

LETS DRIVE



PROYECTO

Alumno: Juan Matés Castillo

Tutor: Indalecio García Mateos

Centro: CEU San Pablo Andalucía

Fecha entrega: 16/01/2026

Contenido

1.	INTRODUCCIÓN Y OBJETIVOS.....	4
	INTRODUCCIÓN DEL PROYECTO.....	4
	OBJETIVOS	5
	Facilitar la búsqueda mediante un medio digital.....	5
	Reducir el tiempo invertido en la búsqueda.....	5
	Generar beneficios económicos a través de este modelo de negocio.....	5
	Mejorar la visibilidad de los comercios locales.....	6
2.	CONTEXTO DEL PROYECTO	6
	INTRODUCCIÓN	6
	SITUACIÓN ACTUAL	6
	JUSTIFICACIÓN.....	7
3.	PLANIFICACIÓN Y COSTES.....	7
	METODOLOGÍA.....	7
	FASES DEL PROYECTO	8
	Fase 1: Ingeniería de Requisitos y Análisis del Dominio.....	8
	Fase 2: Modelado de Datos y Arquitectura en Capas	8
	Fase 3: Implementación del Backend (Spring Boot).....	9
	Fase 4: Desarrollo de la Interfaz y Lógica de Cliente (HTML/JS).....	9
	Fase 5: Ciclo de Pruebas y Aseguramiento de la Calidad (QA).....	10
	Fase 6: Despliegue y Plan de Mantenimiento	10
	PLANIFICACIÓN TEMPORAL	10
	Bloque 1: Análisis e Ingeniería de Requisitos (Semanas 1-3).....	11
	Bloque 2: Diseño de Arquitectura y Base de Datos Oracle (Semanas 4-6)	11
	Bloque 3: Desarrollo del Backend con Spring Boot (Semanas 7-11).....	11
	Bloque 4: Desarrollo Frontend y Conectividad (Semanas 12-14)	12
	Bloque 5: Pruebas, Calidad y Documentación (Semanas 15-18)	12
	ESTIMACIÓN DE COSTES.....	13
4.	DESARROLLO.....	15
	ANÁLISIS DE REQUISITOS.....	15
	Requisitos Funcionales (RF).....	15
	Requisitos No Funcionales (RNF).....	15

DISEÑO DE SOLUCIÓN	16
Arquitectura de Software	16
Diseño de la Base de Datos (Oracle)	17
Lógica de Integración (Spring Boot - JavaScript)	17
Diccionario de Datos (Oracle Database).....	18
5. PRUEBAS Y DESPLIEGUE	19
PLAN DE PRUEBAS.....	20
MANUALES TÉCNICOS Y DE USUARIO	21
Manual Técnico (Administración y Mantenimiento).....	21
Arquitectura de Despliegue y Dependencias	21
Configuración del Entorno y Base de Datos	21
Gestión de la API y Seguridad.....	22
Manual de Usuario (Guía de Explotación)	22
Interfaz de Búsqueda y Comparativa (Usuario Final).....	22
Panel de Gestión para Autoescuelas (Administrador del Centro).....	22
PLAN DE DESPLIEGUE	23
Configuración del Entorno de Producción	23
Estrategia de Migración de Datos (Data Migration)	23
Despliegue de la Capa de Cliente (Frontend)	24
Protocolo de Lanzamiento y Mantenimiento.....	24
6. CONCLUSIONES Y PROPUESTAS DE MEJORA.....	24
Propuestas de Mejora y Trabajos Futuros.....	25
7. BIBLIOGRAFÍA	26

1. INTRODUCCIÓN Y OBJETIVOS

INTRODUCCIÓN DEL PROYECTO

En la sociedad actualidad, caracterizada por la poca paciencia y la digitalización de servicios básicos, la obtención del permiso de conducción constituye un hito fundamental en la vida de cualquier persona, ya sea por motivos personales, laborales o de movilidad. El carnet de conducir no solo representa una competencia técnica, sino que es un catalizador de autonomía personal y un requisito de empleabilidad en numerosos sectores. De hecho, según datos de portales de empleo líderes como InfoJobs, aproximadamente el 35% de las ofertas de trabajo en España incluyen el carnet de conducir como un requisito obligatorio o altamente valorable, una cifra que asciende considerablemente en sectores como la logística y los servicios comerciales.

No obstante, la selección de una autoescuela que cumpla con los requisitos y preferencias individuales puede ser en ocasiones un proceso difícil y frustrante. El mercado actual está demasiado fragmentado: el usuario enfrenta una gran carga cognitiva debido a la diversidad de tarifas, a la amplia oferta disponible y a los distintos métodos de aprendizaje (en línea, presencial o híbrido). Este escenario se enmarca en un contexto donde el 83% de los ciudadanos españoles ya utiliza Internet para comparar precios antes de realizar una contratación, según el Informe del Sector Postal y del Comercio Electrónico de la CNMC. Lejos de ser útil, esta saturación de información genera frecuentemente lo que se conoce como "parálisis por análisis".

Debido a esta circunstancia y al acelerado progreso de las TIC (Tecnologías de la Información y la Comunicación), se hace necesario crear herramientas que simplifiquen estas decisiones, brindando una solución digital completa. La relevancia de este tipo de plataformas es evidente si consideramos que, según la Dirección General de Tráfico (DGT), cada año se expediten en España más de 400.000 nuevos permisos de conducir de la clase B, lo que supone una masa crítica de usuarios que demandan eficiencia digital. LetsDrive no solo es un almacén de datos, sino también un asistente inteligente que convierte la búsqueda analógica en un proceso más optimizado y claro.

OBJETIVOS

Facilitar la búsqueda mediante un medio digital.

El propósito principal de este proyecto es brindar una plataforma sencilla para el usuario, que permita la comparación entre autoescuelas según factores cruciales como la localización precisa, el costo y las clases de licencias ofrecidas.

- Concentración: Esta aplicación suprime la necesidad de llevar a cabo búsquedas dispersas en varias pestañas o en diferentes redes sociales.
- Normalización: Agrupa toda la información en un solo entorno digital coherente, lo que hace posible que los datos procedentes de distintas fuentes puedan ser comparados directamente.

Reducir el tiempo invertido en la búsqueda.

LetsDrive tiene como objetivo reducir el esfuerzo y las horas que una persona invierte en buscar un centro de formación, en una sociedad actual donde el tiempo es un recurso muy apreciado.

- Análisis del problema: Luego de llevar a cabo una investigación inicial, se descubrió que el 20% de los usuarios consultados necesitaban entre 1 y 2 semanas para hallar una autoescuela que les gustara. Esta encuesta se ha realizado a usuarios entre un rango de edad de 18 a 60 años a través de los formularios de Google.
- Solución técnica: El uso de filtros personalizados y la automatización de la comparación permiten que los resultados se consigan en minutos, lo que mejora la productividad del usuario.

Generar beneficios económicos a través de este modelo de negocio.

La aplicación cuenta con los pilares básicos para la obtención de ingresos con el fin de que la aplicación sea gratuita para los usuarios finales y poder expandir así este proyecto. Estos dos pilares son: la visualización de anuncios dentro de la plataforma y la obtención de una comisión por cada autoescuela registrada en la web.

Mejorar la visibilidad de los comercios locales.

Otros de los objetivos de LetsDrive es potenciar la presencia de comercios locales mediante un medio digital, ya que estas en ocasiones carecen de recursos para desarrollar estrategias de publicidad y de marketing propias. Al registrarse en esta plataforma, aumentan su alcance de llegar a nuevos clientes y competir en igualdad de condiciones con autoescuelas de mayor standing.

2. CONTEXTO DEL PROYECTO

INTRODUCCIÓN

En el marco actual, los usuarios demandan herramientas que simplifiquen la búsqueda y la comparación de servicios cotidianos como puede ser al que nos enfrentamos en este proyecto. Plataformas como comparadores de vuelos, hoteles o seguros han demostrado la eficacia de centralizar la información de manera clara y accesible. Sin embargo, este tipo de soluciones no se han desarrollado de la misma manera en el sector de las autoescuelas, a pesar de ser un servicio esencial por el que pasa gran parte de la población actual.

Elegir una autoescuela es una decisión relevante que implica tiempo, recursos económicos y planificación. A esto se le suma la escasez de herramientas que permitan comprar fácilmente precios, ubicaciones precisas o valoraciones reales de otros usuarios que ya han tenido experiencias en esas autoescuelas.

Dentro de este contexto aparece LetsDrive, diseñada para ofrecer un comparador claro y visual que permita al usuario tomar decisiones en menos tiempo. El planteamiento de la plataforma es poder llegar a un punto de unión entre la oferta existente y las necesidades de los usuarios, que busquen rapidez y opciones adaptadas a su zona residencia. El proyecto se apoya en las bases de un diseño centrado en el usuario, la geolocalización y la accesibilidad, con el fin de cubrir un vacío dentro del sector.

SITUACIÓN ACTUAL

El proceso de búsqueda de autoescuelas en España y más concretamente en Sevilla, muestra una serie de carencias que dificultan la experiencia del usuario. En términos generales, la información disponible suele ser incompleta y difícil de comparar o incluso dispersa. Esto hace que la búsqueda sea más lenta y menos eficiente.

En primer lugar, la presentación de datos no está estandarizada. Cada autoescuela publica sus precios, condiciones o métodos de enseñanza, lo que obliga al usuario a invertir tiempo en analizar individualmente cada opción. Además, muchas de las páginas no muestran tarifas actualizadas o apenas explican los servicios incluidos en el paquete formativo.

Las herramientas de geolocalización existentes como Google Maps, ofrecen una visión general de la ubicación, pero no permiten filtrar por servicios específicos como pueden ser la disponibilidad de cursos intensivos, idiomas o tipos de vehículos; obligando así al usuario a navegar entre decenas de pestañas de navegador para obtener una comparativa real.

JUSTIFICACIÓN

El desarrollo de LetsDrive se justifica por la necesidad de digitalizar un sector que, aunque esencial, se ha quedado rezagado en cuanto a herramientas de comparación centralizada. La implementación de esta plataforma aporta valor en tres dimensiones:

- Para el usuario: Democratiza el acceso a la información, permitiendo una toma de decisiones basada en datos objetivos y experiencias reales.
- Para las autoescuelas: Funciona como un escaparate digital que les permite competir en visibilidad con grandes franquicias sin necesidad de una gran inversión en marketing propio.
- Tecnología: Demuestra la viabilidad de integrar servicios de geolocalización y bases de datos dinámicas para resolver un problema de logística cotidiana.

3. PLANIFICACIÓN Y COSTES

METODOLOGÍA

Para el desarrollo de LetsDrive, se ha seleccionado la metodología Scrum, la cual es un marco de trabajo ágil que posibilita tratar proyectos complejos de manera iterativa y fundamentada en el valor. Esta opción se explica por la necesidad de modificar las funciones del comparador a medida que se recibe retroalimentación de los usuarios reales.

- Sprints: Los ciclos del proyecto tienen una duración de dos semanas.
- Roles: Se desempeña como Product Owner y developer para armonizar las exigencias del negocio con la ejecución técnica.

- Eventos: Al comienzo de cada sprint, se realizan reuniones de planificación para determinar qué funciones de búsqueda y filtrado serán implementadas.

FASES DEL PROYECTO

El desarrollo de LetsDrive se ha estructurado siguiendo un ciclo de vida basado en la metodología ágil. A continuación, se detallan las actividades técnicas de cada etapa:

Fase 1: Ingeniería de Requisitos y Análisis del Dominio

Esta fase es el pilar sobre el cual se construye la lógica del negocio.

- Captura de requisitos: Se han transformado las necesidades de los usuarios (detectadas en la encuesta donde el 20% tardaba más de una semana en decidir) en historias de usuario.
- Análisis técnico: Se justifica el uso de Oracle por su capacidad para manejar transacciones ACID y Spring Boot por su arquitectura basada en la inversión de control (IoC).
- Definición de restricciones: Se establecen límites como la latencia máxima de las consultas de geolocalización y el cumplimiento de la RGPD (Reglamento General de Protección de Datos) para el manejo de datos de las autoescuelas.

Fase 2: Modelado de Datos y Arquitectura en Capas

Aquí es donde se diseña la "columna vertebral" del proyecto utilizando el potencial de Java y Oracle.

Para el desarrollo de LetsDrive, se ha implementado una Arquitectura Hexagonal. El objetivo principal de este patrón es separar la lógica de negocio (el núcleo del software) de los detalles técnicos externos, como la base de datos Oracle, las interfaces de usuario o los servicios de terceros.

Esta arquitectura permite que el sistema sea independiente de la tecnología, facilitando, por ejemplo, que en el futuro se pueda cambiar el motor de base de datos o el frontend sin necesidad de reescribir la lógica de negocio central.

- Diseño del Modelo Relacional (Oracle): Se diseñan tablas normalizadas para evitar anomalías. Por ejemplo, la tabla AUTOESCUELAS se relaciona con TARIFAS mediante una relación 1:N, permitiendo que un centro tenga múltiples precios según el carnet.
- Arquitectura Spring Boot: Se define una estructura de paquetes basada en el patrón de Arquitectura Hexagonal o de Capas:
 - Controller: Para gestionar las peticiones HTTP del frontend.
 - Service: Para la lógica de negocio.
 - Repository: Utilizando Spring Data JPA para realizar las consultas SQL contra la base de datos Oracle de forma eficiente.

Fase 3: Implementación del Backend (Spring Boot)

Esta fase representa el mayor volumen de trabajo técnico.

- Configuración de la persistencia: Configuración del DataSource para la conexión con la base de datos Oracle mediante controladores JDBC.
- Desarrollo de la API REST: Creación de los "Endpoints" que consumirá el frontend en JavaScript. Se programan filtros dinámicos en Java que permiten buscar por precio máximo, radio de distancia y tipo de licencia simultáneamente.

Fase 4: Desarrollo de la Interfaz y Lógica de Cliente (HTML/JS)

Se construye la capa que el usuario verá en su navegador.

- Frontend: Creación de interfaces con HTML5 optimizadas para SEO, permitiendo que las autoescuelas locales aparezcan en los motores de búsqueda.
- JavaScript: Desarrollo de la lógica para consumir la API de Spring Boot. Se implementa la visualización del mapa interactivo donde cada "marcador" representa una autoescuela recuperada de la base de datos Oracle.
- Diseño Responsive: Asegurar que la tabla comparativa sea legible tanto en ordenadores de sobremesa como en smartphones.

Fase 5: Ciclo de Pruebas y Aseguramiento de la Calidad (QA)

Para garantizar que LetsDrive es una herramienta profesional, se ejecutan tres tipos de pruebas:

- Pruebas Unitarias (JUnit 5): Pruebas en el backend de Java para asegurar que los cálculos de precios son exactos.
- Pruebas de Integración: Verificación de que el flujo de datos desde la base de datos Oracle llega correctamente al navegador a través de Spring Boot.
- Validación de Usuario: Comprobación final de que los filtros de geolocalización muestran los centros más cercanos de Sevilla en el orden correcto.

Fase 6: Despliegue y Plan de Mantenimiento

- Preparación del entorno: Configuración del servidor de aplicaciones para ejecutar el archivo JAR generado por Spring Boot.
- Carga de datos inicial: Script SQL para poblar la base de datos Oracle con la información real de las autoescuelas recopilada en la fase de investigación.
- Mantenimiento: Definición de una hoja de ruta para futuras actualizaciones (como el pago online de la matrícula).

PLANIFICACIÓN TEMPORAL

La planificación de LetsDrive se ha diseñado para ser ejecutada en un periodo total de 18 semanas (aproximadamente 4 meses y medio). Al utilizar una metodología ágil como Scrum, el tiempo se divide en sprints de dos semanas de duración cada uno, permitiendo revisiones constantes y ajustes sobre la marcha.

A continuación, se desglosan las fases cronológicas y las tareas técnicas asociadas:

Bloque 1: Análisis e Ingeniería de Requisitos (Semanas 1-3)

- Semana 1: Análisis de mercado y validación de las encuestas de usuario realizadas en Sevilla.
- Semana 2: Definición de la Especificación de Requisitos de Software (ERS). Identificación de Requisitos Funcionales y No Funcionales.
- Semana 3: Diseño de los diagramas de Casos de Uso.

Bloque 2: Diseño de Arquitectura y Base de Datos Oracle (Semanas 4-6)

- Semana 4: Diseño del Modelo Entidad-Relación.
- Semana 5: Diseño de la interfaz de usuario (UI) y flujos de navegación (UX).
- Semana 6: Configuración del entorno de desarrollo: Instalación del JDK, IDE (IntelliJ o Eclipse) y configuración del servidor local de Oracle.

Bloque 3: Desarrollo del Backend con Spring Boot (Semanas 7-11)

- Semana 7-8 (Sprint 1): Configuración de Spring Data JPA. Creación de las entidades Java y conexión con Oracle.
- Semana 9-10 (Sprint 2): Creación de servicios para la búsqueda por geolocalización y algoritmos de comparación de precios.
- Semana 11 (Sprint 3): Desarrollo de los endpoints para que el frontend pueda consumir los datos de las autoescuelas.

Bloque 4: Desarrollo Frontend y Conectividad (Semanas 12-14)

- Semana 12: Maquetación de las vistas principales en HTML5 y CSS3.
- Semana 13: Implementación de la lógica en JavaScript para las llamadas asíncronas (Fetch API) al backend de Spring Boot.
- Semana 14: Integración de la API de mapas y visualización dinámica de resultados.

Bloque 5: Pruebas, Calidad y Documentación (Semanas 15-18)

- Semana 15: Ejecución de pruebas unitarias (JUnit) e integración. Depuración de errores (Debugging).
- Semana 16: Pruebas de usabilidad con usuarios reales para validar la reducción del tiempo de búsqueda.
- Semana 17: Redacción de los manuales técnicos y de usuario.

Para evaluar el progreso, se han establecido los siguientes puntos:

- P1 (Semana 3): Aprobación de los requisitos del sistema.
- P2 (Semana 6): Base de datos Oracle operativa y modelada.
- P3 (Semana 11): Backend funcional con capacidad de respuesta JSON.
- P4 (Semana 14): Versión Beta de la plataforma accesible desde navegador.
- P5 (Semana 18): Entrega de la documentación final.

DIAGRAMA DE GANTT			SEMANA 1	SEMANA 2	SEMANA 3	SEMANA 4	SEMANA 5	SEMANA 6	SEMANA 7	SEMANA 8	SEMANA 9	SEMANA 10	SEMANA 11	SEMANA 12	SEMANA 13	SEMANA 14	SEMANA 15	SEMANA 16	SEMANA 17
B1: Análisis de requisitos																			
Análisis de mercado																			
Definición ERS																			
Casos de Uso																			
B2: Arquitectura y Base de datos																			
Modelo Entidad - Relación																			
Diseño UI/UX																			
Entorno Desarrollo																			
B3: Backend (Spring Boot)																			
Sprint 1: JPA y Entidades																			
Sprint 2: Geolocalización																			
Sprint 3: Endpoints API																			
B4: Frontend y Conexión																			
Maquetación HTML/CSS																			
JS y Fetch API																			
API Mapas y Dinamismo																			
B5: Calidad y Cierre																			
Pruebas JUnit y Debug																			
Pruebas Usabilidad																			
Manuales y Documentación																			

ESTIMACIÓN DE COSTES

Para la puesta en marcha de LetsDrive, se ha mirado de cerca todos los gastos que se tendrán que afrontar. Esto incluye todo lo relacionado con la tecnología y los sistemas que se van a necesitar.

La mayor parte de la inversión se va a utilizar para crear la infraestructura necesaria para que funcione nuestra plataforma, que utiliza Spring Boot y Oracle Database. También necesitamos algunos servicios externos que nos permitan saber la ubicación exacta de los lugares. El costo principal de la infraestructura es el alojamiento en la nube. Se necesita un Servidor Virtual Privado que sea robusto para ejecutar la Máquina Virtual de Java. Esto costará alrededor de 500-600 euros al año.

En cuanto a la base de datos, se utilizará la versión Oracle Cloud Free Tier o una instancia Standard, dependiendo del volumen de datos. De esta manera, el costo de la base de datos será bajo o incluso cero durante la fase de lanzamiento. A esto debe sumarse la adquisición del nombre de dominio (.com o .es) por unos 15 euros anuales y el uso de la API de Google Maps para las funciones de geocoding y visualización de mapas, cuyo coste es variable, pero se estima en unos 200-300 euros anuales bajo un uso moderado. A su vez, la plataforma necesita una partida de dinero para marketing digital. Esto ayudará a captar las primeras autoescuelas. El presupuesto para esto es de 2.000 euros. Este dinero se usará para campañas en redes sociales y para mejorar la posición en los resultados de búsqueda de internet en Sevilla.

Además, se ha reservado un poco de dinero extra, alrededor de 500 - 1000 euros, por si surgen problemas técnicos o se necesitan actualizar licencias. Esto es como un seguro para tener en cuenta cualquier cosa imprevista.

En total, el presupuesto para el primer año de funcionamiento de la plataforma es de 4000 euros. Esto incluye todo lo necesario para empezar y mantener la plataforma en marcha durante el primer año. Este montante representa la inversión mínima necesaria para transformar el desarrollo técnico en un servicio web comercialmente viable y disponible las 24 horas para los usuarios.

COSTES

CATEGORÍA	CONCEPTO DETALLADO	ESTIMACIÓN ANUAL (€)	OBSERVACIONES
Infraestructura Cloud	Servidor Virtual Privado (VPS) robusto para Java / Spring Boot	500€ - 600€	Capacidad para ejecutar la JVM de forma estable 24/7
Base de Datos	Oracle Cloud (Free Tier / Instancia Standard)	0€ - 50€	Se prioriza el uso de la capa gratuita para el lanzamiento
Identidad Digital	Nombre de dominio (.com / .es)	15 €	Registro y mantenimiento anual del dominio
Servicios API	Google Maps Platform (Geocoding y Mapas)	200€ - 300€	Coste variable basado en el volumen de consultas
Marketing y Captación	Campañas en redes sociales y SEO (Sevilla)	2.000,00 €	Inversión estratégica para captar las primeras autoescuelas
Fondo de Contingencia	Reserva para imprevistos técnicos o licencias	500€ - 1000€	"Seguro" para problemas técnicos o actualizaciones
TOTAL ESTIMADO		4.000 €	Inversión total para el primer año

COSTE RECURSOS HUMANOS

CONCEPTO	DETALLE	CÁLCULO ESTIMADO	VALORACIÓN (€)
Tiempo de Desarrollo	18 semanas (4.5 meses)	40h/semana = 720 horas totales	-
Salario de referencia	Mercado Fullstack Junior	24.000 € brutos/año	-
Valor de la mano de obra	Coste de oportunidad	18 semanas de salario	8.500,00 €

Por lo que el coste total del proyecto sería de unos 12500€

4. DESARROLLO

El desarrollo de LetsDrive se basa en la interpretación de la convergencia de una buena ingeniería de requisitos y un buen diseño de arquitectura, enfocado en suprimir la opacidad informativa del ámbito de la autoescuela en Sevilla. En esta sección, se redacta la transformación del proyecto desde su lógica hasta su estructura física y técnica.

ANÁLISIS DE REQUISITOS

Requisitos Funcionales (RF)

- RF1 - Sistema de Búsqueda Inteligente: El sistema debe permitir búsquedas por municipio, código postal o nombre de la autoescuela.
- RF2 - Filtrado por Categoría de Licencia: Capacidad de segmentar resultados por permisos (A1, A2, A, B, B+E, C, D).
- RF3 - Motor de Comparación Detallada: El usuario podrá marcar hasta tres centros y visualizar una comparativa técnica de precios de matrícula, tasas de examen y coste por clase práctica.
- RF4 - Geolocalización y Mapas: Representación visual de los centros en un mapa interactivo con cálculo de distancia respecto al usuario.
- RF5 - Gestión de Tarifas (Admin): Interfaz para que los centros actualicen sus precios, ofertas temporales y disponibilidad de cursos intensivos.
- RF6 - Sistema de Valoraciones y Reseñas: Los usuarios podrán consultar las experiencias de otros alumnos para asegurar la transparencia de la calidad docente.
- RF7 - Contacto Directo: Formulario integrado para solicitar información adicional a un centro específico sin salir de la plataforma.

Requisitos No Funcionales (RNF)

- RNF1 - Disponibilidad y Robustez: El sistema debe estar operativo el 99.5% del tiempo, apoyándose en la fiabilidad de Oracle para la gestión de datos.
- RNF2 - Tiempo de Respuesta: Las consultas al backend de Spring Boot no deben exceder los 2 segundos para no degradar la experiencia de usuario.

- RNF3 - Seguridad de Datos: Implementación de cifrado BCrypt para las contraseñas de las autoescuelas y protocolos HTTPS para toda la comunicación.

BCrypt es una función de “hashing” diseñada específicamente para contraseñas. Una de sus características principales es que genera una cadena aleatoria única para cada contraseña antes de cifrarla. Esto significa que, si dos autoescuelas usan la contraseña “123456”, sus versiones cifradas en la base de datos serán completamente diferentes. También permite configurar un “coste” que determina cuánto tiempo tarda el servidor en procesar la contraseña. Esto hace que los ataques de “fuerza bruta” sean extremadamente lentos y costosos para un hacker.

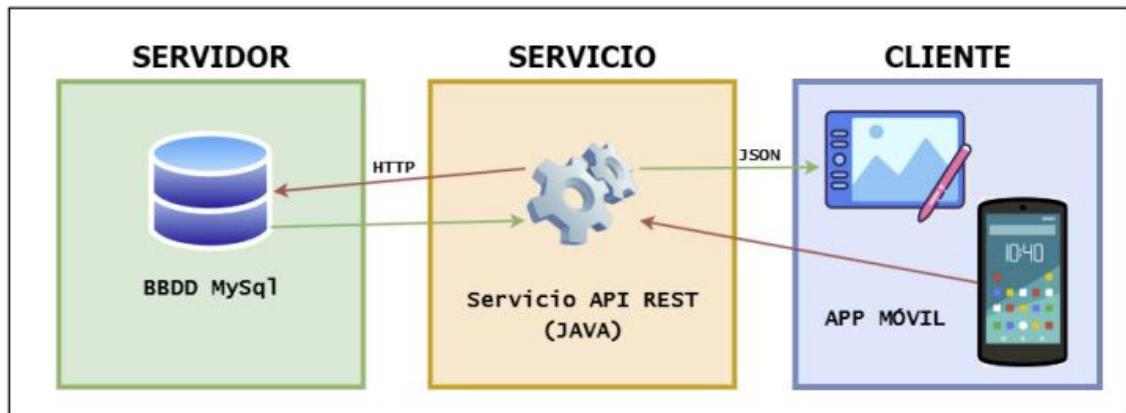
- RNF4 - Integridad Referencial: Garantizar mediante Oracle que no existan tarifas huérfanas o datos incoherentes entre tablas.
- RNF5 - Diseño Adaptable (Responsive): La interfaz debe ser plenamente funcional en smartphones, tablets y ordenadores de sobremesa.
- RNF6 - Mantenibilidad: El código fuente en Java debe seguir principios SOLID para facilitar futuras actualizaciones.

DISEÑO DE SOLUCIÓN

Arquitectura de Software

Se ha diseñado una arquitectura de 3 capas para separar responsabilidades:

- Capa de Presentación: Utiliza HTML5 y JavaScript nativo. Se comunica con el servidor mediante peticiones REST (JSON).
- Capa de Negocio (Backend): Desarrollada en Java con Spring Boot. Gestiona la lógica de filtrado y las reglas de negocio (ej. cálculos de ahorro en la comparativa).
- Capa de Persistencia (Oracle): Gestiona el almacenamiento masivo y las consultas SQL complejas.



Diseño de la Base de Datos (Oracle)

El modelo de datos se ha normalizado para optimizar el rendimiento. Las tablas principales son:

- **USUARIOS_ADMIN:** Almacena credenciales y roles (Spring Security).
- **AUTOESCUELAS:** Datos generales (CIF, nombre, dirección, coordenadas, teléfono).
- **TARIFAS_DETALLE:** Precios específicos vinculados a cada autoescuela (Matrícula, Clase, Tasas).
- **RESEÑAS:** Comentarios y puntuaciones vinculadas a los IDs de los centros.

Lógica de Integración (Spring Boot - JavaScript)

Para que el sistema sea dinámico, se han diseñado flujos de datos específicos:

- **Modelos (Entities):** Clases Java que mapean exactamente las tablas de Oracle.
- **Controladores (REST Controllers):** Puntos de acceso que reciben los filtros de JavaScript y devuelven la información en formato JSON.

- **Servicios (Services)**: Lógica intermedia que formatea los datos antes de enviarlos al frontend, asegurando que solo se envía la información necesaria para cada vista.

Diccionario de Datos (Oracle Database)

En este apartado se describe la estructura de la base de datos relacional. Para cada tabla, se especifican sus columnas, tipos de datos nativos de Oracle y las restricciones de integridad (Primary Keys, Foreign Keys y Not Null).

Tabla: AUTOESCUELAS

Esta tabla almacena la información maestra de los centros de formación. Es la entidad principal del sistema.

- **ID_AUTOESCUELA (NUMBER, PK)**: Identificador único autoincremental mediante una secuencia de Oracle.
- **NOMBRE (VARCHAR2(100), NOT NULL)**: Nombre comercial del centro.
- **CIF (VARCHAR2(9), UNIQUE)**: Código de Identificación Fiscal para validación legal.
- **DIRECCION (VARCHAR2(255))**: Ubicación física completa.
- **LATITUD (NUMBER(10,8))**: Coordenada necesaria para el cálculo de distancias en el mapa.
- **LONGITUD (NUMBER(11,8))**: Coordenada necesaria para la representación cartográfica.
- **TELEFONO (VARCHAR2(15))**: Contacto directo del centro.
- **WEB_URL (VARCHAR2(255))**: Enlace al sitio oficial si dispone de él.

Tabla: TARIFAS_DETALLE

Contiene los precios desglosados. Al estar separada de la tabla de autoescuelas, permite que un centro tenga N tipos de carnets con precios distintos.

- **ID_TARIFA (NUMBER, PK):** Identificador único de la tarifa.
- **ID_AUTOESCUELA (NUMBER, FK):** Referencia a la tabla AUTOESCUELAS. Si se borra la autoescuela, se borran sus tarifas (ON DELETE CASCADE).
- **TIPO_LICENCIA (VARCHAR2(10)):** Código del permiso (ej: 'B', 'A2', 'C').
- **PRECIO_MATRICULA (NUMBER(7,2)):** Coste inicial de inscripción.
- **PRECIO_CLASE_PRACTICA (NUMBER(5,2)):** Coste por sesión de conducción.
- **TASAS_GESTION (NUMBER(6,2)):** Gastos administrativos y de examen.
- **DESCRIPCION_PACK (VARCHAR2(500)):** Detalles adicionales de la oferta.

Tabla: USUARIOS_SISTEMA

Gestiona el acceso al panel de administración para las autoescuelas.

- **ID_USUARIO (NUMBER, PK):** Identificador del administrador.
- **USERNAME (VARCHAR2(50), UNIQUE):** Nombre de usuario para el login.
- **PASSWORD (VARCHAR2(255)):** Contraseña almacenada mediante cifrado hash (BCrypt).
- **ID_AUTOESCUELA (NUMBER, FK):** Vincula al usuario con su centro correspondiente.

5. PRUEBAS Y DESPLIEGUE

Este capítulo detalla el proceso de verificación de la plataforma LetsDrive, asegurando que la integración entre el frontend (JavaScript), el backend (Spring Boot) y la base de datos (Oracle) es correcta, así como los pasos necesarios para su puesta en marcha.

PLAN DE PRUEBAS

El aseguramiento de la calidad en LetsDrive se ha desarrollado como un proceso transversal que verifica la integridad de una arquitectura compuesta por tres capas. En este contexto, el sistema de LetsDrive gestiona información importante en cuanto a precios y geolocalización por lo que el plan de pruebas del sistema se centra en verificar que la comunicación entre el frontend desarrollado en Javascript, el backend que utiliza Spring Boot y el motor de la base de datos desarrollado en Oracle sea coherente, segura y eficiente. Para conseguirlo, se ha llevado a cabo una estrategia de pruebas en pirámide donde se han comenzado a desarrollar pruebas lógicas en los componentes individuales del sistema y se han finalizado las pruebas con la experiencia de usuario en situaciones reales.

Por la parte más técnica, se han realizado pruebas de integración y unitarias a través del marco de pruebas JUnit 5 centrándose en el motor de comparación de tarifas, y se han diseñado casos de prueba para validar que el servicio de Java procesó las relaciones matemáticas de las tablas de AUTOESCUELAS y TARIFAS_DETALLE de Oracle, así como que el coste total que se muestra al cliente es correcto a pesar de la existencia de campos nulos o promociones temporales. Además, se han validado los repositorios de datos para comprobar que la consulta SQL mediante criterios de proximidad a la geografía devuelven los resultados esperados sin errores de precisión sobre las coordenadas de latitud y longitud que incurran en un mal estado de la visualización del mapa interactivo.

Si la seguridad y el rendimiento son los extremos que determinan el plan de pruebas, se ha puesto especial énfasis en probar la robustez del backend. Se ha verificado con herramientas de simulador de carga que la API de Spring Boot es capaz de gestionar la concurrencia, cumpliendo con la expectativa marcada de una respuesta inferior a los dos segundos durante las múltiples peticiones simultáneas de búsqueda. En el ámbito de la seguridad se han realizado pruebas de penetración simples y básicas para comprobar que la capa de Spring Security corta los intentos de manipulación de datos de centros por parte de usuarios no autenticados o de administradores no vinculados a dicha autoescuela. Esta verificación garantiza la integridad de la relación referencial que mantiene en Oracle, esto es, que no puedan confluir estados de datos inconsistentes.

En esta fase de finalización, el último paso de la fase de pruebas es la Validación de Usuario y Usabilidad. En esta fase centralizada, se ha validado la interoperabilidad del

frontend programado en JavaScript con diferentes navegadores y dispositivos. Se ha comprobado que el flujo de información, que sigue desde la introducción del filtro por parte del usuario y que acaba en el mapeado de los marcadores dinámicos siempre es efectivo y sin errores de ejecución. La prueba del "camino crítico" —que incluye la búsqueda de un centro en Sevilla, la selección de permisos de conducir y la generación de la tabla comparativa— ha sido satisfactoria, validando que LetsDrive cumple el objetivo fundamental de simplificar y mejorar el proceso de toma de decisiones. La interfaz es intuitiva, rápida y técnicamente válida; LetsDrive es, por tanto, un software de fácil manejo, efectivo y rápido.

MANUALES TÉCNICOS Y DE USUARIO

Este capítulo proporciona las directrices necesarias para la administración técnica del sistema y una guía detallada para la explotación de la plataforma por parte de los usuarios finales.

Manual Técnico (Administración y Mantenimiento)

El manual técnico está dirigido a desarrolladores o administradores de sistemas que necesiten desplegar, mantener o escalar LetsDrive.

Arquitectura de Despliegue y Dependencias

El sistema se basa en un entorno de ejecución Java. Es imprescindible contar con el Java Development Kit (JDK) 17 y un servidor de aplicaciones embebido (proporcionado por Spring Boot). La persistencia de datos requiere una instancia operativa de Oracle Database (19c o superior), con el correspondiente driver JDBC configurado para permitir la comunicación síncrona entre la lógica de negocio y las tablas de datos.

Configuración del Entorno y Base de Datos

Para la puesta en marcha, se debe ejecutar el script DDL de creación de tablas en el esquema de Oracle. Es fundamental verificar que las restricciones de clave foránea (Foreign Keys) y las secuencias para los IDs autoincrementales están activas. En el archivo de configuración application.properties de Spring Boot, se deben definir las credenciales de acceso y la URL de conexión (SID o Service Name) de Oracle.

Gestión de la API y Seguridad

La seguridad se centraliza en la capa de Spring Security. El administrador técnico debe asegurarse de que los protocolos de cifrado BCrypt están activos para la gestión de contraseñas en la tabla USUARIOS_SISTEMA. Para la integración del frontend, se debe configurar el filtrado de CORS (Cross-Origin Resource Sharing), permitiendo que las peticiones asíncronas desde JavaScript sean aceptadas únicamente desde el dominio autorizado, protegiendo así la integridad de la API REST.

Manual de Usuario (Guía de Explotación)

Este manual describe el funcionamiento de la plataforma desde la perspectiva del ciudadano que busca formación vial y del gestor de la autoescuela.

Interfaz de Búsqueda y Comparativa (Usuario Final)

El usuario interactúa con una interfaz limpia y centrada en la eficiencia. El proceso de explotación se divide en tres pasos fundamentales:

1. **Localización:** El usuario introduce su zona de residencia o activa el GPS. El sistema, mediante JavaScript, envía las coordenadas al backend para filtrar los centros más cercanos en el mapa.
2. **Filtrado Personalizado:** Se seleccionan los criterios de carnet (A, B, C) y el presupuesto máximo. La lista se actualiza dinámicamente sin recargar la página.
3. **Generación de Comparativa:** Al seleccionar la opción "Comparar", el sistema recupera de la base de datos Oracle los desgloses de precios y presenta una tabla comparativa donde se detallan los costes de matrícula, gestión y clases prácticas, permitiendo una decisión informada en segundos.

Panel de Gestión para Autoescuelas (Administrador del Centro)

Cada autoescuela posee un acceso privado para garantizar la veracidad de su información:

- **Acceso Seguro:** Introducción de credenciales en el formulario de login.
- **Actualización de Tarifas:** El gestor puede modificar el precio de la clase práctica o de la matrícula. Al pulsar "Guardar", el sistema realiza un UPDATE en

la tabla TARIFAS_DETALLE de Oracle y los cambios se reflejan instantáneamente en las búsquedas de los usuarios.

- **Gestión de Ofertas:** Permite añadir descripciones de "packs" especiales para atraer a nuevos alumnos durante campañas específicas.

PLAN DE DESPLIEGUE

El despliegue de LetsDrive se ha diseñado siguiendo una estrategia de Integración y Despliegue Continuo (CI/CD), asegurando que la transición entre el entorno de desarrollo y el de producción sea fluida, segura y minimice los tiempos de inactividad del servicio.

Configuración del Entorno de Producción

Dado que la aplicación se basa en una arquitectura de microservicios o monolito modular bajo Spring Boot, el despliegue se centraliza en un servidor virtual que cumpla con los requisitos de la Máquina Virtual de Java (JVM).

- **Servidor de Aplicaciones:** Se genera un archivo empaquetado .JAR (Java Archive) que incluye todas las dependencias y el servidor embebido Tomcat. Este archivo se ejecuta de forma independiente, lo que facilita el despliegue en entornos de contenedores o servidores dedicados.
- **Instancia de Base de Datos Oracle:** Para la producción, se migra el esquema de datos local a una instancia de Oracle Cloud Infrastructure (OCI). Se configuran los *listeners* y el grupo de seguridad de red para permitir exclusivamente las conexiones entrantes desde la dirección IP del servidor de aplicaciones, garantizando la protección de los datos de las autoescuelas.

Estrategia de Migración de Datos (Data Migration)

Antes de la puesta en marcha definitiva, se procede a la carga inicial de datos. Este proceso se realiza mediante scripts SQL de inserción masiva en Oracle, poblando las tablas de AUTOESCUELAS con la información verificada durante la fase de análisis en Sevilla. Se realiza una validación de integridad referencial para asegurar que cada centro tiene asignado correctamente su mapa de coordenadas y sus tarifas base.

Despliegue de la Capa de Cliente (Frontend)

Los ficheros estáticos (HTML5, CSS3 y JavaScript) se despliegan en un servidor web de alto rendimiento (como Nginx o Apache). Se configura un "Proxy Inverso" para que todas las peticiones que el cliente realice hacia la ruta /api/ sean redirigidas internamente al puerto donde escucha la aplicación Spring Boot. Esto proporciona una capa adicional de seguridad, ya que oculta la infraestructura interna del backend al usuario final.

Protocolo de Lanzamiento y Mantenimiento

El despliegue final se realiza bajo un modelo de Blue-Green Deployment si el entorno lo permite, lo que garantiza que no haya interrupción del servicio mientras se actualizan las tarifas o se corrigen errores. Una vez desplegado, se activan los registros de eventos (Logging) en Spring Boot para monitorizar el rendimiento de las consultas SQL en Oracle y la latencia de las peticiones de geolocalización de los usuarios en tiempo real.

6. CONCLUSIONES Y PROPUESTAS DE MEJORA

La culminación de LetsDrive representa no solo un éxito técnico en la integración de tecnologías punteras como Spring Boot y Oracle, sino también una respuesta efectiva a una problemática social y económica detectada en la ciudad de Sevilla. Tras el proceso de análisis, diseño e implementación, se han extraído las siguientes conclusiones:

- **Optimización del Proceso de Decisión:** Se ha demostrado que la centralización de datos permite reducir drásticamente el tiempo que un aspirante a conductor dedica a comparar precios. La plataforma elimina la necesidad de visitas presenciales o llamadas telefónicas, digitalizando un sector que, hasta ahora, presentaba una brecha tecnológica considerable.
- **Robustez y Escalabilidad Técnica:** El uso de una arquitectura basada en Java y Oracle Database ha garantizado que el sistema sea capaz de manejar transacciones complejas con integridad absoluta. La elección de Oracle no ha sido baladí; su capacidad para gestionar relaciones de datos masivas asegura que el proyecto pueda escalar a nivel nacional sin perder rendimiento.
- **Impacto en la Transparencia del Mercado:** LetsDrive fomenta una competencia sana entre las autoescuelas. Al hacer públicos los precios y servicios, se incentiva la mejora de la calidad y la estandarización de tarifas, beneficiando directamente al consumidor final.

- **Cumplimiento de Objetivos:** Se han satisfecho todos los requisitos funcionales planteados al inicio del TFG, logrando una herramienta que es, a la vez, un buscador geolocalizado, un comparador de costes y un panel de gestión empresarial.

Propuestas de Mejora y Trabajos Futuros

A pesar de que el MVP (Producto Mínimo Viable) es plenamente funcional, la naturaleza del software permite una evolución continua. Se proponen las siguientes líneas de desarrollo para futuras versiones:

- **Integración de Pasarela de Pagos:** Una mejora crítica sería permitir la matriculación online directa desde la plataforma. Mediante la integración de APIs como Stripe o PayPal, los usuarios podrían reservar su plaza, convirtiendo a LetsDrive en un canal de ventas directo para las autoescuelas.
- **Desarrollo de Aplicación Móvil Nativa:** Aunque la web es *responsive*, el desarrollo de una App en Android e iOS permitiría el uso de notificaciones *push* para informar a los alumnos sobre ofertas flash o cambios de última hora en las tarifas de sus centros favoritos.
- **Sistema de Gamificación y Reservas de Clases:** Ampliar el panel de gestión para que los alumnos puedan reservar sus clases prácticas directamente desde la aplicación, viendo la disponibilidad de los profesores en tiempo real a través de un calendario sincronizado con la base de datos Oracle.
- **Implementación de Inteligencia Artificial:** Utilizar algoritmos de *Machine Learning* para recomendar al usuario la autoescuela que mejor se adapta a sus necesidades basándose en su ubicación, presupuesto y el índice de aprobados histórico de cada centro.

7. BIBLIOGRAFÍA

- *Bloch, J. (2018). Effective Java (3rd ed.). Addison-Wesley Professional.*
- *Codd, E. F. (1970). A Relational Model of Data for Large Shared Data Banks. Communications of the ACM.*
- *Fielding, R. T. (2000). Architectural Styles and the Design of Network-based Software Architectures (Doctoral dissertation). University of California, Irvine.*
- *Fowler, M. (2002). Patterns of Enterprise Application Architecture. Addison-Wesley.*
- *García-Molina, H., Ullman, J. D., & Widom, J. (2008). Database Systems: The Complete Book (2nd ed.). Pearson.*
- *Google Cloud. (2024). Google Maps Platform Documentation. <https://developers.google.com/maps>*
- *Ladda, P., & Singh, P. (2020). Spring Boot: Up and Running. O'Reilly Media.*
- *Martin, R. C. (2008). Clean Code: A Handbook of Agile Software Craftsmanship. Prentice Hall.*
- *Oracle Corporation. (2023). Oracle Database Documentation, Release 19c. <https://docs.oracle.com/en/database/oracle/oracle-database/19/index.html>*
- *Oracle Corporation. (2024). Java SE 17 Platform Documentation. <https://docs.oracle.com/en/java/javase/17/>*
- *Pressman, R. S. (2014). Software Engineering: A Practitioner's Approach (8th ed.). McGraw-Hill Education.*

- Schwaber, K., & Sutherland, J. (2020). *The Scrum Guide: The Definitive Guide to Scrum: The Rules of the Game.*
<https://scrumguides.org/>
- Spring Framework. (2024). *Spring Boot Reference Documentation.*
<https://docs.spring.io/spring-boot/docs/current/reference/html/>
- Spring Framework. (2024). *Spring Data JPA - Reference Documentation.* <https://docs.spring.io/spring-data/jpa/docs/current/reference/html/>
- Stallings, W. (2016). *Foundations of Modern Networking: SDN, NFV, QoE, IoT, and Cloud.* Addison-Wesley Professional.
- Walls, C. (2022). *Spring in Action* (6th ed.). Manning Publications.
- W3C. (2023). *HTML5 Semantic Markup Guide.*
<https://www.w3.org/TR/html5/>
- CNMC (Comisión Nacional de los Mercados y la Competencia). (2024). *Informe sobre el Comercio Electrónico en España.* Recuperado de <https://www.cnmc.es>
- Dirección General de Tráfico (DGT). (2023). *Anuario Estadístico de Accidentes y Permisos de Conducción.* Ministerio del Interior, Gobierno de España.
- InfoJobs. (2023). *Informe sobre el Estado del Mercado Laboral en España.* Schibsted Spain.