrama de clases detalla la estructura del sistema, definiendo las clases principales, sus atributos, métodos y las relaciones entre ellas. Con este diagrama se puede visualizar la estructura del programa, con esto identificar las dependencias y responsabilidades de cada

clase. También ayuda con la planificación de la base de datos ya que las relaciones del diagrama se traducen en las entidades del modelo relacional.



El Modelo Entidad-Relación (MER V3, la tercera versión que se realizó debido a la falta de atributos en algunas relaciones) muestra cómo están organizados y relacionados los datos en la base de datos, ilustrando las entidades principales y sus relaciones. Este modelo se realizó previo al desarrollo, con esto se indicó claramente la distribución que la base de datos debía tener. Además de las claves primarias y foráneas que son esenciales para la evitar las restricciones al insertar borrar o modificar datos de las tablas (sin olvidar que la clave primaria nunca podía ser nula ni repetida)

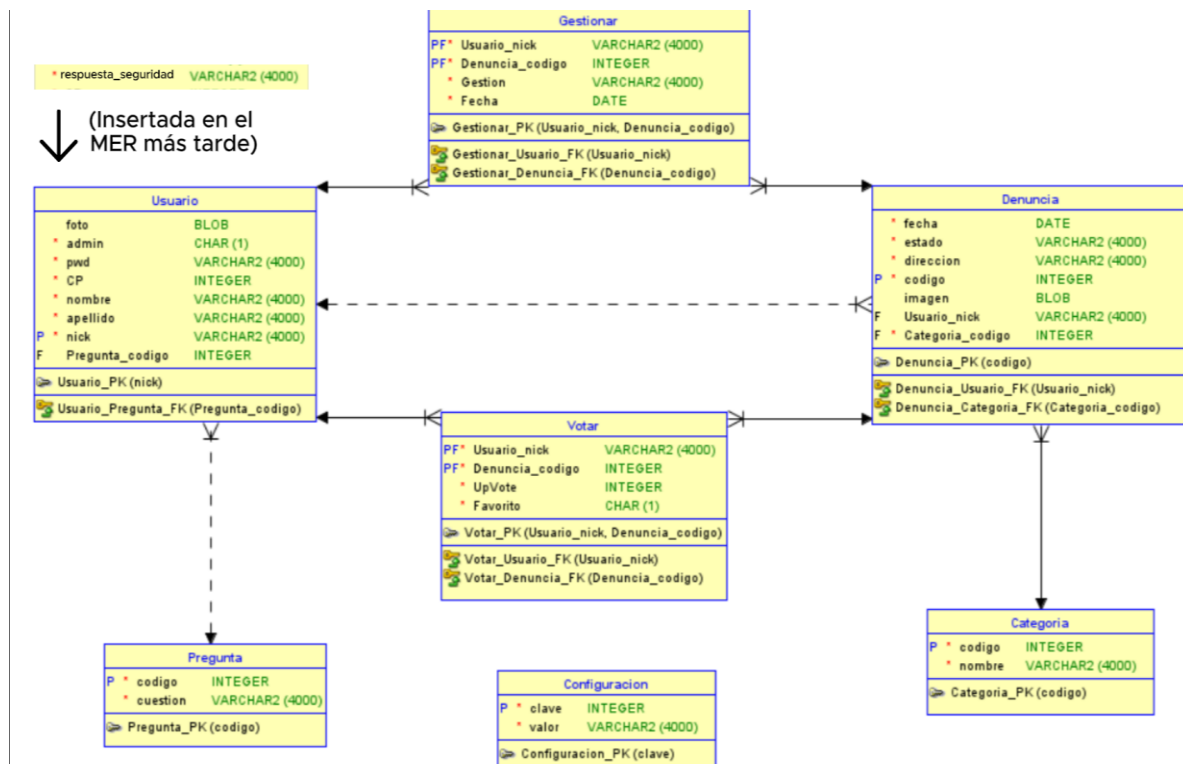
Con este MER vieron las tablas necesarias para el funcionamiento del proyecto, las creadas en una relación mediante una clave primaria (la unión de las dos claves foráneas que se unen con la relación, una vista más clara desde el Modelo Relacional).



El Modelo Relacional se realizó después del Modelo Entidad-Relación (MER) para convertir el esquema conceptual en un esquema lógico, optimizando el diseño de la base de datos. El Modelo Relacional toma las entidades y relaciones del MER y las traduce en tablas, columnas y claves, facilitando la implementación en un sistema de gestión de bases de datos (SGBD).

Para este paso, se utilizó la herramienta de Oracle Data Modeler, este permitió a el equipo de trabajo generar el Modelo Relacional automáticamente (haciendo una descripción de los campos existentes en el MER). Además, este programa permitió generar un script de la creación de la base de datos en SQL, que se insertó en Oracle (que más tarde fue llevado a MySQL).

Este MR no contempló la columna de "respuesta_seguridad" en la tabla usuario, que se insertó posteriormente en una nueva versión del MER (v3). El equipo de trabajo se percató de la falta de esta columna al implementar la funcionalidad del olvido de contraseña.



En el desarrollo de la aplicación en Java, se implementó el patrón de diseño Modelo Vista Controlador (MVC), lo cual resultó en múltiples ventajas que mejoraron tanto la estructura como el mantenimiento del código, este patrón permitió separar claramente la lógica de la aplicación, la interfaz de usuario y la gestión de datos, haciendo el proyecto más fácil de manejar. Todas estas clases del patrón se conocerán entre ellas en el Main del proyecto.

La aplicación de este patrón se realizó de siguiendo unas reglas en cada clase de este, el Modelo contenía la lógica y la gestión de datos. Todas las operaciones relacionadas con la base de datos como consultas o actualizaciones (además de la conexión con la base de datos) se manejaron desde el Modelo.

La Vista es la interfaz de usuario de la aplicación, cada una de las pantallas de nuestro proyecto hereda de la interface Vista. Además, esta interfaz toma los datos del modelo y presenta al usuario de forma más sencilla, todas las entradas del usuario las capturará la vista y mandará al controlador para que este las procese.

El Controlador actúa como intermediario entre la Vista y el Modelo, este recibe las entradas de la vista, las procesó (modificando el modelo) y actualizó la vista según fuera necesario, esta capa coordina el flujo de datos entre la vista y el modelo.

Una vez aplicado este patrón, la clase que contenía el método “main”, quedó mucho más libre y despejada, en ella únicamente se inicializaban todas las vistas (las pantallas del proyecto) y se conocían las clases de Modelo, Vista y Controlador (además de poner visible la primera pantalla de la aplicación).

En cuanto a la relación de la aplicación con la base de datos, esta fue creada en phpMyAdmin con MySQL (anteriormente realizada en Oracle con SQL). Se decidió cambiar el alojamiento de la base de datos debido a la posibilidad de alojar esta en AWS (Amazon Web Services). Este fue un reto al que se enfrentó el equipo (especialmente Henry Illescas) que dedicó mucho tiempo a ello, pero a su vez la aplicación dio un gran salto de calidad ya que se almacena toda la información en la web.

Se realizaron los scripts de creación, inserción y modificación de la base de datos originalmente en SQL, para posteriormente cambiarlos a MySQL siendo el lenguaje que utiliza phpMyAdmin (algo que fue relativamente sencillo debido a la similitud entre estos dos lenguajes).

Siguiendo por la interfaz gráfica de la aplicación, el equipo trató de realizar un diseño simple, pero a su vez muy visual y por supuesto con una alta usabilidad. Se basa en ayudar al usuario durante su experiencia en la aplicación, por eso esta desarrollada de manera intuitiva, diseñada para ser usada por todos los públicos.

El equipo de trabajo tomó la decisión de utilizar los colores verdes en la aplicación simulando así un camino (o sendero) con vegetación tratando así de idealizar estos caminos como “RutasSinBaches”, es decir, senderos perfectos por los que cualquier persona desearía caminar. Así los usuarios de la aplicación desearán que las calles de su localidad, barrio o distrito se vean tan bien como este camino.

Por ello el logo se compone por una carretera rodeada de un círculo verde que simboliza este camino ideal. Adjuntamos imagen del logo.



Se trató de realizar un inicio de sesión y registro rápido sin intervención de terceros (no hay que confirmar mediante correo electrónico, un autorregistro) de esta manera el usuario no tomará la tarea de registro como algo pesado. Con esto se trata de atraer a el máximo número de usuarios nuevos a la aplicación. Estas pantallas relacionadas con entrar en la aplicación tienen una estética similar, ocupando la mitad de la pantalla con una imagen ilustrativa de lo que sería una ruta ideal que busca el usuario.

Antes de entrar con las funcionalidades se trabajó el diseño gráfico de las pantallas, utilizando el plugin de Eclipse de WindowBuilder. En este únicamente se realizaron las maquetas de las ventanas (que cambiaron considerablemente con los cortos plazos dados).

Una vez entrada la aplicación esta sigue con el diseño intuitivo, facilitando todas las acciones lo máximo posible a el usuario. El proyecto en su conjunto trabaja con las tablas creadas en la base de datos, siendo así una manera muy ordenada de visitar las denuncias publicadas (además pudiendo manejar estas con botones con funcionalidades propias).

Con estos botones mencionados se realizan en la tabla de las denuncias las acciones de agregar como favorita, eliminar tu denuncia o en caso de administrador poder modificar estas.

Además, el usuario puede acceder a su perfil en cualquier momento de ejecución de la aplicación, pudiendo allí modificar sus datos personales introducidos en el registro.

Durante el desarrollo de la aplicación los integrantes del grupo trabajaron en esta por funcionalidades, cada vez que había una nueva implementación se realizaban pruebas manuales para verificar el correcto funcionamiento del nuevo cambio. Durante todas estas pruebas los integrantes del equipo testeaban todas las opciones que el usuario tuviese para que ninguna de ellas llegara a un error en la aplicación que la rompiera.

Una vez desarrollada la nueva funcionalidad, se lanzaba una versión a la que sometían a pruebas. Se llevaban a cabo pruebas exhaustivas, donde se exploraban todas las rutas posibles que un usuario podría tomar. Esto incluía la interacción con diferentes elementos de la interfaz, el manejo de entradas inesperadas y la navegación a través de distintas secciones de la aplicación. Entre ellas todas las posibilidades de inicio de sesión (como admin o usuario genérico)

Estas pruebas realizadas pudieron llegar a ser una de las tareas más complicadas (y que más tiempo llevó) durante todo el desarrollo de la aplicación, cada vez que los miembros del equipo (actuando como testers) detectaban un nuevo error, significaba una debacle en el ánimo del grupo. Cuando parecía que todo iba sobre ruedas, llegaba ese momento de encontrar el fallo, pero a su vez la satisfacción de todos al solucionar este problema mantenía a todos con la motivación para sacar adelante la aplicación.

En cuanto al diseño, para diferenciar el proyecto el equipo decidió utilizar un paquete de Desing, que sirvió para implementar nuevos temas a las pantallas, utilizando aún con más frecuencia el color verde (con esto la aplicación dio un gran salto de calidad visualmente).

Esto se realizó creando una nueva carpeta en el proyecto de diseño, esta está compuesta por una clase .java (que contiene el extends del paquete flatlaf) y una .properties (donde se introdujeron las propiedades visuales de las vistas). Por último, en el Main se introdujo la creación del objeto Desing, además de añadir una librería de tipo flatlaf.

5. Metodología

Para este apartado se recomienda una extensión de 3 a 6 páginas. Explicación de cómo se ha aplicado la metodología Scrum al trabajo, representando el flujo de trabajo con Sprint Backlog y Burndown chart, y añadir un enlace a Git y una imagen del tablero Trello al final del proyecto (junto al enlace). Cómo se ha coordinado el equipo para poder finalizar el trabajo.

En el presente proyecto se utilizó la metodología Scrum, ampliamente aplicada en proyectos de gran envergadura. Este método permitió dividir el trabajo en partes, facilitando un avance significativo en un menor tiempo. A pesar de contar con menos tiempo del previsto, esta metodología posibilitó la finalización del proyecto a tiempo y con todos los requisitos solicitados.

El rol de Scrum Master fue desempeñado por Arpad Kiss, cuya función principal fue controlar el repositorio Git mediante GitKraken, resolviendo conflictos surgidos durante la fusión de las ramas de los distintos integrantes con la rama principal. Además de gestionar el repositorio Git, Arpad Kiss revisó constantemente el trabajo, tanto a nivel de código como de interfaz, asegurando que se cumplieran los estándares de calidad.

La metodología Scrum permitió una organización eficiente del trabajo, promoviendo la colaboración y comunicación entre los miembros del equipo. La revisión constante del trabajo no solo garantizó el cumplimiento de los estándares de calidad, sino que también fomentó un entorno de aprendizaje y mejora continua.

El liderazgo de Arpad Kiss fue fundamental para mantener la coherencia y calidad del proyecto. Su habilidad para resolver conflictos de integración en GitKraken aseguró una base de código estable, minimizando retrasos y errores. La revisión constante del trabajo no solo

garantizó el cumplimiento de los estándares de calidad, sino que también fomentó un entorno de aprendizaje y mejora continua entre los miembros del equipo.

A lo largo del proyecto, se percataron de que no estaban dedicando el tiempo suficiente, gracias al Sprint Backlog que les permitió monitorear su avance diario. En ese momento, decidieron dedicar el fin de semana a trabajar durante 48 horas utilizando una metodología que no habían empleado hasta entonces.

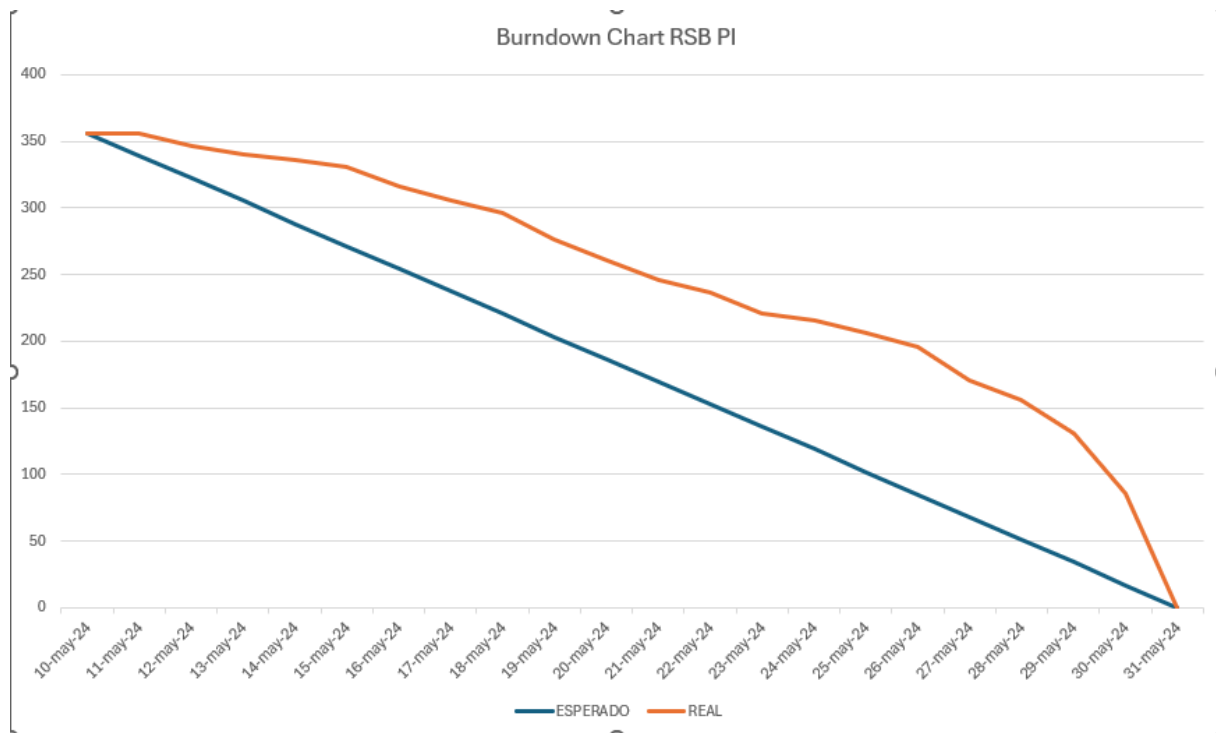
La metodología aplicada durante el fin de semana de trabajo fue el Pair Programming. Esto consiste en que una persona programa mientras otra supervisa, o sea, uno de los integrantes del grupo programaba y revisaba el código, mientras otro observaba su labor para asistirle en caso de errores o dificultades. Esta supervisión permitió una visión más general y facilitó la identificación de errores con mayor rapidez.

Durante este período intensivo de trabajo, la colaboración estrecha entre los miembros del equipo se convirtió en una herramienta crucial para avanzar significativamente en el proyecto. La metodología del Pair Programming demostró ser efectiva al potenciar la calidad del código y la coeficiencia del proceso de desarrollo. Gracias a esta estrategia, lograron mejorar su productividad y cumplir con los objetivos planteados para ese fin de semana.

Requisito	Tarea	Quien	Estado (No iniciada / en progreso / completada)	Dias:	1	2	3	4	5	6	7	8	9	10
				Horas										
				pendientes	356	314	288	241	235	195	120	37	11	0
RQ1	Registro	Mario	Finalizada		3									
RQ2	Registro	Henry	Finalizada		2									
RQ3	Registro	Henry	Finalizada		1									
RQ4	Registro	Kiss	Finalizada		1	1								
RQ5	Registro	Kiss	Eliminada		1	1	5	1						
RQ6	Registro	Mario	Finalizada		1	0	2	1						
RQ7	Registro	Mario	Finalizada		1	1	2	1						
RQ8	Login	Kiss	Finalizada		1	1	2	1						
RQ9	Login	Mario	Finalizada		1	0	1	1	2					
RQ10	Login	Nacho	Finalizada		1	1	5	1	2					
RQ11	Login	Mario	Finalizada		2	1	0	0	1	8			7	
RQ12	Denuncias	Henry	Finalizada		2	1	2	0	1	4	8			
RQ13	Denuncias	Nacho	Finalizada		4	1	3	0	2	4	7			
RQ14	Denuncias	Mario	Finalizada		2	0	2	0	1	4	7			
RQ15	Denuncias	Henry	Finalizada		2	0	1	0	2	8	3			
RQ16	Denuncias	Kiss	Finalizada		1	5	2	0	2	8	8			
RQ17	Denuncias	Nacho	Finalizada		1	2	2	0	3	7	8			
RQ18	Denuncias	Henry	Finalizada		1	1	2	0	2	7	4		5	
RQ19	Denuncias	Mario	Finalizada		3	1	2	0	2	5	5			
RQ20	Denuncias	Kiss	Finalizada		3	2	1	0	2	1	7		3	
RQ21	Denuncias	Mario	Finalizada		2	5	2	0	3	2	7			
RQ22	Ayuda/informacion	Nacho	Finalizada		2	0	5	0	2	8	5	4		
RQ23	Perfil usuario	Henry	Finalizada		2	1	4	0	2	2	7	1	8	
RQ24	Usabilidad	Nacho	Finalizada		2	1	2	0	1	7	4	6	1	2
				42	26	47	6	30	75	83	26	9		

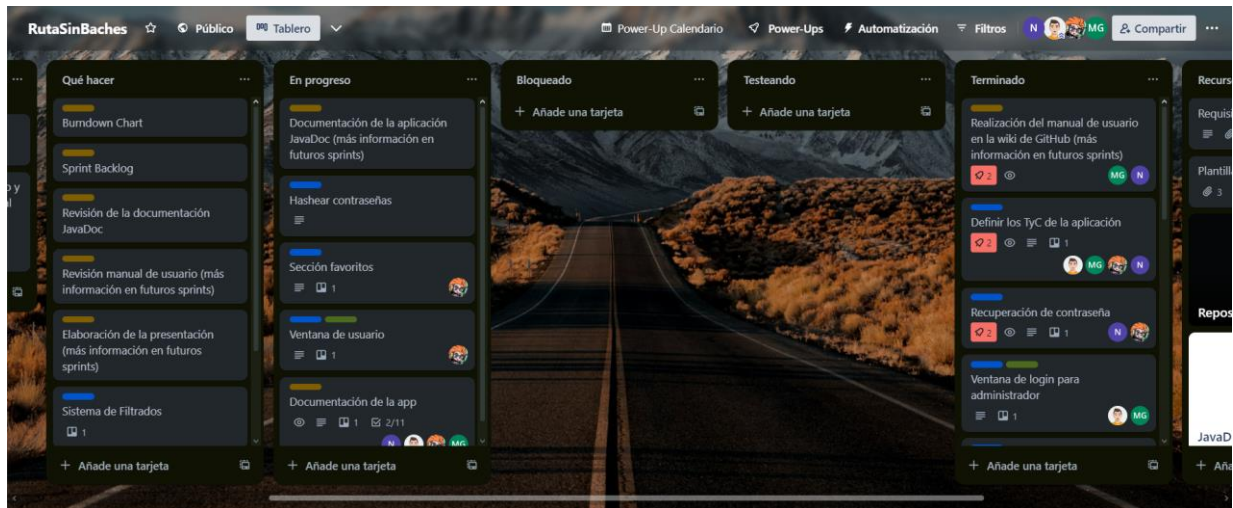
11

Otra de las metodologías implementadas fue Kanban, la cual permitió un mayor control y organización del trabajo durante el desarrollo de la aplicación. Utilizaron una herramienta en línea llamada Trello para aplicar esta metodología. Trello les permitió crear y gestionar etiquetas en diferentes categorías, como "En proceso", "Terminado" y "Por hacer". Esta categorización facilitó la visualización del flujo de trabajo y la identificación de tareas relacionadas que debían completarse conjuntamente.



Para gestionar el proyecto se realizó un Burndown Chart en el cual se mostró el problema ya comentado de la gestión del tiempo y los cortos plazos del proyecto.

Además, Trello ofreció la posibilidad de incluir enlaces, imágenes, videos y archivos PDF, lo que enriqueció la gestión del proyecto. Por ejemplo, podían adjuntar un PDF con los requisitos del proyecto o un enlace a GitHub para supervisar el progreso de los proyectos en desarrollo. Esta funcionalidad adicional mejoró la coordinación y la eficiencia del equipo, asegurando que todos los miembros tuvieran acceso a la información relevante y a los recursos necesarios para avanzar en sus tareas. Gracias a Kanban y Trello, el equipo logró una organización más efectiva y un seguimiento más preciso de las tareas, optimizando así el proceso de desarrollo.



Aquí deajo el enlace al Trello: <https://trello.com/b/ncskMBsP/rutasinbaches>

Aquí deajo el enlace a Git: <https://github.com/Hiromy24/RutaSinBaches>

6. Resultados y conclusiones

El proyecto "Rutas Sin Baches" se desarrolló con éxito en Java, contando con una interfaz intuitiva que facilita a los usuarios el reporte de baches. Se implementó el patrón de diseño MVC (Modelo Vista Controlador) para mejorar la estructura y el mantenimiento del código. Además, la base de datos se gestionó eficientemente con phpMyAdmin en AWS, garantizando alta disponibilidad y seguridad.

Se llevaron a cabo pruebas unitarias, de integración y de aceptación del usuario para asegurar la estabilidad y seguridad de la aplicación. También se implementaron medidas de seguridad para proteger los datos personales de los usuarios y asegurar la integridad de la base de datos.

El grupo concluye que el proyecto "Rutas Sin Baches" ha sido exitoso en su fase de desarrollo y está bien posicionado para un lanzamiento efectivo. La metodología permitió una organización eficiente del trabajo y una colaboración efectiva entre los miembros del equipo. La asignación clara de roles y responsabilidades facilitó la coordinación y la entrega de funcionalidades incrementales.

A pesar del poco tiempo obtenido para la realización del proyecto, el equipo queda satisfecho con los resultados obtenidos, ha sido capaz de oponerse a las muchas adversidades del camino para hacer una entrega a tiempo.

Entre los desafíos más significativos se encontraron problemas técnicos imprevistos, dificultades en la integración de los componentes del sistema y la necesidad de realizar ajustes sobre la marcha debido a cambios en los requisitos del cliente.

A pesar de estas dificultades, la complicitad y la sinergia entre los cuatro integrantes del equipo fueron fundamentales para superar los obstáculos (y no romper su relación de amistad). Cada miembro aportó sus habilidades y conocimientos de manera complementaria,

lo que permitió una colaboración efectiva y eficiente. La comunicación y el apoyo mutuo fueron clave para mantenernos enfocados y motivados, incluso en los momentos más duros.

El equipo queda orgulloso del trabajo realizado y del esfuerzo colectivo invertido en este proyecto. La calidad del resultado final es la recompensa de la dedicación y perseverancia del grupo. A través de un compromiso constante y una actitud positiva, se ha logrado cumplir con los objetivos propuestos, demostrando nuestra capacidad para entregar un producto sólido y bien estructurado.

En resumen, este proyecto no solo nos ha permitido crecer profesionalmente, sino también fortalecer nuestros lazos como equipo, dejando claro que el trabajo conjunto y la determinación son esenciales para alcanzar el éxito, incluso en circunstancias adversas.

7. Trabajos futuros

A pesar de los logros alcanzados y de haber superado numerosas adversidades, siempre hubo margen para la mejora y expansión del proyecto. En caso de haber contado con más tiempo y un conocimiento más profundo del lenguaje y las tecnologías involucradas, se habrían varias mejoras y adiciones.

En cuanto a la interfaz de usuario, se podría proponer un diseño responsivo que permita la adaptación de la aplicación a diferentes dispositivos (pasando de la aplicación de escritorio), además de incluir soporte para usuarios con discapacidades, mejorando así la accesibilidad.

Es crucial también realizar mejoras en la base de datos. Almacenar y analizar datos históricos es vital para planificar mantenimientos de manera efectiva, asegurando tanto la escalabilidad como el rendimiento, por lo que su implementación tendría un impacto increíble en el proyecto.

Para los usuarios, se deben incorporar funcionalidades como notificaciones personalizadas, que incluyan alertas en tiempo real sobre el estado de los reportes y reparaciones, así como permitir la personalización de las preferencias de notificaciones. Adicionalmente, un sistema de recompensas puede incentivar a los usuarios, mediante la creación de tablas de clasificación y logros que fomenten la participación activa.

En el ámbito de la seguridad y privacidad, la protección de datos es primordial. Se propone el uso de cifrado avanzado para proteger la información y la implementación de la autenticación de dos factores para aumentar la seguridad.

También, se deben establecer políticas claras que incluyan el consentimiento informado de los usuarios sobre el uso de sus datos, y permitir la gestión y eliminación de datos personales para darles mayor control (ya que no existe un documento de estos términos aceptados).

La aplicación requiere de sólida documentación y soporte, por lo que es esencial crear guías y tutoriales detallados y establecer un sistema de soporte técnico eficiente (muchos administradores capaces de estar a la confirmación y modificación de las denuncias publicadas). Las actualizaciones deben planificarse regularmente para asegurar una mejora continua del sistema, y se debe implementar un sistema de feedback que permita a los usuarios sugerir mejoras y valorar la aplicación.

En conclusión, con más tiempo y conocimientos, se podrían implementar mejoras significativas en "Rutas Sin Baches", aumentando su eficiencia, funcionalidad y seguridad. Esto no solo beneficiaría a los usuarios, sino que también contribuiría a mejorar la calidad de vida urbana, en la que todos convivimos.

Anexos

Anexo I – Listado de requisitos de la aplicación

Registro de usuario

RQ1: Durante el registro no intervendrán terceros, será mediante un autorregistro y una prueba anti-robots (un captcha).

RQ2.Consentimiento de notificaciones y términos al terminar el registro (incluyendo que eres mayor de 14 años)

RQ3: Después del registro, se mostrará una ventana de login.

RQ4.-Recuperar la contraseña a través de preguntas

RQ5-Contraseña segura=> la contraseña se guarda hasheada.

RQ6-Dos opciones a elegir si eres administrador o no y el administrador debe de tener que introducir una contraseña específica.

RQ7- Campos: Nombre y apellidos, nickname (opción de generación automática y debe ser único), CP, contraseña, y repetir contraseña para asegurarse de que está bien escrita. Todos son obligatorios.

Login

RQ8 -Ventana única para administrador en el cual al administrador se le envíe a una sección reservada para los administradores los cuales podrán moderar las incidencias, y por el otro lado a los usuarios se le enviaría a la sección para poner las incidencias.

RQ9 - Para evitar fuerza bruta, cerrar el programa al tercer intento fallido.

RQ10 - Tener que introducir el nickname y la contraseña. (Ambos son obligatorios).

RQ11: Al iniciar la aplicación se mostrará la pantalla de login y desde esta podrás ir al registro si todavía no lo has realizado, y en desde aquí a la pestaña del olvido de contraseña (contestando a las preguntas seleccionadas en el registro).

Denuncias

RQ12: Para denunciar un problema, se deberá proporcionar la siguiente información: -

Foto - Ubicación (se generará automáticamente en la versión 2) - Dirección postal -

Descripción del problema - Categorías.

RQ13: El administrador, autorizará los problemas, definirá cuando los casos estén cerrados confirmando la solución.

RQ14: Habrá un apartado donde se vean las denuncias marcadas como favoritas. Para que se pueda hacer un especial seguimiento de ellas.

RQ15: Un usuario puede avisar de que el problema ya ha sido arreglado.

RQ16: El administrador podrá ver la información completa de las denuncias, es decir, información de auditoría.

RQ17: Los usuarios podrán confirmar una denuncia ya existente.

RQ18: Sistema de filtrados: Para usuarios y administrador, los koalas las denuncias tengan categorías o información que haga más fácil buscar publicaciones que estás buscando (Usuario debe ponerlas).

RQ19: Si el usuario introduce una denuncia ya escrita (que coincida con el código postal y mismo tipo de denuncia), la aplicación le preguntará si se refiere a una de las publicadas, si es así podrá reconfirmar.

RQ20: El panel de administrador mostrará estadísticas sobre los tipos de problemas, las zonas y el tiempo de resolución de forma sencilla.

RQ21: Las denuncias pueden pasar por los siguientes estados: Nueva (solo visible para el creador y el administrador), Publicada/Rechazada, el administrador puede publicar y rechazar las denuncias (si es rechazada explicar el porqué), En proceso y Finalizada. Cada nuevo estado se registrará la fecha.

Ayudas o Información

RQ22: El usuario recibirá mensajes informativos de la funcionalidad de la aplicación cuando entras por primera vez a esta.

Perfil del usuario

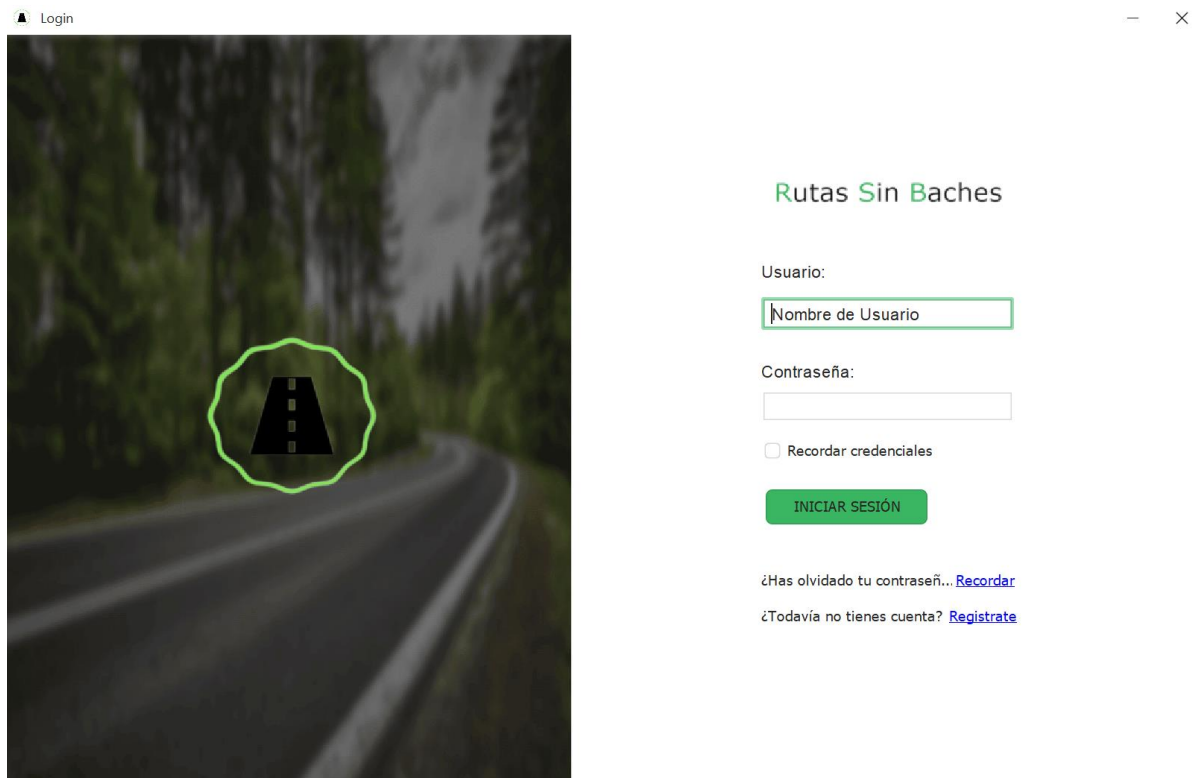
RQ23: Poder editar tu perfil personal como foto, cambio de contraseña, código postal, etc.

Usabilidad y accesibilidad


RQ24: La aplicación debe ser fácil de usar, accesible y segura.

Anexo II – Guía de uso de la aplicación

Incluirse en este anexo capturas de la aplicación, a modo de manual de usuario, incluyendo tanto el acceso de usuario, como de admin.



En esta primera pantalla, se ofrece la opción de iniciar sesión introduciendo el nombre de usuario (nickname) y la contraseña. Una vez que se presiona el botón de "Inicio de sesión", si todos los datos son correctos, el usuario será redirigido a la pantalla principal.

 Registro

¿Ya tienes cuenta? [Inicia Sesión](#)

Rutas Sin Baches

Nombre:

Apellidos:

Código Postal:


☐ Acepto los Términos y Condiciones

☒ Deseo recibir notificaciones

☐ Soy mayor de 14 años

☐ Soy Administrador

1/2



En la siguiente pantalla, se encuentra la primera parte del registro que solicitará al usuario su nombre, apellidos y código postal. Además, será necesario marcar las casillas de Términos y Condiciones, así como la casilla de edad, como requisitos mínimos para avanzar al siguiente paso del registro. En la parte superior de la pantalla, hay un enlace a la página de inicio de sesión si el usuario ya tiene una cuenta. Si se marca la casilla de "Soy administrador", se redirigirá a la pantalla de registro para administradores.

Registro

¿Ya tienes cuenta? [Inicia Sesión](#)

Rutas Sin Baches

Usuario:

Contraseña:

Repetir contraseña:

Selecciona la pregunta de seguri...

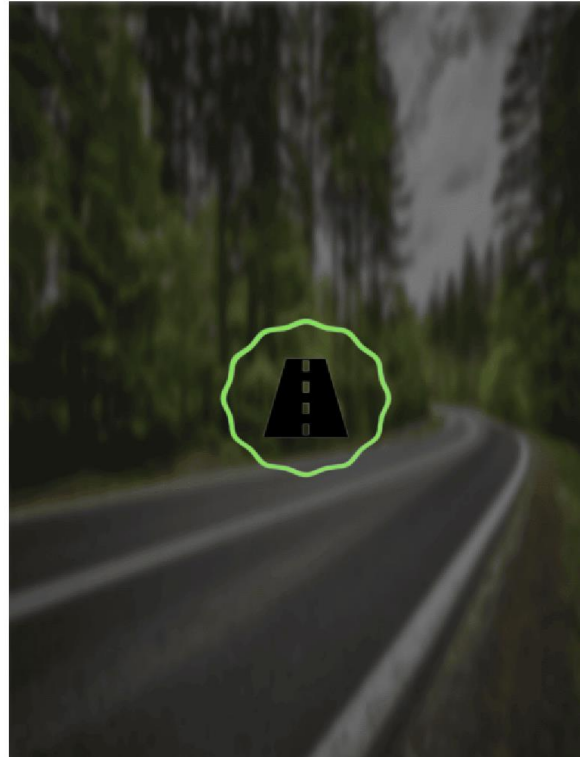
¿Cuál es tu nombre de pila? 

☐ No soy un robot



1/2

Registrarse



En esta pantalla, la segunda parte del registro presenta un enlace para iniciar sesión en la parte superior. Para completar el registro, se requiere rellenar todos los datos solicitados y realizar una prueba anti-robots. Tras finalizar el registro, se redirigirá de nuevo a la pantalla de inicio de sesión.

Registro

¿Ya tienes cuenta? [Inicia Sesión](#)

Rutas Sin Baches

Usuario:

Contraseña:

Repetir contraseña:

Selecciona la pregunta de seguri...

¿Cuál es tu nombre de pila?

Contraseña admin:

☐ No soy un robot

1/2

En esta pantalla, se mantiene el mismo formato que en la segunda parte del registro de usuario, con la diferencia de que este es un registro para administradores en el cual se solicitará una contraseña aparte de la que se utilizará para iniciar sesión.

Recuperar Contraseña

Rutas Sin Baches

Usuario:

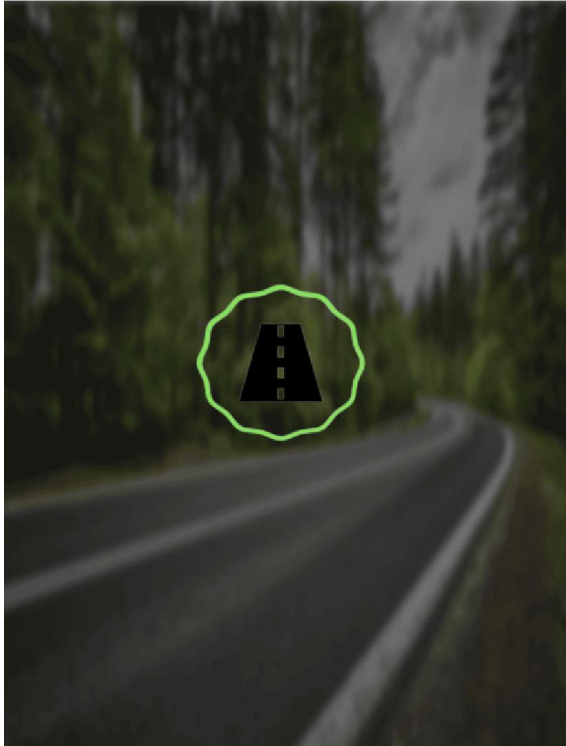
Selecciona la pregunta de seguridad:

¿Cuál es tu nombre de ...

Si se selecciona "recuperar contraseña" en el inicio de sesión, se redirigirá a una pantalla que

solicita el nombre de usuario y la respuesta a la pregunta de seguridad. Si se decide no continuar con la recuperación de la contraseña, se puede utilizar el botón de atrás para volver al inicio de sesión.

Recuperar Contraseña



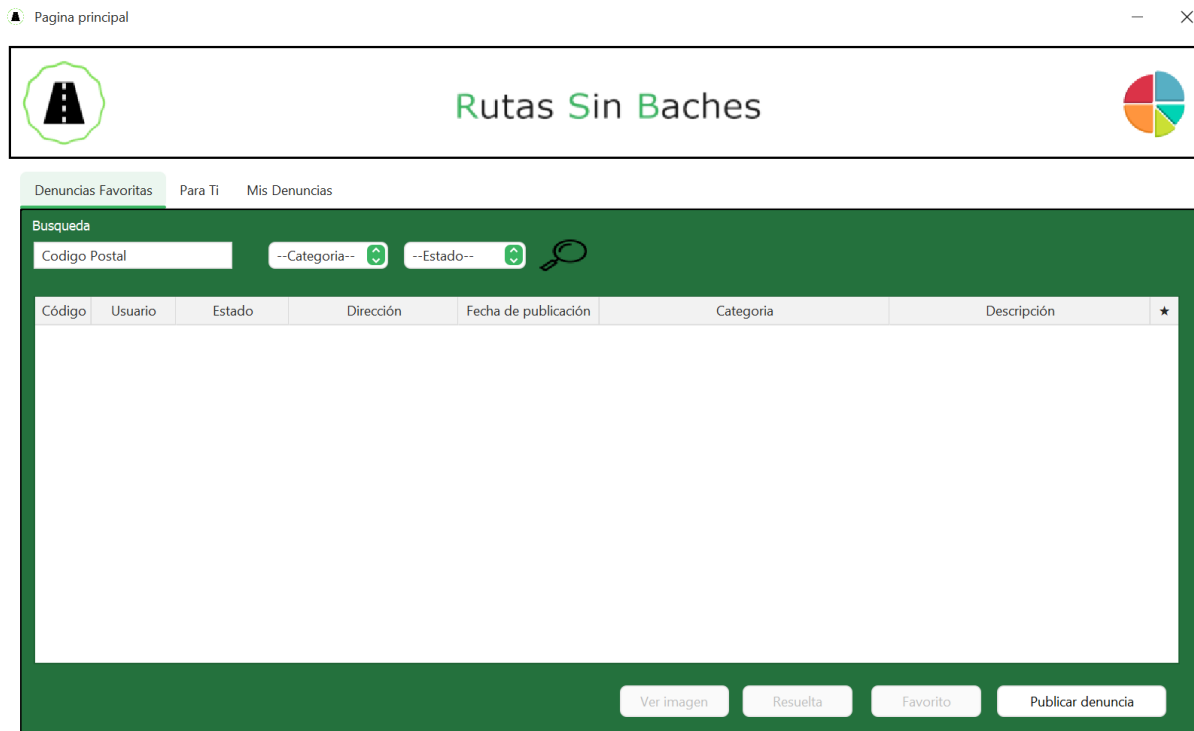
Rutas Sin Baches

Nueva contraseña:

Repetir contraseña:

[Cambiar contrase...](#)

Cuando se pasa de la primera pantalla para recuperar la contraseña, se verá la siguiente pantalla, en la cual se solicita introducir una nueva contraseña para proceder al cambio. Una vez cambiada la contraseña, se redirigirá nuevamente a la pantalla de inicio de sesión.



En la pantalla principal, se puede ver en primer lugar el encabezado, donde a la derecha aparece la foto de perfil cuando se haya subido una. Al hacer clic en la foto de perfil, se accede a la información personal. Debajo del encabezado, se encuentra la parte principal de la página, que muestra todas las denuncias insertadas en tablas. Estas denuncias se pueden ordenar por denuncias favoritas, "Para Ti" (ordenadas por cercanía), y "Mis Denuncias" (que muestra las denuncias publicadas por el usuario).

Los botones ofrecen diversas opciones, como ver la imagen de la denuncia seleccionada, marcar una denuncia como resuelta, y marcar una denuncia como favorita. Estos botones solo se pueden usar cuando se ha seleccionado una denuncia. Otra opción disponible es la de publicar una denuncia, la cual abrirá una nueva pantalla donde se podrá completar el proceso de publicación.

Publicar denuncia



Rutas Sin Baches



uta in aches




Publicar una denuncia

Código postal:

Provincia:

Ciudad:

Calle:

Categoría:

Iluminación 

Descripción:

Publicar denuncia

Al publicar una denuncia, se podrá indicar la ubicación del desperfecto. Además, será necesario especificar su tipo, ya sea de iluminación, edificios, etc. Se podrá subir una foto pinchando en el icono de la imagen y, por último, se deberá proporcionar una descripción para informar sobre la naturaleza del desperfecto. Una vez publicada la denuncia, se redirigirá a la página principal, donde se podrá ver la denuncia que se acaba de publicar.

Información personal



Rutas Sin Baches

Cambia tu información personal

Nombre:

Apellidos:

Código postal:

Contraseña:

Pregunta de seguridad:

Respuesta:



Hola @mariooseoanee



En esta pantalla podremos modificar todos los datos de nuestro perfil únicamente rellanando los campos que queramos modificar y después dándole a modificar, si pinchamos en la imagen podremos ponernos una foto de perfil y si ya tenemos una cambiarla. Pinchando en el icono puedes volver a la pantalla inicial.

Pagina principal admin


Rutas Sin Baches


Denuncias Favoritas
Para Ti
Mis Denuncias
Gestionar denuncias

Busqueda
Codigo Postal
--Categoria--
--Estado--

Código	Usuario	Estado	Dirección	Fecha de publicación	Categoria	Descripción	★
2	PedroSR	Publicado	Calle de mi otra casa 4 5...	2024-05-26	Edificios	Descripcion2	Y

Ver imagen
Favorito
Publicar denuncia

Esta pantalla es la página principal de admin, tiene el mismo funcionamiento, pero añadimos una parte en la que podremos gestionar las denuncias pinchando en la fila y modificando los datos y luego dándole a gestionar denuncia, un botón a la izquierda de imagen.

Pagina principal admin

Rutas Sin Baches

Denuncias Favoritas

Para Ti

Mis Denuncias

Gestionar denuncias

Busqueda

Codigo Postal

--Categoria--

--Estado--

Código	Usuario	Estado	Dirección	Fecha de publicación	Categoría	Descripción	★
30	hola	Nuevo estado	Nueva dirección 12345	2024-05-31	Edificios	Nueva descripción	N
26	hola	NUEVA	asd, asd, asd 12345	2024-05-31	Edificios	asd	N
25	hola	NUEVA	asd, sda, sdas 12345	2024-05-31	Edificios	asd	N
23	hola	NUEVA	qwr, ewqr, qwr 12432	2024-05-31	Pavimiento	qrqwr	N
22	hola	NUEVA	234, 342, 234 34233	2024-05-31	Limpieza	24323	N
21	hola	NUEVA	1111, 11111, 11111 11111	2024-05-31	Edificios	11111	N
20	hola	NUEVA	calle de tu puta madre, Pa...	2024-05-31	Edificios	Ladron rumano	N
19	hola	NUEVA	calle de tu puta madre, Pa...	2024-05-31	Edificios	Ladron rumano	N
18	hola	NUEVA	123, 123, 123 21323	2024-05-31	Edificios	123qewq	N
17	hola	NUEVA	yybyb, nuub, hbyh 2222...	2024-05-30	Iluminación	asdasdd	N
16	hola	NUEVA	yybyb, nuub, hbyh 12345	2024-05-30	Iluminación	asdasdd	N
15	hola	NUEVA	yybyb, nuub, hbyh 5555...	2024-05-30	Iluminación	asdasdd	N
14	hola	NUEVA	hgh, ghg, hjg 12345	2024-05-30	Iluminación	juohj	N
13	hola	NUEVA	sda, asda, ads 12345	2024-05-30	Iluminación	addasd	N
11	hola	NUEVA	er, madrid, parla 123	2024-05-30	Edificios	esf	N

Estadísticas

Modificar

Eliminar denuncia

Ver imagen

Favorito

Publicar denuncia

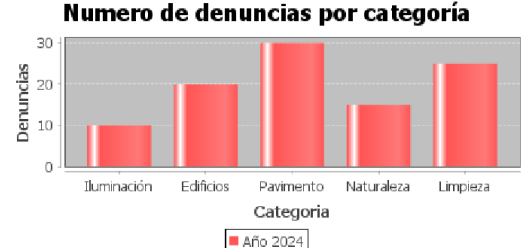
Por último, en la pantalla de gestionar denuncias también puedes ver las estadísticas como admin

PaginaPrincipal

Rutas Sin Baches

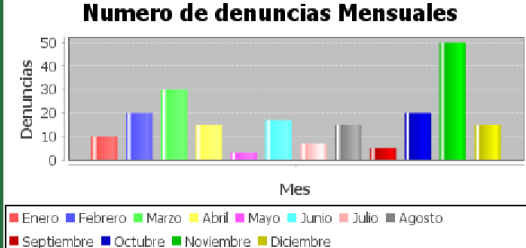
Estadísticas

Numero de denuncias por categoría



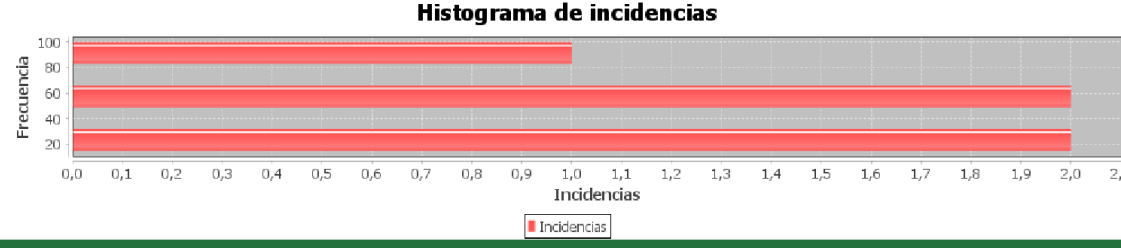
Categoría	Denuncias
Iluminación	10
Edificios	20
Pavimento	30
Naturaleza	15
Limpieza	25

Numero de denuncias Mensuales



Mes	Denuncias
Enero	10
Febrero	20
Marzo	30
Abril	15
Mayo	10
Junio	20
Julio	15
Agosto	10
Septiembre	10
Octubre	20
Noviembre	50
Diciembre	15

Histograma de incidencias



Incidencias	Frecuencia
0,0 - 0,1	100
0,1 - 0,2	100
0,2 - 0,3	100
0,3 - 0,4	100
0,4 - 0,5	100
0,5 - 0,6	100
0,6 - 0,7	100
0,7 - 0,8	100
0,8 - 0,9	100
0,9 - 1,0	100
1,0 - 1,1	100
1,1 - 1,2	100
1,2 - 1,3	100
1,3 - 1,4	100
1,4 - 1,5	100
1,5 - 1,6	100
1,6 - 1,7	100
1,7 - 1,8	100
1,8 - 1,9	100
1,9 - 2,0	100
2,0 - 2,1	100

39