

Manual Técnico: EasyOrder

Alejandro Bernal Bryan
Carbajal Mendoza Eduardo
Martinez Nicolas Francisco Leonardo
Romero Razo Alexis Daniel

Equipo 1

**UPIICSA – Unidad Profesional Interdisciplinaria de Ingeniería y Ciencias
Sociales y Administrativas IPN**

Tabla de Contenido

Contents

1	Introducción	3
2	Arquitectura de la aplicación	3
2.1	Frontend (Aplicación móvil)	3
2.1.1	Componentes clave del frontend	3
3	Backend	4
3.1	Componentes clave del backend	4
3.2	Esquema de la base de datos	4
4	Desarrollo de la API	5
4.1	Endpoints principales	5
4.2	Autenticación y seguridad	5
5	Tecnologías utilizadas	6
6	Despliegue e infraestructura	7
6.1	Flujo de despliegue	7
7	Pruebas	8
7.1	Pruebas de la API	8
7.2	Pruebas de la interfaz de usuario	8
8	Conclusiones	9
9	Webgrafía	9

1 Introducción

Este manual está destinado a proporcionar una visión técnica completa de la aplicación **EasyOrder**, diseñada para optimizar el proceso de pedidos y control de stock en bares y restaurantes. El manual cubrirá los aspectos técnicos de la infraestructura, las tecnologías utilizadas y las estrategias de desarrollo implementadas.

2 Arquitectura de la aplicación

La arquitectura de **EasyOrder** sigue un enfoque de **arquitectura cliente-servidor** en el que la aplicación móvil interactúa con un servidor backend para procesar y almacenar datos.

2.1 Frontend (Aplicación móvil)

La aplicación móvil está desarrollada utilizando **Android Studio** con el lenguaje de programación **Java** y **Kotlin**. Utiliza el diseño de interfaz de usuario (UI) basado en **Material Design** para garantizar una experiencia fluida y amigable para el usuario.

2.1.1 Componentes clave del frontend

- **Activity y Fragments:** Controlan las pantallas de la aplicación y sus interacciones.
- **RecyclerView:** Para mostrar las listas de productos y pedidos.
- **ViewModel y LiveData:** Utilizados para gestionar la UI y las operaciones en segundo plano.
- **Retrofit:** Para la comunicación HTTP con el servidor backend.

3 Backend

El backend de **EasyOrder** está basado en una arquitectura RESTful y utiliza **Node.js** con **Express.js** como framework para manejar las solicitudes HTTP y la lógica del servidor. La base de datos es **MySQL**, que se usa para almacenar los datos de usuarios, pedidos, productos y el historial de transacciones.

3.1 Componentes clave del backend

- **Express.js:** Framework para la creación de APIs RESTful.
- **JWT (JSON Web Token):** Para la autenticación segura de usuarios y administradores.
- **MySQL:** Sistema de gestión de bases de datos relacional utilizado para almacenar datos persistentes.
- **Sequelize:** ORM utilizado para interactuar con la base de datos de manera eficiente.

3.2 Esquema de la base de datos

La base de datos de **EasyOrder** se organiza en varias tablas clave:

- **Usuarios:** Almacena la información de los usuarios, incluyendo su nombre, correo electrónico y contraseña cifrada.
- **Pedidos:** Contiene información sobre cada pedido realizado, incluyendo el usuario, productos, y estado del pedido.
- **Productos:** Información sobre los productos disponibles en el restaurante, como el nombre, precio y categoría.
- **Historial:** Registra el historial de pedidos realizados por cada usuario.

4 Desarrollo de la API

La API de **EasyOrder** está diseñada para manejar las operaciones CRUD (Crear, Leer, Actualizar, Eliminar) relacionadas con los pedidos y los usuarios. A continuación, se describen algunos de los endpoints clave.

4.1 Endpoints principales

- POST /api/usuarios: Registra un nuevo usuario.
- POST /api/login: Inicia sesión de usuario y devuelve un JWT.
- GET /api/pedidos: Obtiene todos los pedidos realizados por un usuario.
- POST /api/pedidos: Crea un nuevo pedido.
- PUT /api/pedidos/id: Actualiza el estado de un pedido.

4.2 Autenticación y seguridad

La autenticación se maneja mediante **JWT**. Al iniciar sesión, el servidor genera un token JWT que el cliente utiliza para autenticar futuras solicitudes. El token se valida en cada solicitud utilizando el middleware de autenticación en Express.js.

5 Tecnologías utilizadas

La aplicación **EasyOrder** está construida utilizando una variedad de tecnologías modernas para garantizar un rendimiento óptimo y una experiencia de usuario excelente.

- **Frontend:**
 - Android Studio
 - Java/Kotlin
 - Retrofit (para la comunicación con la API)
 - Material Design (para el diseño de la UI)
- **Backend:**
 - Node.js
 - Express.js
 - MySQL
 - Sequelize (ORM)
 - JWT (para autenticación)
- **Herramientas adicionales:**
 - Git (control de versiones)
 - Postman (para pruebas de la API)
 - Firebase (para notificaciones push)

6 Despliegue e infraestructura

La aplicación se despliega utilizando servicios en la nube para asegurar la disponibilidad y escalabilidad. Se utiliza **Heroku** para el despliegue del backend y **Firebase** para el manejo de notificaciones push.

6.1 Flujo de despliegue

- El código se gestiona en **GitHub**.
- El backend se despliega automáticamente en **Heroku** a través de integración continua (CI).
- La base de datos MySQL está alojada en **Amazon RDS**.
- Las notificaciones push se gestionan mediante **Firebase Cloud Messaging**.

7 Pruebas

Para garantizar la calidad del software, se implementan pruebas automatizadas utilizando **Jest** y **Mocha** en el backend, y **Espresso** en el frontend.

7.1 Pruebas de la API

Las pruebas unitarias y de integración se realizan utilizando **Jest** y **Mocha** para asegurar que los endpoints funcionen correctamente y que los datos se manejen de manera segura.

7.2 Pruebas de la interfaz de usuario

Se utilizan **Espresso** y **UI Automator** para realizar pruebas automatizadas en la aplicación Android, asegurando que la UI se comporte correctamente en diferentes dispositivos y versiones de Android.

8 Conclusiones

El desarrollo de **EasyOrder** ha sido un proceso desafiante pero gratificante. Utilizando una arquitectura moderna basada en tecnologías escalables, la aplicación está diseñada para ser fácilmente mantenible y extensible en el futuro. La integración de notificaciones push, la autenticación segura mediante JWT y el uso de bases de datos en la nube aseguran que **EasyOrder** sea una solución eficiente y robusta para bares y restaurantes.

9 Webgrafía

- Node.js (<https://nodejs.org>) último acceso 01/03/2024
- Express.js (<https://expressjs.com>) último acceso 01/03/2024
- MySQL (<https://www.mysql.com>) último acceso 01/03/2024
- Firebase (<https://firebase.google.com>) último acceso 01/03/2024
- Android Developer (<https://developer.android.com>) último acceso 01/03/2024