AutoSmart: Aplicación de Gestión de Mantenimientos de Vehículos

Fatima Mortahil Chachou
Profesor: David Arriero Ollero
Curso: 2º DAM
Centro: IES Ribera del Tajo

23 de mayo de 2025

Índice

1. Introducción

AutoSmart es una aplicación multiplataforma desarrollada como proyecto final del ciclo de Desarrollo de Aplicaciones Multiplataforma. El objetivo principal es facilitar a los usuarios la gestión y el control de los mantenimientos de sus vehículos, permitiendo registrar revisiones, cambios de aceite, ITV y otros servicios, así como recibir notificaciones para no olvidar ninguna tarea importante.

Motivación: Elegí este proyecto porque considero que la gestión del mantenimiento de los vehículos es una necesidad real tanto para particulares como para empresas. Muchas personas olvidan realizar mantenimientos periódicos, lo que puede derivar en averías costosas o sanciones. AutoSmart pretende solucionar este problema mediante recordatorios automáticos y un historial digital accesible.

Objetivos:

- Facilitar la gestión y el registro de mantenimientos de vehículos.
- Notificar al usuario de los mantenimientos próximos y vencidos.
- Permitir la consulta del historial de servicios y gastos asociados.
- Ofrecer una interfaz sencilla, intuitiva y adaptada a dispositivos móviles.
- Garantizar la seguridad y privacidad de los datos del usuario.

2. Estudio de mercado y viabilidad

El sector de la automoción y la movilidad está en constante crecimiento y transformación, impulsado por la digitalización y la necesidad de optimizar costes y recursos. En España, el parque automovilístico supera los 30 millones de vehículos, y tanto particulares como empresas buscan soluciones para gestionar el mantenimiento y evitar averías costosas o sanciones por incumplimiento de revisiones e ITV.

Análisis de la competencia

Actualmente existen aplicaciones como Drivvo, Fuelio, MyCar y Car Expenses. Sin embargo, presentan las siguientes limitaciones:

- Muchas son de pago o incluyen publicidad intrusiva.
- No todas están traducidas o adaptadas al mercado español (ITV, normativa local).
- Algunas requieren registro obligatorio o acceso a datos personales sensibles.
- Pocas ofrecen integración con servicios en la nube gratuitos y exportación de datos.

Necesidades detectadas

- Usuarios particulares: necesitan recordar revisiones, cambios de aceite, ITV y controlar gastos.
- Empresas de flotas: requieren control centralizado y notificaciones para varios vehículos.

- Talleres: pueden recomendar a sus clientes el uso de apps para fidelización y recordatorio de citas.

Oportunidades de negocio

- Creciente uso de smartphones y apps de productividad.
- Falta de soluciones gratuitas, sencillas y adaptadas a la normativa española.
- Posibilidad de integración futura con talleres, aseguradoras o servicios de ITV.
- Potencial para ofrecer servicios premium (exportación, informes, sincronización multiusuario).

Viabilidad técnica y económica

- **Técnica:** El desarrollo es viable con herramientas gratuitas (Android Studio, Java, Firebase, Room) y recursos disponibles online.
- **Económica:** No requiere inversión inicial, salvo el tiempo de desarrollo. El mantenimiento es mínimo gracias a la nube.
- **Legal:** Cumple la LOPD y la RGPD, ya que los datos se almacenan de forma segura y sólo se usan para la funcionalidad de la app.

Estudio de viabilidad FOL

- **Obligaciones fiscales y laborales:** Al ser un proyecto académico, no genera obligaciones. Si se comercializara, habría que darse de alta como autónomo o empresa y cumplir con la fiscalidad y la seguridad social.
- **Prevención de riesgos:** El desarrollo se realiza en entorno seguro, sin riesgos físicos. Se recomienda pausas regulares y ergonomía.
- **Ayudas y subvenciones:** Existen programas de apoyo a la digitalización y emprendimiento joven (Injuve, Cámara de Comercio, etc.).

Guion de trabajo

1. Análisis de necesidades y competencia
2. Diseño funcional y técnico
3. Implementación y pruebas
4. Documentación y presentación

Requisito	Obligatorio	Cumplido
Registro de usuario	Sí	Sí
Gestión de vehículos	Sí	Sí
Registro de mantenimientos	Sí	Sí
Notificaciones	Sí	Sí
Historial de mantenimientos	Sí	Sí
Exportación a PDF	No	No

Cuadro 1: Tabla de requisitos funcionales

Fase	Fecha inicio	Fecha fin
Análisis de requisitos	01/03/2024	05/03/2024
Diseño	06/03/2024	10/03/2024
Implementación	11/03/2024	25/03/2024
Pruebas	26/03/2024	30/03/2024
Documentación	31/03/2024	05/04/2024

Cuadro 2: Cronograma del proyecto AutoSmart

Tabla de requisitos funcionales

Anexo VI: Cronograma del Proyecto

Anexo III: Diagrama de Casos de Uso

Anexo IV: Diagrama Entidad-Relación

Anexo V: Diagrama de Clases

3. Arquitectura de desarrollo

- **Lenguaje principal:** Java (Android)
- **Base de datos local:** Room (SQLite)
- **Base de datos en la nube:** Firebase Realtime Database
- **Notificaciones:** Android NotificationManager
- **Control de versiones:** GitHub
- **Diseño de interfaz:** Material Design

Se ha elegido Java por su robustez y compatibilidad con Android, y Firebase por su facilidad de integración y escalabilidad. El diseño sigue las guías de Material Design para ofrecer una experiencia de usuario moderna y accesible.

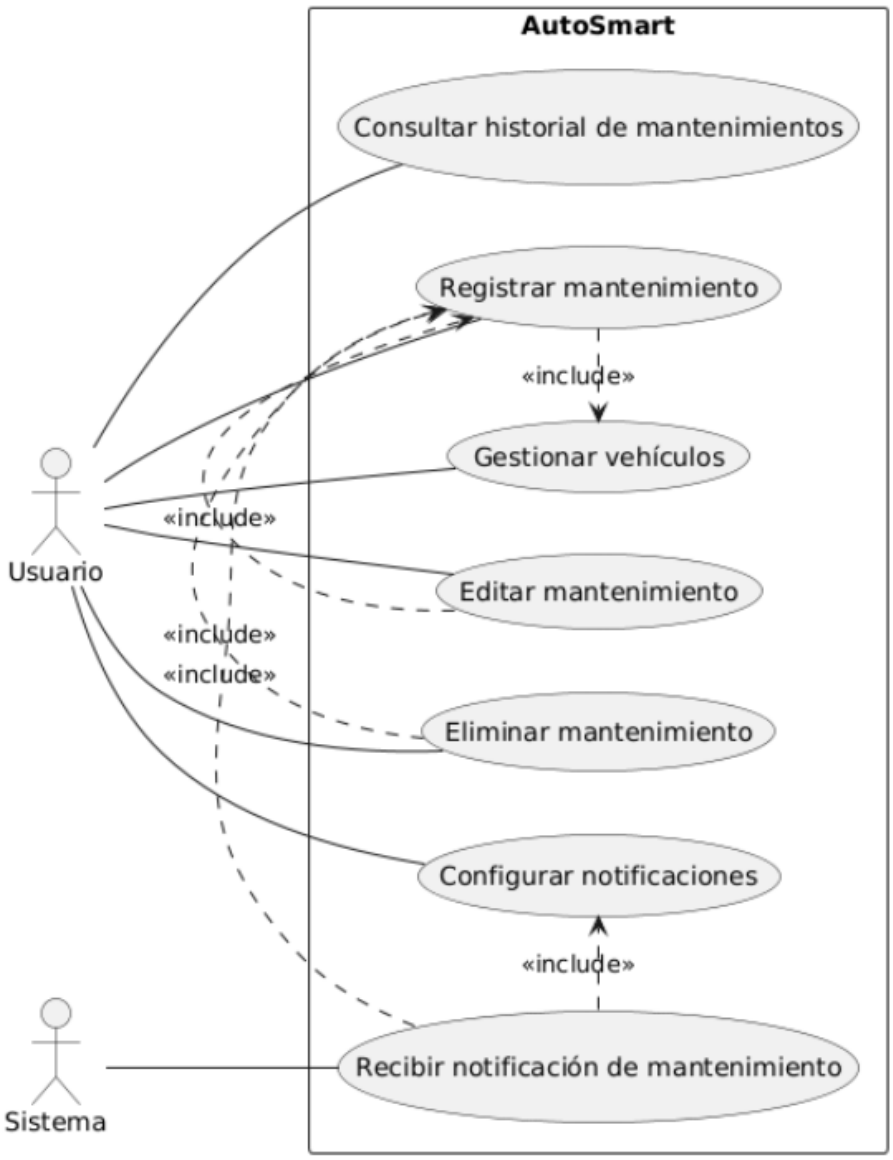


Figura 1: Diagrama de Casos de Uso

4. Estructura funcional del sistema

Requisitos funcionales:

- Registro y autenticación de usuarios.
- Alta, edición y baja de vehículos.
- Registro de mantenimientos (fecha, tipo, coste, descripción).
- Notificaciones de mantenimientos próximos.
- Consulta de historial de mantenimientos y gastos.

- Personalización de perfil y preferencias.
- Exportación de datos a PDF (mejora futura).

Módulos principales:

- Módulo de usuarios
- Módulo de vehículos
- Módulo de mantenimientos
- Módulo de notificaciones
- Módulo de configuración

Capturas de pantalla:

5. Casos de uso

Actores:

- Usuario registrado
- Sistema

Casos de uso principales:

- El usuario añade un nuevo vehículo.
- El usuario registra un mantenimiento.
- El sistema notifica al usuario de un mantenimiento próximo.
- El usuario consulta el historial de mantenimientos.
- El usuario edita o elimina un mantenimiento.

Ejemplo de diagrama de casos de uso:

6. Diseño de la base de datos

Tablas principales:

- **Usuarios:** id, nombre, email, foto
- **Vehículos:** id, usuarioId, marca, modelo, año, matrícula
- **Mantenimientos:** id, vehiculoId, fecha, tipo, descripción, coste

Relaciones: Un usuario puede tener varios vehículos, y cada vehículo puede tener varios mantenimientos.

7. Estructura del software

Estructura de carpetas:

```
com.example.autosmart
     activities
     adapters
     database
     models
     receivers
     utils
```

Clases principales:

- `MainActivity`
- `AddVehicleActivity`
- `AddMaintenanceActivity`
- `MaintenanceReminderReceiver`
- `AppDatabase`

8. Soluciones de diseño adoptadas

Patrón de diseño: Se ha utilizado el patrón MVC (Modelo-Vista-Controlador) para separar la lógica de negocio de la interfaz de usuario.

Validaciones: Los formularios validan que los campos obligatorios estén completos y que las fechas y horas sean correctas. Se muestran mensajes claros en caso de error.

Gestión de errores: Se emplean Snackbar y notificaciones visuales para informar al usuario de errores o acciones requeridas.

Capturas de pantalla:

9. Pruebas realizadas y resultados

Se han realizado pruebas funcionales de:

- Registro y autenticación de usuarios.
- Alta y edición de vehículos.
- Registro y edición de mantenimientos.
- Recepción de notificaciones.
- Validación de fechas y horas.

Resultados: Todas las funcionalidades principales funcionan correctamente. Las notificaciones se reciben en el momento programado y los errores se notifican de forma clara al usuario.

10. Manual de instalación y usuario

Requisitos:

- Dispositivo Android 7.0 o superior
- Conexión a Internet

Instalación:

1. Descargar el archivo APK desde el repositorio de GitHub.
2. Instalar la aplicación en el dispositivo.
3. Conceder permisos de notificación y almacenamiento.

Guía de uso:

1. Registrarse con email y contraseña.
2. Añadir un vehículo.
3. Registrar los mantenimientos realizados.
4. Activar las notificaciones en la configuración.
5. Consultar el historial y editar o eliminar registros.

11. Presupuesto

- Horas de desarrollo: 40h x 0€ (proyecto académico)
- Herramientas: Android Studio, Firebase (gratuito)
- Coste total: 0€

12. Desafíos y resolución de problemas

Durante el desarrollo de AutoSmart, me encontré con varios desafíos técnicos y de diseño que me ayudaron a crecer como desarrolladora. A continuación, detallo los principales problemas y cómo los solucioné:

- **Gestión de notificaciones en Android:** Inicialmente, las notificaciones de mantenimiento no se mostraban correctamente en todos los dispositivos. Tras investigar la documentación oficial y foros, descubrí que era necesario crear un canal de notificación y solicitar permisos explícitos en versiones recientes de Android. Implementé la comprobación de permisos y la creación dinámica del canal, logrando que las notificaciones funcionaran en todos los casos.
- **Validación de fechas y horas:** Los usuarios podían seleccionar fechas y horas incorrectas para los mantenimientos. Añadí validaciones en los formularios y mensajes de error claros usando Snackbar, mejorando la experiencia de usuario y evitando registros inválidos.

- **Persistencia de datos y sincronización:** Integrar Room (SQLite) para el almacenamiento local y Firebase para la sincronización en la nube supuso un reto, especialmente para mantener la coherencia de los datos. Implementé métodos de sincronización y pruebas exhaustivas para asegurar que los datos se guardaran y recuperaran correctamente.
- **Diseño adaptativo y accesibilidad:** Conseguir que la interfaz fuera intuitiva y se adaptara a diferentes tamaños de pantalla requirió varios ajustes en los layouts y el uso de Material Design. Probé la app en distintos dispositivos y resolví problemas de visualización y usabilidad.
- **Errores de compilación y dependencias:** En ocasiones, surgieron errores por dependencias incompatibles o recursos no encontrados. Aprendí a leer los mensajes de error detenidamente, buscar soluciones en Stack Overflow y limpiar/reconstruir el proyecto para resolverlos.

Enfrentar estos desafíos me permitió mejorar mis habilidades de resolución de problemas, aprender a documentarme y a no rendirme ante los obstáculos técnicos. Cada dificultad superada ha contribuido a la calidad final del proyecto.

13. Mejoras futuras

- Añadir soporte multiplataforma (iOS, web).
- Integrar recordatorios de ITV y seguro.
- Permitir exportar el historial a PDF.
- Añadir gráficos de gastos de mantenimiento.
- Integrar autenticación biométrica.

14. Bibliografía

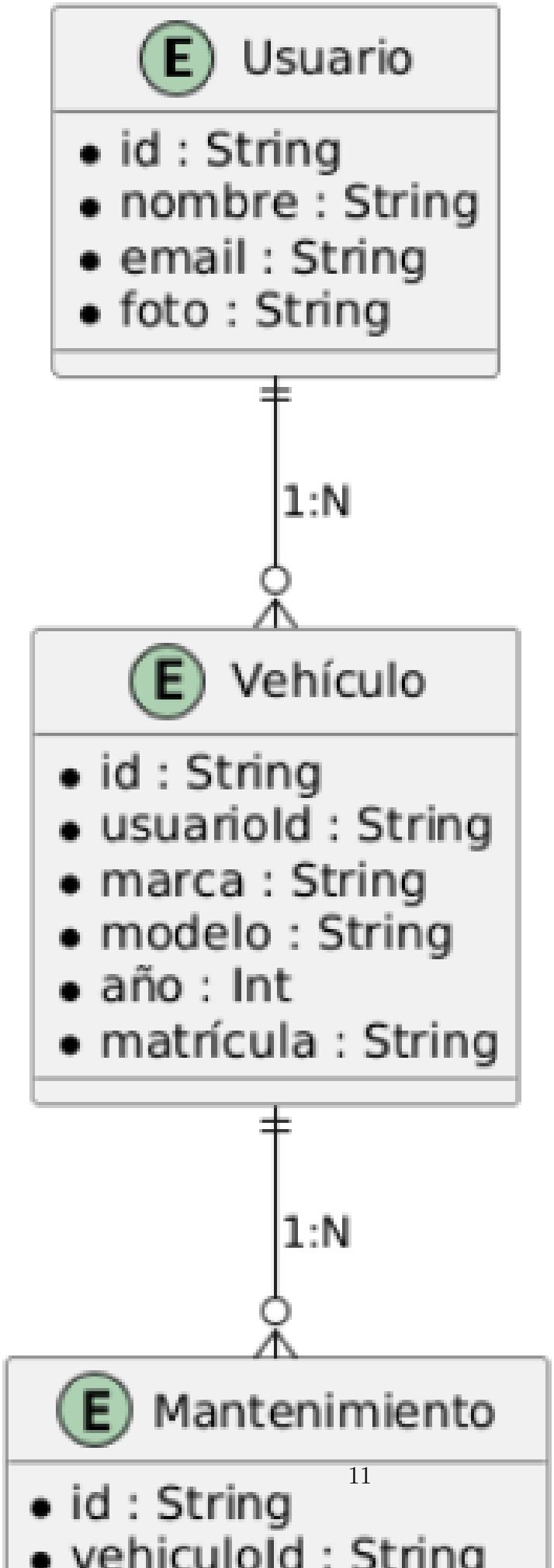
- Documentación oficial de Android: `https://developer.android.com/`
- Firebase Realtime Database: `https://firebase.google.com/docs/database`
- Curso de Programaciones didácticas y procesos de evaluación en FP (Edición 1)
- Portal Educación Castilla-La Mancha
- Material Design: `https://m3.material.io/`

Anexo I: Rúbrica de evaluación

Criterio	Peso	Instrumento	Puntuación
Identificación de necesidades y estudio de mercado	20 %	Aplicación	2.0

Diseño y fases del proyecto	30 %	Aplicación	3.0
Planificación y documentación	30 %	Documentación	3.0
Seguimiento, control y calidad	20 %	Exposición	2.0

Anexo II: Capturas y diagramas


Usuario
• id : String
• nombre : String
• email : String
• foto : String
1:N
Vehículo
• id : String
• usuariold : String
• marca : String
• modelo : String
• año : Int
• matrícula : String
1:N
Mantenimiento
• id : String
• vehiculold : String

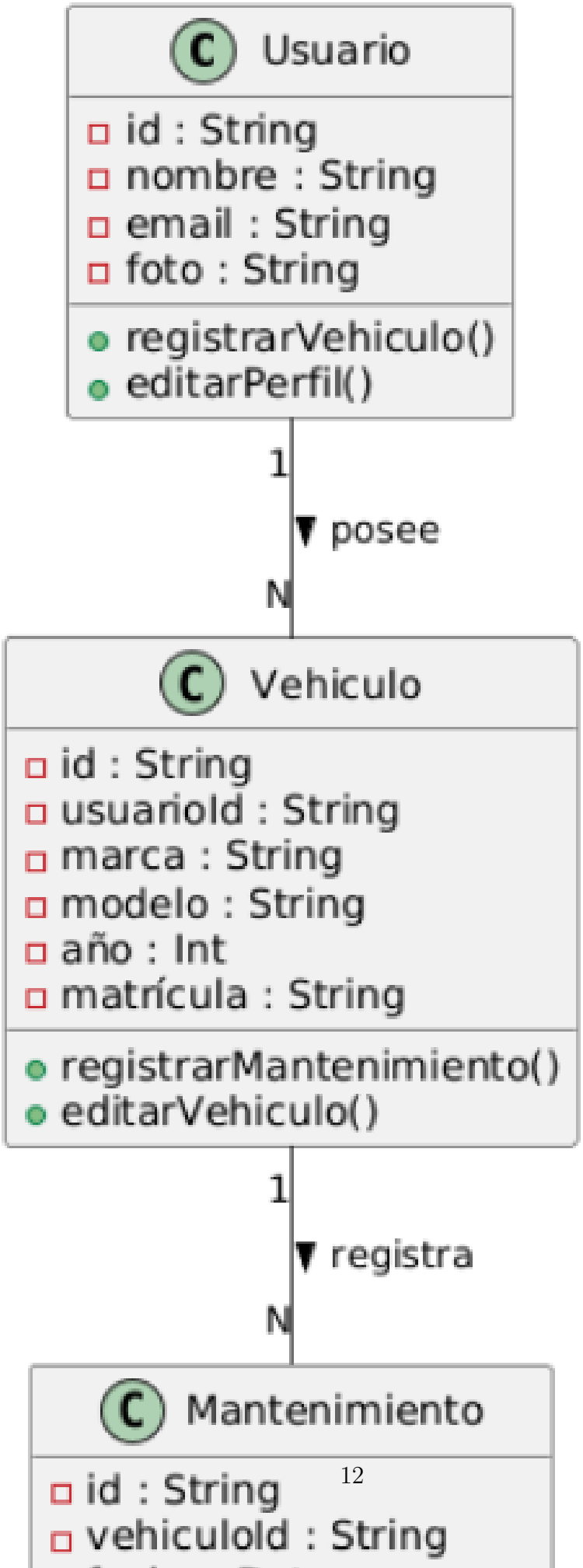

Usuario
id : String
nombre : String
email : String
foto : String
registrarVehiculo()
editarPerfil()
1
posee
N
Vehiculo
id : String
usuariold : String
marca : String
modelo : String
año : Int
matrícula : String
registrarMantenimiento()
editarVehiculo()
1
registra
N
Mantenimiento
id : String
vehiculold : String

Figura 4: Pantalla de inicio de sesión

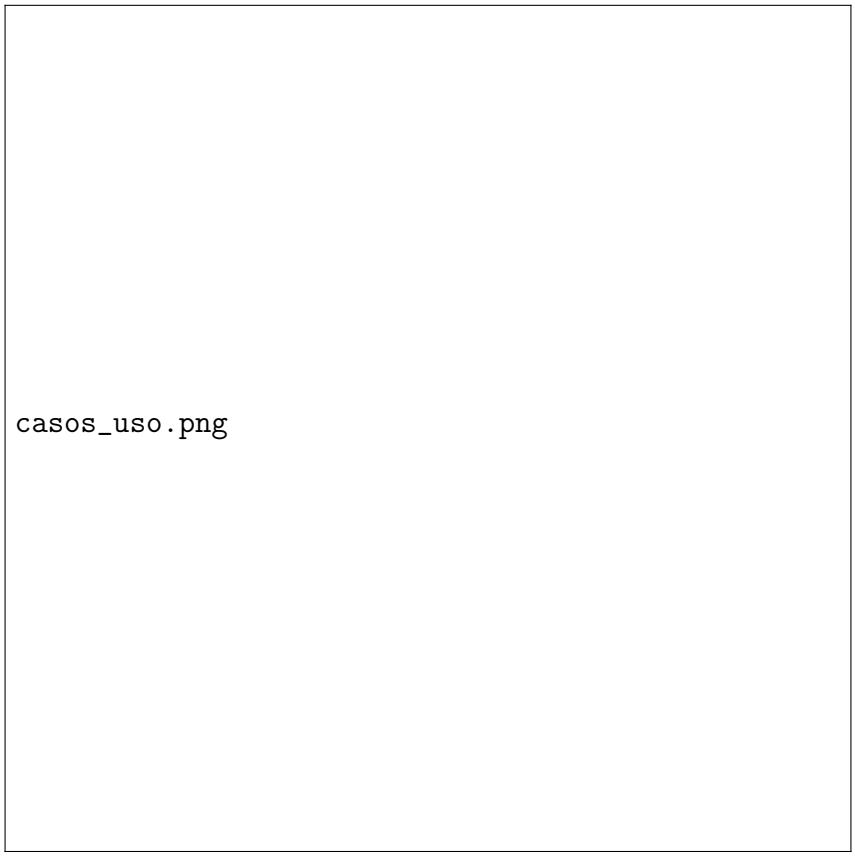


Figura 5: Diagrama de casos de uso

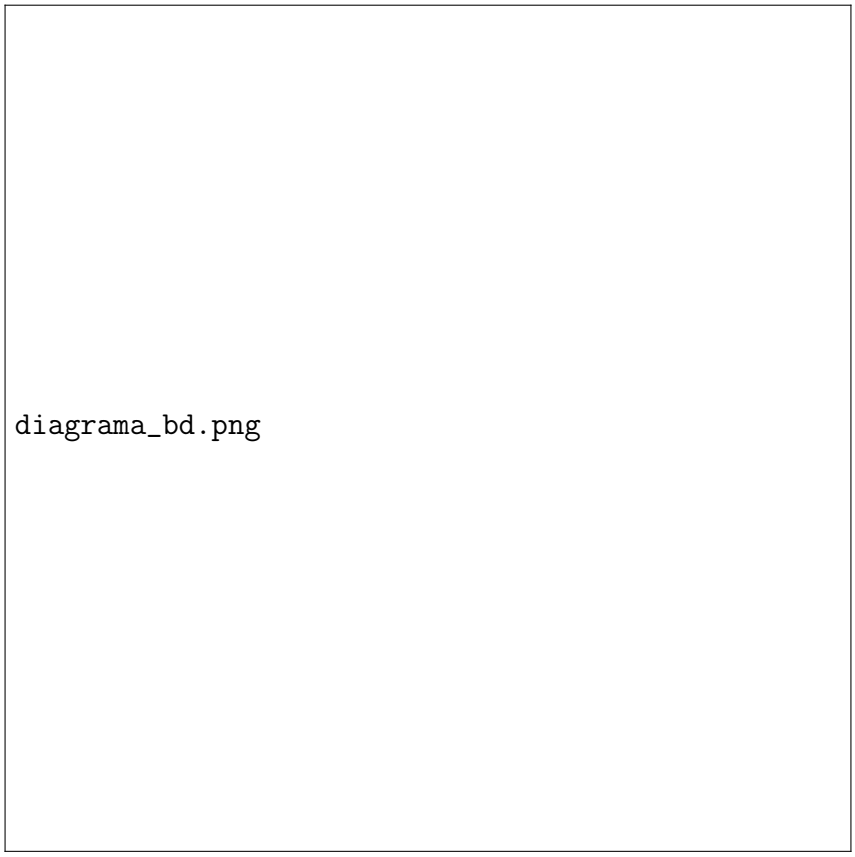


Figura 6: Diagrama entidad-relación

Figura 7: Formulario de registro de mantenimiento

Figura 8: Historial de mantenimientos

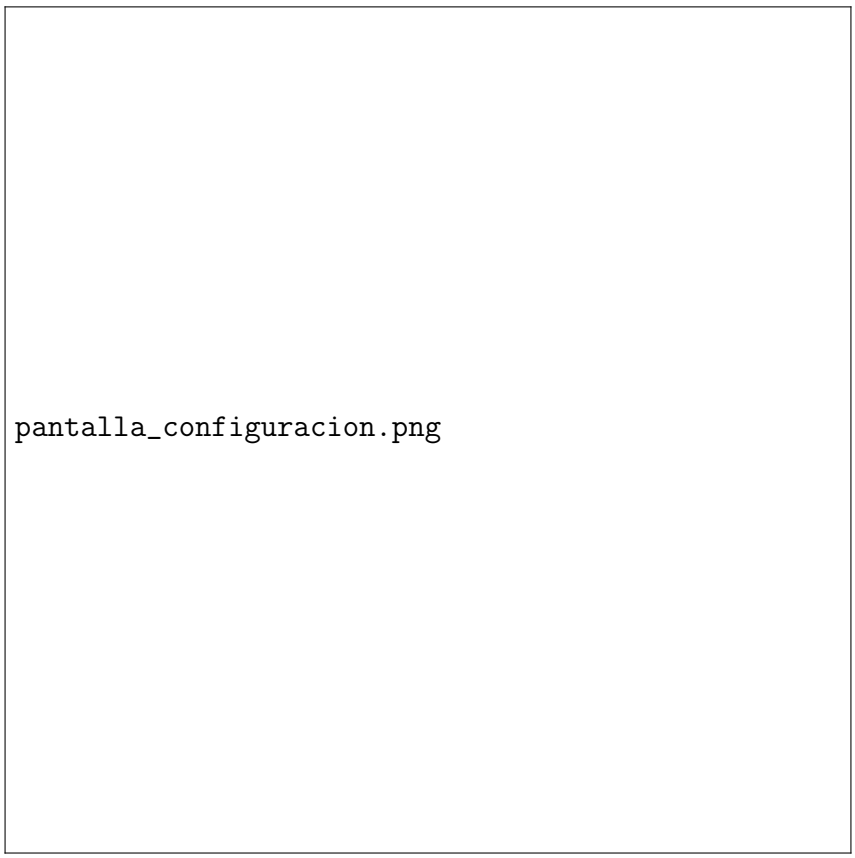


Figura 9: Pantalla de configuración de notificaciones