

# EASYORDER - DOCUMENTACIÓN DEL PROYECTO

---



Daniel Escobar Molina 2ºDAM DUAL

## ÍNDICE

1. 1 Introducción
2. 2 Justificación del proyecto
3. 3 Aspectos generales (funcionamiento, versiones,...)
4. 4 Contenidos
5. 4.1 Acceso a datos
6. 4.2 Base de datos
7. 4.3 Despliegue
8. 4.4 Interfaz gráfica
9. 5 Análisis del proyecto (descripción detallada)
10. 6 Desarrollo e implementación
11. 7 Pruebas e incidencias
12. 8 Manual de usuario
13. 9 Preguntas frecuentes
14. 10 Conclusiones
15. 11 Referencias (bibliografía y web)

## 1 INTRODUCCIÓN

La digitalización en el sector de la hostelería ha crecido de forma acelerada en los últimos años. Con la necesidad de optimizar tiempos, reducir errores y mejorar la atención al cliente, los bares y restaurantes han comenzado a incorporar soluciones tecnológicas que faciliten la gestión de sus procesos internos.

En este contexto surge **EasyOrder**, una aplicación móvil multiplataforma cuyo objetivo principal es facilitar la organización y el seguimiento de pedidos dentro de establecimientos de restauración.

Esta documentación técnica recoge el desarrollo, análisis e implementación de EasyOrder, detallando tanto los aspectos funcionales como tecnológicos de la solución. La app ha sido diseñada con tecnologías modernas como **React Native** y **Firestore**, buscando ofrecer una experiencia eficiente, rápida y adaptable a distintos entornos de trabajo.

## 2 JUSTIFICACIÓN DEL PROYECTO

El proceso tradicional de gestión de pedidos en bares y restaurantes suele estar basado en papel, comunicación verbal y desplazamientos constantes entre mesas, barra y cocina. Este modelo puede dar lugar a errores de interpretación, pérdidas de tiempo y, en consecuencia, una menor calidad en el servicio.

EasyOrder nace con el propósito de dar respuesta a estas problemáticas mediante una herramienta tecnológica intuitiva y eficiente. Al permitir la toma de pedidos digitalmente desde el móvil, su envío directo a cocina y la gestión de estados en tiempo real, se mejora la comunicación interna, se reducen errores y se agilizan los tiempos de atención.

Además, el uso de una arquitectura basada en **Firestore** permite que los datos se sincronicen en tiempo real y que el sistema sea escalable, adaptable y de fácil mantenimiento. Por todo ello, se considera que el proyecto tiene una justificación sólida tanto desde el punto de vista técnico como funcional.

### 3 ASPECTOS GENERALES (FUNCIONAMIENTO, VERSIONES,...)

EasyOrder es una aplicación desarrollada con React Native, lo que permite su despliegue tanto en Android como en iOS. Su propósito es facilitar la toma y gestión de pedidos en bares y restaurantes desde dispositivos móviles.

La app está conectada a Firebase, que actúa como backend completo proporcionando servicios de base de datos en tiempo real, autenticación de usuarios y almacenamiento de archivos (por ejemplo, imágenes de los menús o datos adjuntos).

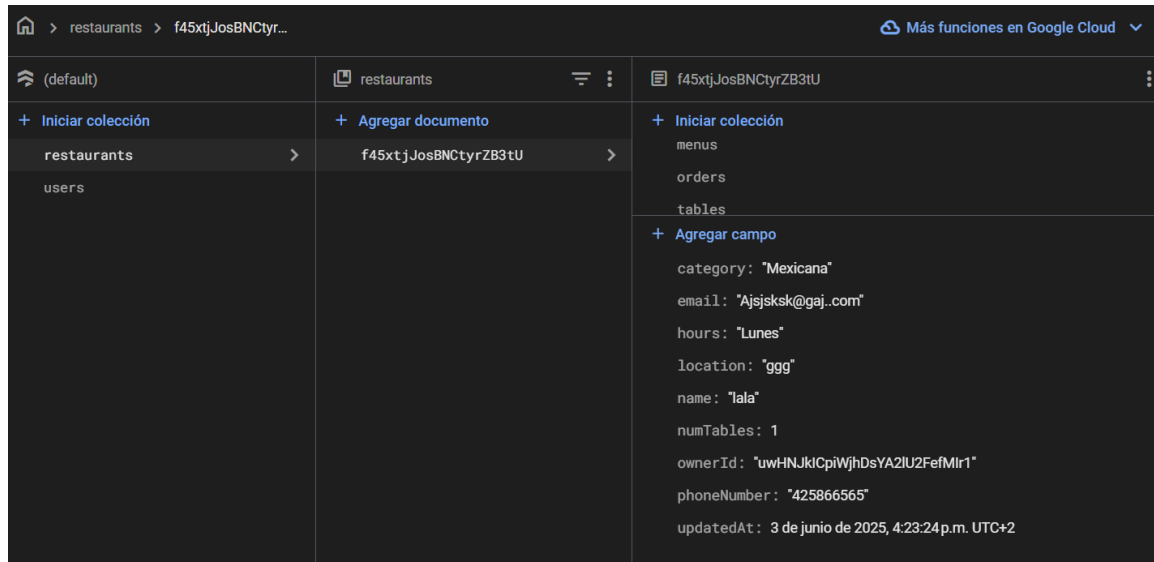
#### 4.1 ACCESO A DATOS

El acceso a datos se realiza mediante los servicios de Firebase, específicamente Cloud Firestore para los datos estructurados como productos, pedidos y usuarios.

Los datos se sincronizan en tiempo real entre los dispositivos y la base de datos en la nube. La autenticación de usuarios se gestiona mediante Firebase Authentication, usando métodos como correo electrónico/contraseña o inicio de sesión con cuentas de Google.

#### 4.2 BASE DE DATOS

La base de datos utilizada es Firebase Cloud Firestore. Su estructura no relacional está organizada por colecciones y documentos que representan usuarios, productos, mesas, pedidos, entre otros.



Este modelo facilita la escalabilidad y sincronización en tiempo real, especialmente útil en un entorno con múltiples camareros y clientes activos. La integridad de los datos se garantiza mediante reglas de seguridad personalizadas.

#### 4.3 DESPLIEGUE

El despliegue de EasyOrder se gestiona mediante Expo, lo que simplifica el proceso de construcción y distribución de la aplicación en dispositivos móviles. Las actualizaciones pueden realizarse sin necesidad de pasar por las tiendas oficiales mediante actualizaciones OTA (Over The Air).

No se requiere infraestructura propia para el backend gracias al uso de Firebase, lo que reduce los costes operativos y permite escalar automáticamente.

#### 4.4 INTERFAZ GRÁFICA

La interfaz gráfica de EasyOrder ha sido desarrollada con **React Native**, utilizando componentes personalizados y bibliotecas como **React Navigation**, **React Native Paper** y

**Styled Components** para mantener una experiencia de usuario moderna, intuitiva y coherente entre plataformas móviles.

Se han seguido principios de diseño responsive para adaptarse a diferentes resoluciones de pantalla. Además, se han incorporado íconos visuales y colores diferenciados para facilitar la navegación y la identificación rápida de funciones esenciales, como la creación de pedidos, asignación de mesas y seguimiento en cocina.

La experiencia de usuario (UX) ha sido validada mediante pruebas internas con usuarios de bares reales, lo que ha permitido realizar ajustes visuales y funcionales para mejorar la eficiencia en entornos de trabajo con alta carga operativa.

## 5 ANÁLISIS DEL PROYECTO (DESCRIPCIÓN DETALLADA)

El análisis del proyecto EasyOrder se ha centrado en cubrir los principales flujos de trabajo en bares y restaurantes, principalmente la gestión eficiente de pedidos, mesas y usuarios desde dispositivos móviles.

A nivel funcional, la aplicación permite a los camareros iniciar sesión mediante Firebase Authentication, seleccionar una mesa, consultar el menú (obtenido desde Firestore), añadir productos al pedido y enviarlo directamente a cocina. Los pedidos se sincronizan en tiempo real, lo que permite que el personal de cocina reciba las órdenes sin retrasos ni errores.

Desde un punto de vista técnico, el sistema se estructura en varios módulos clave:

- **Autenticación:** Mediante Firebase, con verificación de roles (camarero, cocina, administrador).
- **Pedidos y productos:** Gestionados como colecciones en Firestore, con estados que reflejan su ciclo de vida (pendiente, en preparación, servido).
- **Interfaz móvil:** Diseñada con React Native para una navegación fluida entre pantallas y acceso inmediato a funciones clave.

También se ha desarrollado una pequeña interfaz administrativa (web o móvil) para gestionar productos del menú, y detalles del restaurante.

El diseño técnico se complementa con diagramas de flujo y esquemas de datos para documentar los componentes principales y su interacción.

## 6 DESARROLLO E IMPLEMENTACIÓN

El desarrollo de EasyOrder se ha realizado utilizando **React Native**, un framework de desarrollo móvil que permite crear aplicaciones nativas para Android e iOS a partir de una única base de código en JavaScript/TypeScript. Esta decisión tecnológica ha permitido reducir tiempos de desarrollo y facilitar la mantenibilidad del proyecto.

Para la gestión del backend, se ha utilizado **Firebase**, integrando distintos servicios como:

- **Firebase Authentication** para el control de acceso y registro de usuarios.
- **Cloud Firestore** para la gestión de pedidos, usuarios, productos y mesas en tiempo real.
- **Firebase Storage** para almacenar imágenes del menú y otros recursos.
- **Firebase Cloud Messaging (FCM)** para enviar notificaciones push a los dispositivos.

El desarrollo se ha estructurado en sprints semanales bajo una metodología ágil. Se utilizó **Git** como sistema de control de versiones y **GitHub Projects** como tablero para organizar tareas, bugs y mejoras.

Para pruebas en dispositivos reales, se ha empleado **Expo**, facilitando la visualización de cambios y permitiendo la colaboración entre desarrolladores y testers sin necesidad de compilar versiones nativas constantemente.

El código se ha dividido en módulos reutilizables: autenticación, gestión de productos, gestión de pedidos y lógica de roles. Esto permite escalar y mantener fácilmente la aplicación en el futuro.

Se ha prestado especial atención a la eficiencia de las consultas en Firestore, al manejo de estados con herramientas como **React Context API** y a la consistencia visual mediante el uso de estilos centralizados.

## 7 PRUEBAS E INCIDENCIAS

Durante el desarrollo de EasyOrder, se han llevado a cabo diferentes tipos de pruebas para asegurar el correcto funcionamiento de la aplicación en múltiples escenarios y dispositivos.

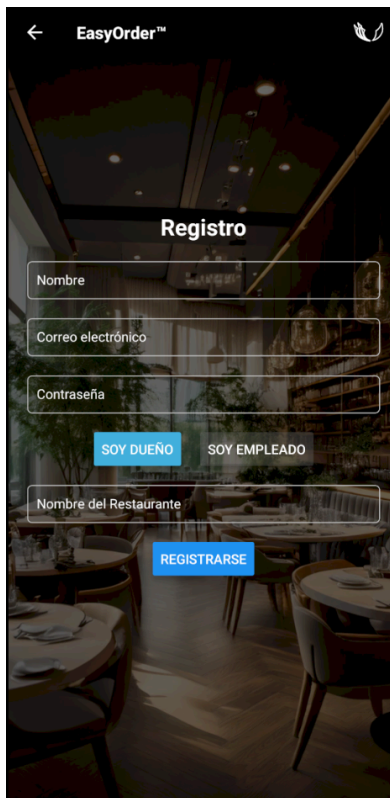
- **Pruebas unitarias:** Se realizaron sobre funciones críticas, como la validación de formularios, lógica de negocio (por ejemplo, cálculo de totales), y control de estados en la interfaz mediante React hooks y contextos.
- **Pruebas de integración:** Se verificó la interacción entre la interfaz y los servicios de Firebase (autenticación, Firestore, almacenamiento), así como la coherencia de los datos mostrados tras operaciones como agregar, modificar o eliminar un pedido.
- **Pruebas en dispositivos reales:** La aplicación fue desplegada y probada en dispositivos Android e iOS utilizando Expo y builds nativas.

## 8 MANUAL DE USUARIO

Para utilizar EasyOrder, el usuario debe registrarse con una cuenta, si va a ser dueño deberá simplemente poner el nombre del restaurante (Fig. 1) y si va a trabajar como empleado habrá que poner un código del restaurante a mano (Fig. 2), o escaneando un



qr que se encuentra en la parte de ajustes (Fig. 3), de “información del restaurante”(Fig. 4).



The screenshot shows the 'Registro' (Registration) screen of the EasyOrder app. The background is a dimly lit restaurant interior. The screen features a back arrow and the 'EasyOrder™' logo at the top. The registration form includes the following elements:

- Registro** (Registration) title
- Nombre** (Name) input field
- Correo electrónico** (Email) input field
- Contraseña** (Password) input field
- SOY DUEÑO** (I am the owner) button
- SOY EMPLEADO** (I am an employee) button
- Nombre del Restaurante** (Restaurant Name) input field
- REGISTRARSE** (Register) button

Fig 1. Registro del dueño

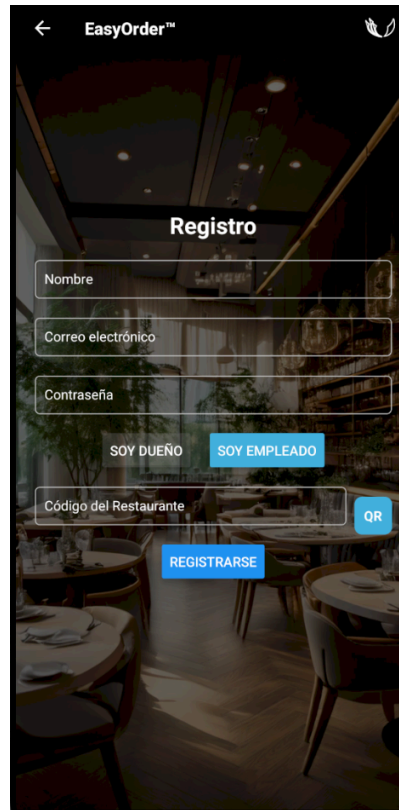


Fig 2. Registro del empleado



Fig 3. Botón ajustes y perfil



Fig 4. Ajustes y configuraciones

Después de haberse registrado correctamente en el restaurante, el siguiente paso es iniciar sesión con esa misma cuenta para poder usar la app.(Fig 5.)

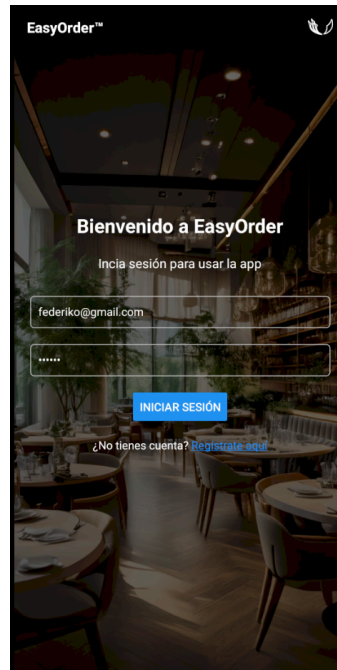


Fig 5. Inicio de sesión

Una vez logueado, el usuario tendrá la siguiente vista, en ella podemos ver el botón de ajustes y de perfil en la cabecera de la pantalla (Fig. 3). Si es la primera vez que entramos como dueño, habrá que realizar una configuración inicial con datos del restaurante (Fig. 6). Una vez realizada ya podemos usar la app. Para ver nuestras mesas, deberemos ir al botón de configuración (Fig. 3) y clicar “configuración de mesas” (Fig. 4).

En esta pantalla (Fig. 7) habrá que clicar en el botón azul de la esquina superior derecha, el cual va añadiendo mesas con cada click, estas mesas se pueden organizar como se desee, para eliminar una mesa basta con tocar el botón rojo de la esquina de una mesa.

Una vez realizada esta configuración, la guardamos con el botón “Guardar posiciones”, y volvemos a la pantalla principal.

En la pantalla principal (Fig. 8) se muestran las mesas que tenemos actualmente en nuestro restaurante, que se pueden ordenar de forma personalizada o automáticamente en filas de tres columnas.

The screenshot shows the 'Configuración del Restaurante' (Restaurant Configuration) screen. At the top is a black header with a back arrow and the word 'Información'. Below the header, the title 'Configuración del Restaurante' is displayed. The form contains several input fields: 'Nombre del Restaurante' with the value 'lala', 'Ubicación' with 'ggg', 'Teléfono' with '425866565', 'Email' with 'Ajsjsk@gaaj..com', 'Categoría' with 'Mexicana', and 'Horario' with 'Lunes'. At the bottom, there is a section for 'Código QR del Restaurante' which displays a QR code.

Fig. 6 Configuración del restaurante

The screenshot shows the 'Organización Personalizada' (Custom Organization) screen. It features a header with a back arrow, the title 'Organización Personalizada', and a blue circular button with a plus sign. The main area displays four table icons arranged in a 2x2 grid, each labeled 'Mesa 1', 'Mesa 3', 'Mesa 4', and 'Mesa 2' respectively. Each icon shows two people at a table with a red minus sign above it. At the bottom right, there is a blue button labeled 'GUARDAR POSICIONES'.

Fig. 7 Configuración de mesas



Fig 8. Pantalla principal

Al hacer click a una mesa aparecerá el siguiente menú (Fig. 9). En él aparecen todos los menús y comidas que ofrece el restaurante previamente creados.

En esta pantalla existe un buscador por nombre, un filtro por tipo de comida y por supuesto todos los platos y bebidas disponibles con unos botones para aumentar o decrementar la cantidad que pida nuestro cliente del menú pedido. Una vez seleccionada la cantidad y menú correctos habrá que clicar en “Crear pedido”.

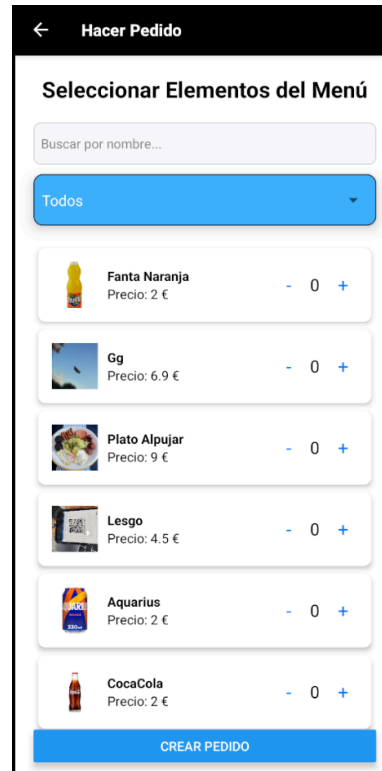


Fig. 9 Pantalla para hacer pedidos

Una vez creado el pedido volveremos a la pantalla principal y notaremos un cambio de color en la mesa que ha pedido de negro a **rojo**, esto significa que esa mesa a **pedido pero no ha llegado** el pedido a mesa. Si hacemos click a esa mesa nos llevará a una pantalla de detalles de ese pedido (Fig. 10). En esta pantalla podemos interactuar de varias formas, con los dos botones “servido” o “pagado”, que servirán para controlar de una manera más eficiente las mesas, “**servido**” pintará la mesa de verde e indicará que el **pedido ha llegado** a esa mesa, y “**pagado**” pintará la mesa de negro, simulando que **está vacía**. El resto de la pantalla indica los menús que ha pedido esa mesa y su respectivo precio para facilitar el cálculo del cobro, además de un botón para pedir más en el caso de que los clientes necesiten otro menú.

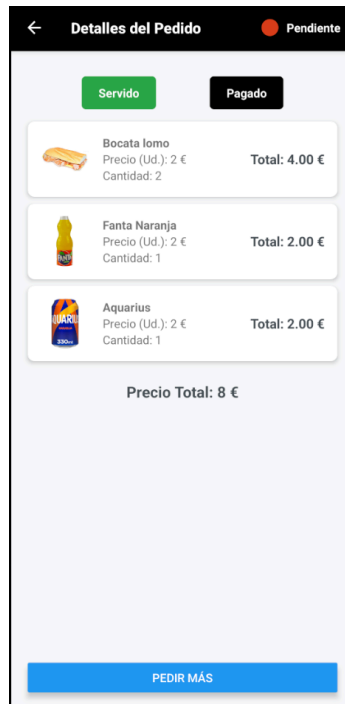


Fig. 10 Detalles del pedido

Para añadir los menús que va a tener nuestro restaurante, deberemos de ir al apartado “gestión de menús” (Fig. 4) en ajustes. En esta pantalla aparecen todos los menús que tenemos en nuestro restaurante (Fig. 11), si es la primera vez deberemos de crear nuestro primer menú. Para ello, habrá que clicar en el botón azul de la esquina superior derecha.

**IMPORTANTE:** Solo pueden añadir y editar menús los usuarios con rol admin, es decir el propietario.

Se abrirá una nueva pantalla en la que habrá que rellenar un formulario con los datos del menú que vamos a crear(Fig.12).

Una vez rellenado le damos click al botón de + abajo a la derecha, lo que creará el menú y volveremos a la pantalla de menús.

Si le damos a un menú ya creado iremos a la pantalla de edición (Fig. 13), en la que podremos cambiar los datos y la foto de ese menú, con las opciones o bien de guardarlo o bien de eliminarlo.

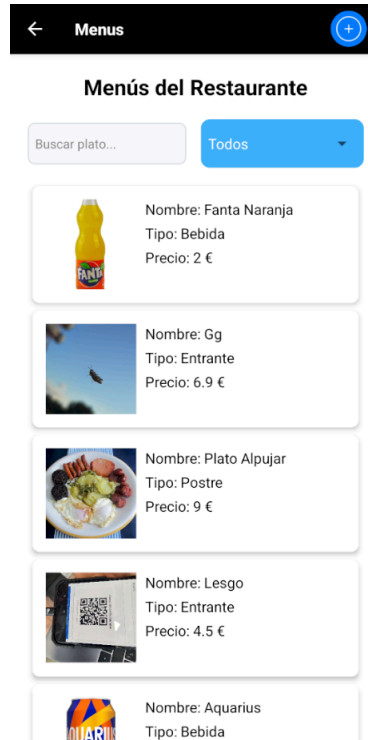


Fig. 11 Configuración de menú



Fig. 12 Nuevo menú



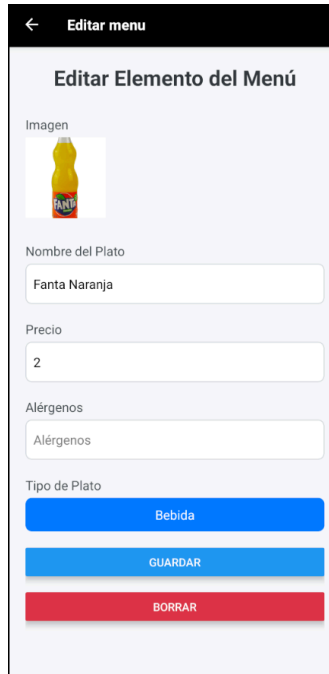


Fig. 13 Editar menú

Si se quiere cerrar sesión basta con tocar el icono del perfil(Fig. 14) y cerrar sesión, en el apartado de ver perfil se pueden editar los datos del usuario(Fig.15).

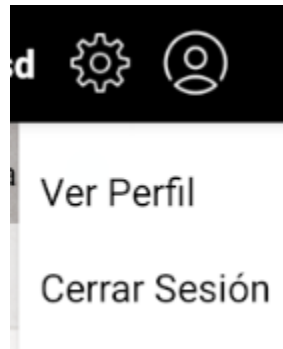


Fig. 14 Opciones de perfil

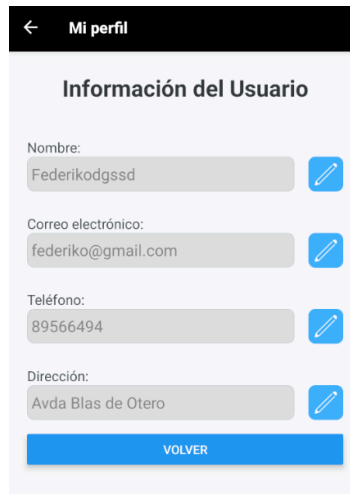


Fig. 15 Información de usuario

## 9 PREGUNTAS FRECUENTES

### ¿Qué hacer si no se puede iniciar sesión?

Se recomienda comprobar que la conexión a internet está activa. Verificar que el correo electrónico y la contraseña introducidos son correctos. Si el problema persiste, contactar con el administrador para restablecer la cuenta a través del panel de Firebase.

### ¿Qué ocurre si se pierde la conexión a internet?

EasyOrder está diseñado para mantener datos locales temporales hasta que se recupere la conexión. Los pedidos serán sincronizados automáticamente con Firestore en cuanto se restablezca el acceso a internet. Sin embargo, se recomienda operar en entornos con conectividad estable para garantizar un funcionamiento fluido.

### ¿Cómo se gestiona el menú del restaurante?

Los productos pueden gestionarse desde una interfaz administrativa o mediante acceso autorizado a Firestore. Los camareros solo tienen permiso de lectura sobre el menú, mientras que los administradores pueden agregar, editar o eliminar productos.

### ¿Puedo usar EasyOrder en varios dispositivos al mismo tiempo?

Sí, Firebase permite sesiones simultáneas, aunque por motivos de seguridad se recomienda una sesión activa por camarero. La información se actualiza en tiempo real en todos los dispositivos conectados.

### ¿Dónde se almacenan las imágenes del menú?

Las imágenes asociadas a los productos del menú se almacenan en Firebase Storage. Cada imagen está vinculada a su producto correspondiente mediante su URL, la cual se guarda en la base de datos.

### ¿La app funciona en iOS y Android?

Sí. Al estar desarrollada en React Native, EasyOrder es compatible con ambos sistemas operativos y mantiene una experiencia de usuario unificada en distintas plataformas.

## 10 CONCLUSIONES

El desarrollo de EasyOrder ha permitido validar la viabilidad y efectividad de una solución tecnológica enfocada en la digitalización de la gestión de pedidos en el sector de la restauración.

El uso de **React Native** ha sido clave para acelerar el proceso de desarrollo y facilitar el mantenimiento de una única base de código para ambas plataformas móviles (Android e iOS), sin comprometer el rendimiento ni la experiencia de usuario.

La integración de **Firebase** como backend ha simplificado la arquitectura general del sistema, ofreciendo servicios robustos de autenticación, base de datos en tiempo real, almacenamiento de archivos y notificaciones push. Esto ha permitido centrarse en la lógica de negocio y en la experiencia del usuario sin tener que gestionar servidores o infraestructura compleja.

Las pruebas funcionales y de usuario han confirmado que la aplicación responde adecuadamente a las necesidades reales del entorno hostelero, mejorando la eficiencia en la toma de pedidos, reduciendo errores humanos y optimizando la comunicación entre el personal de sala y cocina.

Además, el diseño modular y escalable del sistema permite futuras extensiones, como incluir funciones de cobro digital, generación de reportes o integración con sistemas de gestión de restaurantes.

En resumen, EasyOrder representa una solución moderna, funcional y adaptable para entornos con alta rotación de pedidos, ofreciendo beneficios tangibles tanto para los empleados como para los clientes.

## 11 REFERENCIAS (BIBLIOGRAFÍA Y WEB)

### **React Native Documentation**

<https://reactnative.dev/docs/getting-started>

Documentación oficial para el desarrollo de aplicaciones móviles multiplataforma.

### **Firebase Documentation**

<https://firebase.google.com/docs>

Guía oficial de Firebase sobre autenticación, Firestore, almacenamiento y notificaciones.

### **Expo Documentation**

<https://docs.expo.dev/>

Información para crear y testear aplicaciones React Native de forma rápida sin configuración nativa compleja.

### **React Navigation**

<https://reactnavigation.org/docs/getting-started>

Librería principal usada para la navegación entre pantallas en la app.

### **JavaScript / TypeScript Language Reference**

<https://developer.mozilla.org/en-US/docs/Web/JavaScript>

Recursos y documentación técnica sobre el lenguaje usado en la aplicación.

### **Stack Overflow**

<https://stackoverflow.com/>

Plataforma de preguntas y respuestas utilizada para resolver errores técnicos y buenas prácticas.