MANUAL TECNICO

Tec. Ciencias Computacionales

Universidad Don Bosco Campus Soyapango

INTRODUCCION

La finalidad de todo manual técnico es la de proporcionar al lector las pautas de configuración y la lógica con la que se ha desarrollado una aplicación, la cual se sabe que es propia de cada programador; por lo que se considera necesario ser documentada. Aclarando que este manual no pretende ser un curso de aprendizaje de cada una de las herramientas empleadas para el desarrollo del sitio, sino documentar su aplicación en el desarrollo del sitio. Para un mayor detalle acerca de cada una de las herramientas utilizadas, y su forma de operación y aplicación, se recomienda consultar los manuales respectivos de cada una de ellas.

A lo largo de este manual, encontrarás información esencial para comprender y desarrollar desde su diseño arquitectónico hasta su implementación práctica. Cada sección está diseñada para brindar una comprensión completa de cómo funciona el sistema y cómo puede ser configurado y mantenido

Objetivos

Se ha creado dicho documento con el propósito de mostrar cómo fue diseñado el sistema, y al mismo tiempo dar referencias de como interactuar con el programa para que sea actualizado o al mismo tiempo se le dé un mantenimiento adecuado en caso de un fallo.

A grandes rasgos se diseñó con el mero propósito de guiar al programador que este al frente de dicho sistema como se hizo, su proceso de instalación, código fuente, etc.

La aplicación de farmacia permite a los usuarios registrados acceder a una plataforma donde pueden:

- **Ver el catálogo** de productos (medicamentos).
- **Agregar productos al carrito** de compras.
- **Realizar pedidos** y ver la orden.
- **Consultar el historial** de compras previas.

Los usuarios no registrados no podrán realizar compras, pero tendrán acceso a la pantalla de inicio y podrán registrarse.

Alcances

Este documento está Hecho para las personas interesadas a conocer de nuestro proyecto y como se maneja nuestro sistema de desarrollo y herramientas utilizadas se debe tener un poco de experiencia en programación para comprender mejor nuestro sistema

Requerimientos Técnicos

A. Software

- Gestor de bases de datos: (Firebase) para la administración de los registros almacenados
- Un codificador de desarrollo: (Android Studio) para la actualización y modificación del código.
- Cualquier sistema operativo funcional

B. Hardware

- Dispositivo requerido:
  - Laptop
  - Pc
  - Teléfono

C. Requerimiento Mínimos de Hardware.

- Procesado de antigua o nueva generación (no hay requerimientos)
- Memoria RAM (mínimo): 8gb
- Disco Duro: 500gb

Arquitectura Del Proyecto

El proyecto está basado en el patrón **MVC (Modelo-Vista-Controlador)**:

- **Modelo (Model)**: Gestiona la lógica de la aplicación relacionada con los datos, como los modelos de los productos (Product), usuarios (User), y órdenes (Order).
- **Vista (View)**: Se encarga de la interfaz de usuario en archivos XML y el uso de RecyclerViews para mostrar el listado de productos y el historial de órdenes.

- **Controlador (Controller)**: Las actividades y fragmentos que conectan la vista con el modelo. Por ejemplo, MainActivity.kt es responsable de la navegación y de las acciones del usuario.

Componentes Principales A. Cliente (Frontend):

La app proporciona una interfaz intuitiva para que los usuarios busquen comida y realicen compras y administren sus pedidos de una manera efectiva Servidor (Backend):

Desarrolla y gestiona la lógica del negocio, procesa las solicitudes del cliente y se comunica con la base de datos.

Base de datos: se utilizó como base de datos para almacenar cada dato de información que pueda llegarse a ingresar.

Tecnologías Utilizadas

GitHub

GitHub es una herramienta esencial para los desarrolladores de software. Es utilizada por millones de personas en todo el mundo para trabajar en una amplia gama de proyectos, desde pequeños scripts hasta grandes aplicaciones empresariales. Donde ofrece una mayor administración de cada proyecto alojado en la plataforma

¿Por qué se escogió?

En parte se escogió por lo antes mencionado de que ofrece una mayor administración y orden a la hora de estar desarrollándolo y puedes tener el control de cada desarrollador que está participando en el proyecto y también por que ofrece una estabilidad de alojamiento grande y de una mayor capacidad y ayuda a gestionar problemas que pueden llegar a presentarse a la hora del desarrollo.

FireBase

FireBase es una solución poderosa y confiable para administrar bases de datos. Su naturaleza de código abierto, compatibilidad multiplataforma y amplio apoyo de la comunidad lo convierten en una opción popular para diversas aplicaciones y proyectos basados en datos.

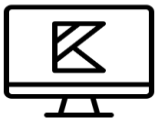
¿Por qué se escogió?

Como base de datos relacional, se utilizó Firebase para almacenar la información de cada usuario ingresado en la aplicación y poder tener una mejor gestión y tráfico de datos.

Android Studio

es un editor de código fuente potente y versátil que satisface las necesidades de los desarrolladores modernos. Su naturaleza gratuita y de código abierto, combinada con sus amplias funciones y personalización, lo convierten en una opción popular para programadores de todos los niveles.

Kotlin



KOTLIN: Sera el lenguaje principal para el desarrollo de la aplicación móvil. Este lenguaje, que es compatible con java y especialmente para el desarrollo en Android,

ofrece una sintaxis concisa y expresiva, lo que facilita la escritura de código limpio y seguro.

Paquetes y Clases

Paquete sv.edu.udb.comida_mex:

1. **Activities**:
    - MainActivity.kt: Actividad principal que gestiona la navegación.
    - LoginActivity.kt: Actividad de inicio de sesión y registro.
    - SelectFoodActivity.kt: Muestra la lista de productos disponibles.
    - OrderActivity.kt: Muestra los detalles de la orden actual.
    - OrderHistoryActivity.kt: Muestra el historial de compras del usuario.
2. **Adapters**:

- SelectFoodAdapter.kt: Adaptador para mostrar la lista de productos.
- OrderHistoryAdapter.kt: Adaptador para mostrar el historial de compras.

3. **Utilidades**:

- SessionManager.kt: Clase para gestionar las sesiones del usuario usando SharedPreferences.

Componentes Clave

Firebase

Firebase es el backend utilizado para gestionar la autenticación de usuarios y almacenar datos de productos y pedidos.

1. **Autenticación**:
    - Los usuarios pueden **registrarse** e **iniciar sesión** con correo electrónico y contraseña.
    - **FirebaseAuth** se utiliza para gestionar el registro y la autenticación.

Firestore:

- **Colección de productos (products)**: Aquí se almacenan los productos disponibles con sus detalles (nombre, precio, img, descripcion).

2. **Diseño de la Base de Datos en Firestore**

- **Colección products**:
    - Nombre: String
    - Precio: Float
    - Img: String (URL a Firebase Storage)
    - Descripcion: String

3. Pantallas Principales

Pantalla de Inicio de Sesión/Registro

- **Descripción**: Permite a los usuarios registrarse e iniciar sesión.
- **Firebase**: Utiliza FirebaseAuth para autenticar usuarios.

4. Pantalla de Productos

- **Descripción**: Muestra la lista de productos disponibles para la compra. Los usuarios pueden agregar productos a la orden.
- **Firebase**: Recupera los productos de la colección products en Firestore.
- **Gestión**: La orden se gestiona temporalmente con SharedPreferences, y al confirmar la compra,

5. Instrucciones para Desarrolladores

Configuración del Entorno

1. Clonar el repositorio del proyecto desde Git.
2. Abrir el proyecto en Android Studio.
3. Configurar Firebase:
    - Conectar el proyecto a Firebase desde el menú de herramientas de Android Studio.

- Añadir el archivo google-services.json proporcionado por Firebase.
- Asegurarse de habilitar **Firestore** y **Firebase Auth** en la consola de Firebase.

6. Ejecución del Proyecto

1. Asegurarse de tener Android Studio configurado correctamente.
2. Sincronizar el proyecto con Firebase.
3. Ejecutar la aplicación en un emulador o dispositivo físico

7. Conclusión

Esta aplicación de farmacia es un ejemplo de cómo combinar tecnologías modernas como Firebase con Kotlin en Android para crear una aplicación móvil robusta y eficiente. La estructura permite una fácil expansión, agregando nuevas funcionalidades o integraciones