



## Glosario

Integrantes:

Mathías Fernández

Eduardo Flores

Federico Acosta

Gabriel Suárez

Pablo Perdomo

Alexander Rodriguez

Tutor:

Fernando Arrieta

**Actor:** Entidad externa (persona o sistema) que interactúa con el sistema. En este proyecto se identifican tres actores principales: Administrador, Vendedor y Cliente.

**Administrador:** Tipo de usuario con permisos de gestión sobre otros usuarios.

**Alta:** Acción de registrar un nuevo elemento en el sistema (usuarios, ómnibus, localidades, viajes, etc.).

**Android:** Sistema Operativo de código abierto móvil ampliamente utilizado (ej. Samsung, Xiaomi, Nokia, Huawei, etc.)

**API:** Interfaz de Programación de Aplicaciones (del inglés: Application Programming Interface)

**API REST:** Interfaz de programación que permite la comunicación entre cliente y servidor utilizando el protocolo HTTP siguiendo principios REST (por ejemplo, GET, POST, PUT, DELETE).

**Arquitectura de software:** Estructura global del sistema que define sus componentes, relaciones, y principios de diseño.

**Arquitectura en capas:** Estilo de diseño donde la aplicación se divide en niveles (presentación, negocio, datos), cada uno con una responsabilidad clara.

**Autenticación:** Proceso mediante el cual un usuario accede al sistema mediante unas credenciales preestablecidas.

**Backend:** Parte del sistema encargada de la lógica de negocio, conexión con la base de datos y procesamiento.

**Baja:** Acción de desactivar o eliminar un elemento del sistema (por ejemplo, usuarios).

**Base de datos relacional:** Tipo de base de datos que organiza la información en tablas con relaciones definidas entre ellas mediante claves.

**Bot:** Programa informático que realiza tareas de forma automática y repetitiva.

**Campo obligatorio:** Atributo que debe contener un valor no nulo en los registros (por ejemplo, cédula del usuario).

**Capa de Datos:** Nivel de la arquitectura responsable de la persistencia, recuperación y manejo de la información almacenada.

**Capa de Negocio:** Componente central del sistema que contiene la lógica de negocio y gestiona los procesos internos.

**Capa de Presentación:** Nivel de la arquitectura que interactúa directamente con el usuario final, ya sea web o móvil.

**Casos de uso:** Representaciones funcionales que describen cómo los actores interactúan con el sistema para alcanzar un objetivo.

**Clave foránea** (foreign key): Atributo que establece una relación con otra tabla, permitiendo enlazar datos de forma referencial.

**Clave primaria** (primary key): Atributo o conjunto de atributos que identifican de manera única cada registro en una tabla.

**Cobertura de pruebas:** Porcentaje de componentes o funcionalidades del sistema que han sido efectivamente validados mediante pruebas.

**Componente:** Unidad funcional dentro de una arquitectura que encapsula cierta funcionalidad del sistema (por ejemplo, un servicio o un controlador).

**Constraint:** Restricción que impone reglas sobre los valores permitidos en una tabla (por ejemplo, unicidad, no nulo, formato).

**Contenedor** (Docker): Entorno aislado que permite ejecutar aplicaciones y servicios de manera portátil y reproducible en distintas máquinas.

**Controlador:** Componente del backend que expone endpoints HTTP. Es probado para verificar el correcto manejo de las peticiones del usuario.

**CRUD:** Crear, Leer, Actualizar y Eliminar datos (del inglés: Create, Read, Update, Delete)

**CSV:** Formato de archivos con valores separados por coma (del inglés: Comma-Separated Values)

**Despliegue** (Deployment): Proceso de instalación y ejecución del sistema en un entorno específico.

**DevOps:** Área que combina el desarrollo (Dev) y las operaciones de TI (Ops).

**E2E** (End-to-End): Tipo de prueba automatizada que valida flujos funcionales completos desde la perspectiva del usuario, de principio a fin.

**Endpoint:** Ruta accesible a través de HTTP que permite ejecutar una operación específica del sistema (ej. /api/usuarios).

**Entidad:** Representación persistente de un objeto de negocio.

**Entorno de desarrollo:** Conjunto de herramientas y configuraciones utilizadas para programar, compilar y ejecutar una aplicación durante su fase de construcción.

**Error funcional:** Comportamiento del sistema que no cumple con lo definido en los requerimientos, detectado a través de pruebas.

**Frontend:** Parte visible del sistema con la que el usuario interactúa directamente.

**Funcionalidad:** Función que un software debe satisfacer para cumplir con la necesidad del usuario.

**GUI:** Interfaz Gráfica de Usuario (del inglés: Graphic User Interface)

**Interfaz de usuario (UI):** Conjunto de elementos visuales y controles con los que el usuario interactúa en una aplicación.

**iOS:** Sistema Operativo de código cerrado desarrollado por Apple para sus propios dispositivos.

**Iteración:** Ciclo de desarrollo dentro del proyecto en el que se construye y entrega una parte funcional del sistema.

**JUnit:** Framework de pruebas unitarias para aplicaciones Java, utilizado para validar la lógica de negocio del backend.

**JWT (JSON Web Token):** Método de autenticación que permite verificar la identidad de un usuario mediante un token en formato JSON firmado.

**Lógica de negocio:** Reglas y procedimientos que definen el funcionamiento interno del sistema, principalmente en la capa de servicios.

**Middleware:** Software intermedio que permite la conexión entre diferentes aplicaciones o servicios dentro del sistema.

**Mock:** Objeto simulado que reemplaza dependencias reales en pruebas, utilizado para controlar y verificar interacciones específicas.

**Mockito:** Biblioteca de Java para simular (mockear) dependencias durante pruebas unitarias, permitiendo aislar el componente bajo prueba.

**Mockup:** Representación visual de la interfaz de usuario de una aplicación, utilizada para definir la estructura, diseño y flujo antes de su desarrollo real.

**Mobile:** Aplicable a teléfonos celulares (móviles).

**MongoDB:** Base de datos NoSQL orientada a documentos. Fue considerada en la evaluación, aunque no seleccionada como tecnología final.

**MVC (Model-View-Controller):** Patrón de arquitectura que separa la lógica de presentación (Vista), la lógica de negocio (Modelo) y el manejo de eventos (Controlador).

**Node.js:** Entorno de ejecución para JavaScript en el servidor, utilizado para aplicaciones web escalables. Considerado durante las pruebas de concepto del frontend.

**Normalización:** Proceso de estructurar una base de datos para reducir redundancia y mejorar integridad mediante la división en múltiples tablas relacionadas.

**ORM (Object-Relational Mapping):** Técnica que permite mapear estructuras de objetos del lenguaje de programación a una base de datos relacional.

**Persistencia:** Almacenamiento duradero de los datos en la base de datos, de forma que sobrevivan al cierre del sistema.

**Playwright:** Herramienta de automatización para pruebas de extremo a extremo sobre interfaces web.

**PoC (Proof of Concept):** Implementación mínima que valida la viabilidad técnica de una tecnología, arquitectura o enfoque antes de adoptarla formalmente.

**Prueba automatizada:** Test ejecutado mediante herramientas de software sin intervención humana directa.

**Prueba funcional:** Verificación del comportamiento del sistema desde la perspectiva del usuario, comparando resultados con lo esperado.

**Prueba manual:** Validación realizada por una persona sin herramientas de automatización, especialmente útil en móviles.

**Prueba unitaria:** Verificación de una clase o método en forma aislada, enfocada en su correcto funcionamiento individual.

**PW:** Panel Web

**Requerimiento funcional:** Condición o capacidad específica que el sistema debe cumplir para satisfacer las necesidades del usuario.

**Requerimiento no funcional:** Restricción o atributo de calidad que el sistema debe cumplir (usabilidad, rendimiento, seguridad, etc.).

**Responsive:** Diseño web adaptativo que permite visualizar correctamente una interfaz en distintos dispositivos (PC, tablets, móviles).

**Servicio (Service):** Componente del backend que contiene lógica de negocio.

**Sprint:** Período de tiempo predeterminado en el que un equipo trabaja para completar una cantidad de trabajo establecida.

**STH:** Sistema de Ticketing de Horarios

**Token de acceso:** Cadena que representa la autorización del usuario para acceder a recursos protegidos del sistema, emitida tras el login.

**Token de autenticación:** Valor que identifica a un usuario autenticado y permite el acceso a funcionalidades restringidas del sistema.

**Usabilidad:** Facilidad con la que los usuarios pueden aprender y operar el sistema.

**UX:** Experiencia de Usuario (del inglés: User eXperience)