Desarrollo de una Aplicación Móvil para una Tienda Erótica Usando Flutter: Erotic Store

Corporación Universitaria Remington.
Facultad de ingeniería.
Ingeniería en Sistemas.

Alexis Castilla Medina.
Jonatan Stick Campos Núñez.
Seminario Programación Mobile.
2025.

Contenido

Indice De Imagenes

Resumen

El presente proyecto de grado consiste en el diseño y desarrollo de una aplicación móvil (aún en proceso), enfocada en la venta de productos eróticos, está siendo construida con la aplicación Flutter como framework multiplataforma, lo que permite su implementación tanto en sistemas Android y iOS a partir de una sola base de código.

El seminario tuvo como propósito introducir las principales características de Flutter, como lo es su arquitectura basada en widgets y su lenguaje de programación Dart. El objetivo es crear una aplicación que facilite comprar productos íntimos, garantizando seguridad en los pagos, anonimato en el manejo de datos y una interfaz amigable para los usuarios.

Se quiere llegar a un producto final el cual debe incluir un catálogo interactivo, interacción con pasarelas de pagos seguras y un sistema de recomendaciones personalizadas que permita mejorar la experiencia de compra digital en el sector adulto, impulsando el comercio electrónico.

Palabras Clave

(Flutter, aplicación móvil, comercio electrónico, seguridad, tienda erótica.)

Glosario

Dart

Es un lenguaje de programación desarrollado por google, el cual es diseñado para hacer o crear aplicaciones de forma rápida, escalables y fáciles de mantener ya facilita mucho su desarrollo orientado a objetos, este lenguaje es muy usado en Flutter el cual permite realizar apps móviles, de escritorio y web con un mismo código.

El lenguaje combina lo mejor de algunos leguajes ya conocidos como lo son Java, JavaScript, C#, o busca hacerlo.

El lenguaje está orientado a objetos, podrían decir que todo en Dart es un objeto, como lo son los números y las funciones.

Flutter

Es el framework creado por google en 2017, este sirve para crear aplicaciones multiplataforma como lo son iOS, Windows, macOS, Android, web y Linux. Esta se caracteriza por permitir crear apps con alto rendimiento las cuales se dejan personalizar y pueden volverse muy atractivas para el destinatario final que le puede llamar usuario.

Este framework permite ver los cambios realizados en el código casi al instante, sin necesidad de reiniciar la aplicación, lo cual es muy ideal para in desarrollo ágil y rápido. De igual forma este usa el lenguaje de programación Dart, esto permite compilar rápidamente y optimizar el rendimiento de la misma.

Se puede decir que todo Flutter es un widget, sus botones, imágenes, textos y layouts.q

Android Studio

Es un entorno de desarrollo integrado (IDE) para crear apps Android, este fue lanzado por Google en 2013. Está basado en IntelliJ IDEA, un IDE muy potente para java, al cual se le agregaron herramientas y configuración para desarrollar aplicaciones móviles en Android.

Su propósito podría ser proporcionar un entorno todo en uno, que facilite escribir código de manera eficiente, diseñar interfaz gráfica con vista previa, depurar y analizar el rendimiento de las aplicaciones, de igual forma probar estas en distintos dispositivos mediante emuladores.

Visual Studio Code

Editor de código, el cual es multiplataforma, es decir funciona en Windows, macOS y Linux. Fue desarrollado por Microsoft en 2015, está diseñado para ser rápido, flexible y

extensible, por esto se adapta a muchos lenguajes y entornos de programación, como lo son Python, C ++, Java, PHP, Dart y Flutter, lo hace mediante extensiones.

También permite abrir una terminal dentro de este mismo, lo que es ideal para ejecutar comandos de npm, pip, git, flutter.

Algo muy importante que es gratuito y compatible con casi todos los lenguajes.

SDK.

Software Development Kit que en español se traduce a Kit de Desarrollo de Software, lo cual se refiere a un conjunto de herramientas, librerías, utilidades y documentación que se proporciona para que los desarrolladores creen apps sobre esta.

Esto normalmente contiene APIs, Compiladores, simuladores y librerías, el SDK se podría considerar importante ya que garantiza compatibilidad con las plataformas, aceleran el desarrollo, es decir, no hay que programar desde cero. También ofrece documentación y reducción de errores.

Widget

Es un componente visual o funcional que describe una parte de la interfaz (cómo debe verse y comportarse).

Interfaz de Usuario

Es todo lo que ves y usas en una aplicación o programa para comunicarte con la computadora o el dispositivo: botones, menús, ventanas, iconos, pantallas, colores, tipografías, etc.

Login

Un login (del inglés *log in*) es el proceso de identificación de un usuario en un sistema, aplicación o página web, normalmente mediante un nombre de usuario (o correo electrónico) y una contraseña.

Objetivo General

Crear una aplicación que facilite la compra de productos íntimos, garantizando seguridad en los pagos, anonimato en el manejo de datos y una interfaz amigable para los usuarios.

Objetivos Especificos

Diseñar e implementar la interfaz de usuario de la aplicación móvil utilizando Flutter, garantizando una experiencia intuitiva, atractiva y adaptable a diferentes dispositivos móviles.

Desarrollar un sistema de autenticación y login seguro, que permita el registro e inicio de sesión de los usuarios mediante correo electrónico y/o redes sociales.

Integrar un catálogo de productos eróticos, organizado por categorías, con imágenes, descripciones y precios dinámicos.

Implementar un carrito de compras interactivo, que permita añadir, modificar y eliminar productos antes de realizar la compra.

Desarrollar un panel de gestión de usuarios y pedidos, donde los administradores puedan administrar el inventario, controlar ventas y actualizar información de productos.

Garantizar la seguridad de la información sensible, aplicando buenas prácticas de encriptación y protección de datos personales.

Realizar pruebas de usabilidad y funcionalidad, con el fin de detectar y corregir errores, asegurando que la aplicación cumpla con los requerimientos planteados.

¿Qué es una Aplicación Movil?

Es una aplicación que descargas e instalas en tu celular o Tablet para realizar tareas específicas: comunicarte, comprar, aprender, entretenerte, etc.

Características:

Se ejecuta en sistemas operativos móviles como Android o iOS**.**

Puede funcionar con conexión a internet (ej. redes sociales) o sin conexión (ej. calculadora, bloc de notas).

Está optimizada para pantallas táctiles y diferentes tamaños de dispositivos.

Suele descargarse desde tiendas oficiales como Google Play Store o Apple App Store**.**

Usabilidad en Aplicaciones Moviles

La usabilidad en aplicaciones móviles es la clave para que una app sea eficiente, fácil de usar y agradable. Una app con buena usabilidad garantiza que los usuarios logren sus objetivos rápidamente, disfruten la experiencia y quieran seguir usándola.

Importancia de las Aplicaciones Móviles

En la actualidad, las aplicaciones móviles constituyen una de las herramientas tecnológicas de mayor impacto en la sociedad, ya que permiten a los usuarios acceder de manera inmediata a una amplia gama de servicios, desde la comunicación y el entretenimiento hasta la educación y el comercio electrónico. Su crecimiento responde a la necesidad de contar con soluciones portátiles, rápidas y accesibles, adaptadas a los estilos de vida modernos y a la expansión global de los teléfonos inteligentes.

Las apps no solo han transformado la forma en que las personas interactúan con la información, sino que también han impulsado el desarrollo de nuevos modelos de negocio en sectores como el transporte, la banca, la salud y la educación. De esta manera, las aplicaciones móviles se convierten en un canal estratégico para las empresas, permitiéndoles ofrecer experiencias personalizadas, fortalecer la relación con los clientes y ampliar su alcance en un mercado altamente competitivo.

Desde una perspectiva tecnológica, estas aplicaciones fomentan la innovación mediante la integración de recursos como la inteligencia artificial, la realidad aumentada y el Internet de las Cosas (IoT), lo que contribuye al diseño de entornos digitales cada vez más interactivos y adaptados a las necesidades de los usuarios. En conclusión, las aplicaciones móviles son hoy un pilar fundamental del ecosistema digital, pues favorecen la

productividad, la inclusión social y el crecimiento económico, a la vez que transforman los hábitos de consumo y comunicación en la sociedad contemporánea.

Tipos de Aplicaciones Móviles

Aplicaciones Nativas

Las aplicaciones nativas son aquellas diseñadas para funcionar exclusivamente en un sistema operativo específico, como Android o iOS**.** Utilizan los lenguajes de programación oficiales de cada plataforma (Java o Kotlin en Android, Swift u Objective-C en iOS) y aprovechan al máximo los recursos del dispositivo, como la cámara, el GPS o los sensores biométricos. Su principal ventaja es el alto rendimiento y la integración profunda con el hardware, aunque su desarrollo implica mayores costos y tiempos de producción, ya que se requiere un proyecto distinto para cada sistema operativo.

Aplicaciones Web

Estas aplicaciones web son en realidad sitios web optimizados para móviles que se ejecutan desde un navegador, sin necesidad de instalación en el dispositivo. Se desarrollan con tecnologías como HTML, CSS y JavaScript, y resultan una opción más económica, pues funcionan en cualquier sistema operativo con acceso a internet. No obstante, presentan limitaciones en cuanto al acceso a las funciones internas del dispositivo y suelen ofrecer un rendimiento inferior en comparación con las aplicaciones nativas.

Aplicaciones Hibridas

Las aplicaciones híbridas combinan características de las nativas y de la web. Se desarrollan con frameworks como Flutter, React Native o Ionic, lo que permite escribir un solo código que puede ejecutarse tanto en Android como en iOS. Esto reduce costos y tiempo de desarrollo, siendo una alternativa muy popular en la actualidad. Sin embargo, aunque ofrecen un buen rendimiento y acceso moderado al hardware, en casos de aplicaciones que demandan un uso intensivo de recursos gráficos o de procesamiento, suelen ser menos eficientes que las nativas.

A conclusión podría decirse que la clasificación de estas permite comprender las distintas estrategias de desarrollo en el ámbito móvil. Lo que cada una presenta ventajas y limitaciones, lo que lleva a que la elección dependa de factores como lo son el presupuesto, el usuario objetivo, el tiempo disponible y el nivel de interacción requerido.

Flutter

Imagen 1 Logo Flutter

Flutter es un framework de desarrollo de interfaces de usuario de código abierto, creado por Google, que permite construir aplicaciones nativas y multiplataforma a partir de una única base de código. Utiliza el lenguaje de programación Dart y se basa en una arquitectura de widgets, lo que facilita la creación de interfaces gráficas altamente personalizables, interactivas y con un alto rendimiento. Su motor gráfico propio posibilita la compilación directa a código nativo, reduciendo la dependencia de componentes externos y garantizando una experiencia de usuario fluida en dispositivos móviles, web y de escritorio.

La elección de Flutter como framework para el desarrollo de la aplicación se fundamenta en su capacidad de crear soluciones multiplataforma a partir de una única base de código, lo que reduce costos y tiempos de desarrollo en comparación con enfoques nativos. Además, su arquitectura basada en widgets permite diseñar interfaces de usuario modernas, dinámicas y adaptables, lo cual resulta esencial para garantizar una experiencia atractiva e intuitiva en una tienda en línea.

Adicionalmente, al contar con un motor gráfico propio y un alto rendimiento, Flutter asegura una navegación fluida y una respuesta inmediata a las interacciones del usuario, aspectos cruciales para generar confianza y satisfacción en el proceso de compra. En este sentido, la implementación de Flutter contribuye no solo a la eficiencia técnica del proyecto, sino también al cumplimiento de los objetivos de usabilidad, accesibilidad y escalabilidad de la aplicación.

Marco Teorico

Instalación de Flutter

Ingresamos a la página oficial de Flutter en donde elegimos nuestro sistema operativo para comenzar a desarrollar.

Imagen 2 Selección del sistema operativo a desarrollar.

Seleccionamos el tipo de aplicación que deseamos desarrollar.

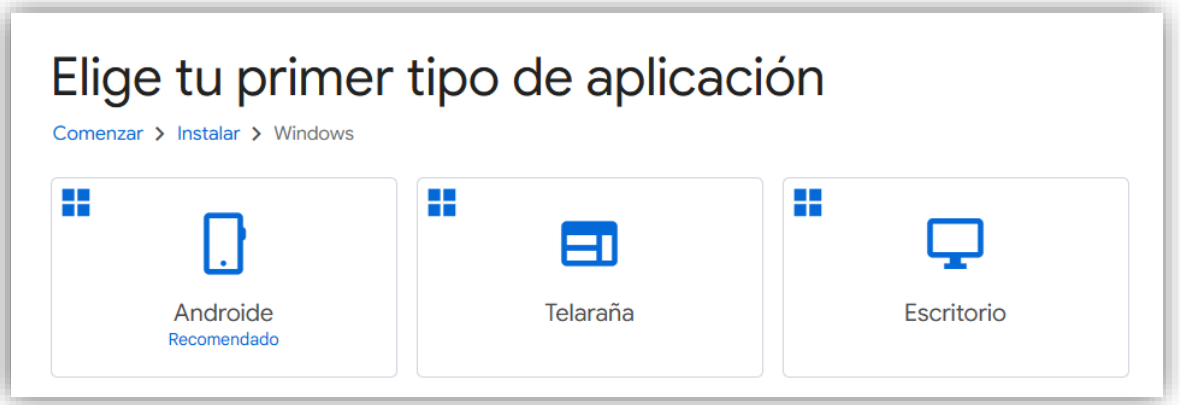


Imagen 3 Tipo de Aplicación a Desarrollar.

Procedemos a configurar el editor de texto a utilizar en el desarrollo de nuestra aplicación.

Nos recomiendan las más populares como lo son Visual Studio Code, Android Studio e IntelliJ IDEA.

Imagen 4 Configuración y elección de editor de texto.

Luego procedemos a instalar el SDK de Flutter.

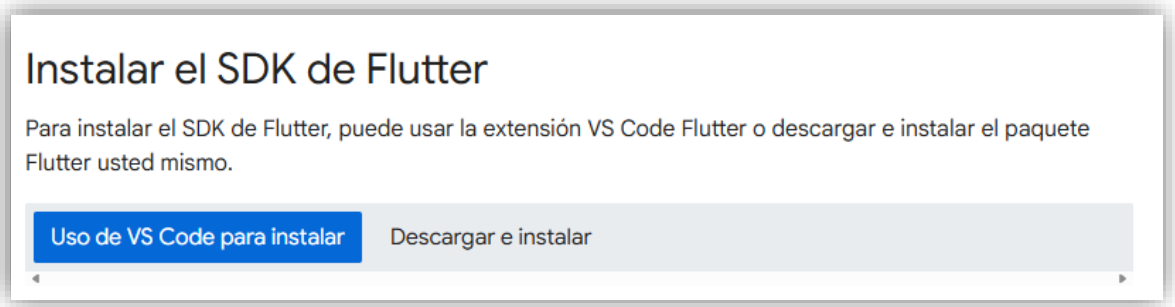
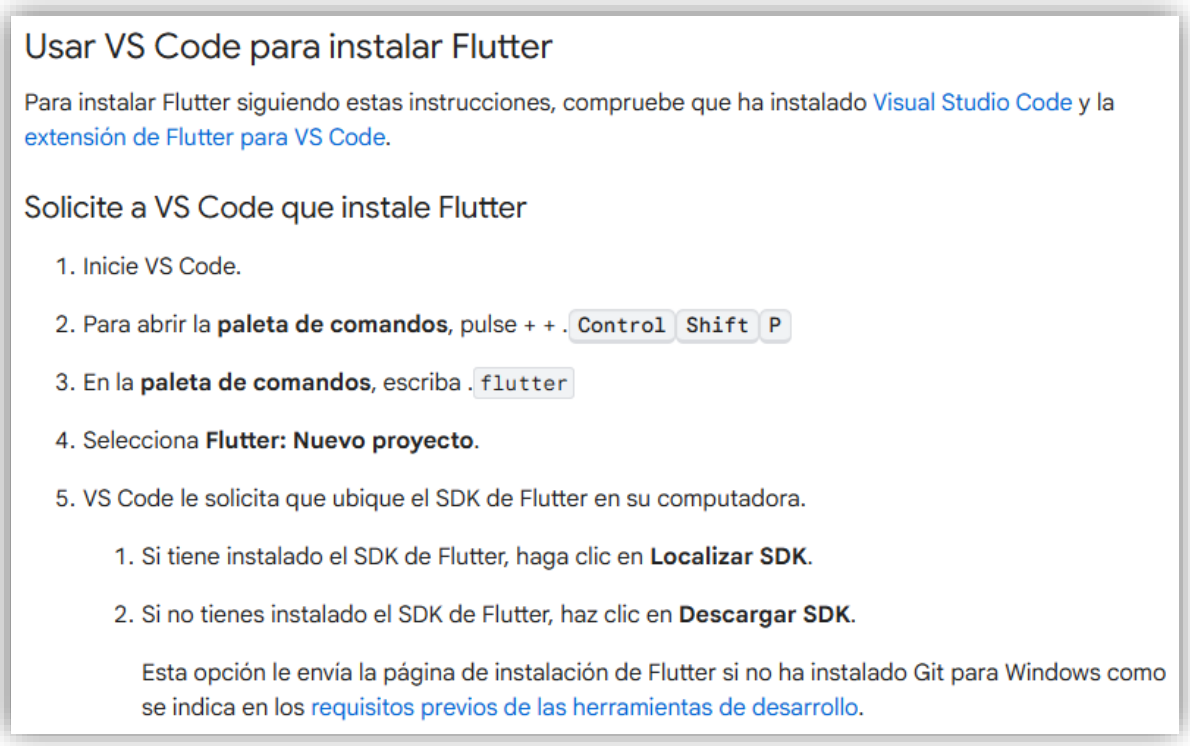


Imagen 5 Instalación del SDK Flutter.

En lo siguiente debemos seguir un paso a paso en cual comprobaremos la instalación de



Usar VS Code para instalar Flutter

Para instalar Flutter siguiendo estas instrucciones, compruebe que ha instalado Visual Studio Code y la extensión de Flutter para VS Code.

Solicite a VS Code que instale Flutter

1. Inicie VS Code.
2. Para abrir la **paleta de comandos**, pulse + + . `Control` `Shift` `P`
3. En la **paleta de comandos**, escriba . `flutter`
4. Selecciona **Flutter: Nuevo proyecto**.
5. VS Code le solicita que ubique el SDK de Flutter en su computadora.
    1. Si tiene instalado el SDK de Flutter, haga clic en **Localizar SDK**.
    2. Si no tienes instalado el SDK de Flutter, haz clic en **Descargar SDK**.

       Esta opción le envía la página de instalación de Flutter si no ha instalado Git para Windows como se indica en los requisitos previos de las herramientas de desarrollo.

nuestro editor de texto, en donde daremos inicio a un nuevo proyecto.

Imagen 6 Paso a Paso para iniciar nuestro Proyecto.

Agregamos el SDK a PAHT, esto nos genera una notificación, cerramos y reiniciamos VS Code.

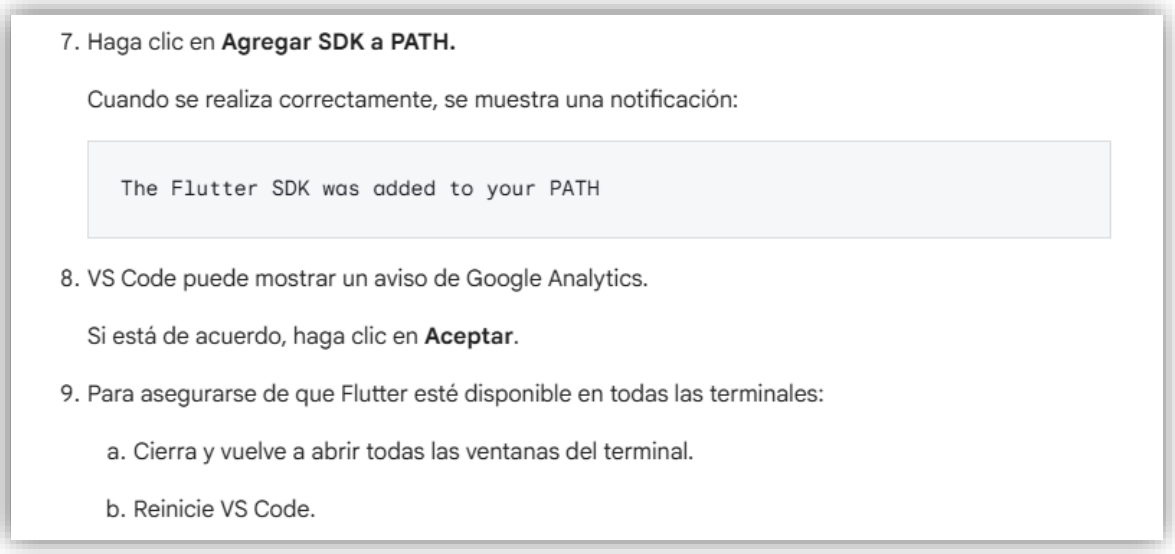


Imagen 7 Agregar SDK y reiniciar VS Code.

Lo siguiente es validar nuestra configuración en nuestra terminal con el comando **flutter doctor –v,** este comando nos debe notificar lo correcto o incorrecto de la instalación.

Imagen 8 Validación de la instalación de nuestro editor de texto.

El siguiente será nuestro entorno de trabajo en el cual desarrollaremos nuestro proyecto.

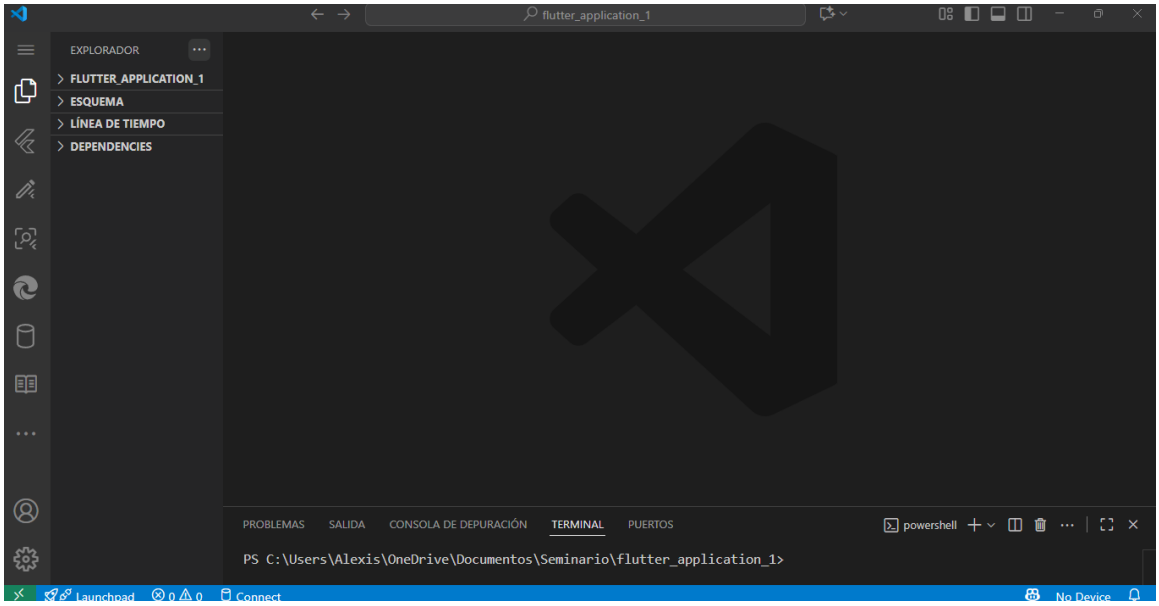


Imagen 9 Entorno de trabajo.

Proyecto

Erotic Store

Decidí realizar mi proyecto a partir de una idea surgida por una persona cercana la cual tiene un emprendimiento con estos productos, esta persona realiza algunos de estos como lo son las cremas y los tónicos.

La aplicación a crear se trata de una tienda en la cual toda persona mayor de 18 años pueda adquirir productos eróticos sin temor a que se revele su identidad, ni se expongan sus datos personales.

Se busca una mayor cercanía con el cliente, dando a conocer nuestros productos por un catálogo extenso lleno de variedad. Donde podrán elegir su mejor opción o lo que se acomode más a su personalidad u ocasión.

La aplicación Erotic Store aún en proceso de desarrollo va de la siguiente manera:

Una pantalla principal en la cual tenemos el nombre de la tienda, una bienvenida, el logo, un pequeño texto en el cual nos describe un poco y un botón que los lleva directamente al catálogo en el cual hay algunos de nuestros productos.

Imagen 10 Pantalla Principal

Catalogo en el cual podrán añadir le producto deseado y comprar la cantidad que quieras siempre y cuando se encuentre en stock, este te dará la opción de añadir varios productos a tu carro de compra.

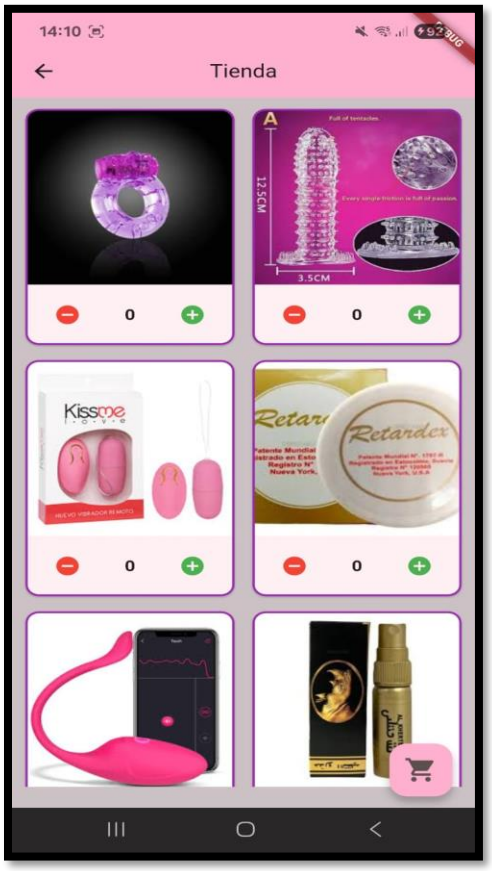


Imagen 11 Catalogo

A continuación, se comparten algunas líneas de código, las cuales representan botones, catalogo, pantalla principal, carrito de compra, etc.

En donde la primera imagen es una parte del main, esta es la función principal de entrada del programa, la cual llama al framework de flutter y pasa el widget raíz de la aplicación.

```dart
class MyApp extends StatelessWidget {
  const MyApp({super.key});

  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Flutter Demo',
      theme: ThemeData(

        colorScheme: ColorScheme.fromSeed(seedColor: const Color.fromARGB(255, 207, 15, 127)),
      ), // ThemeData
      home: const MyHomePage(title: 'Erotic Store'),
      routes: {
        '/inicio': (context) => const MyHomePage(title: 'Erotic Store',), // 👈 Ruta registrada
        '/tienda': (context) => const SecondPage(), // 👈 Ruta registrada
      },
```

Imagen 12 Función Main

En la siguiente imagen una parte de la pantalla principal, en donde aplicamos el color principal de la página y un fondo a esta misma.

```dart
class MyHomePage extends StatefulWidget {
  const MyHomePage({super.key, required this.title});

  final String title;

  @override
  State<MyHomePage> createState() => _MyHomePageState();
}

class _MyHomePageState extends State<MyHomePage> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        backgroundColor: Theme.of(context).colorScheme.inversePrimary,
        title: Text(widget.title),
        centerTitle: true,
      ), // AppBar
      body: Stack(
        children: [
          // Fondo
          Container(
           width: double.infinity,
           height: double.infinity,
          color: const Color.fromARGB(80, 94, 76, 88), // 👈 color de fondo
        ), // Container
```

Imagen 13 Código Pantalla Principal

En la siguiente imagen hacemos una lista en la cual serán ingresadas todas las imágenes de nuestro catálogo.

```dart
import 'package:flutter/material.dart';
class SecondPage extends StatefulWidget {
  const SecondPage({super.key});

  @override
  State<SecondPage> createState() => _SecondPageState();
}

class _SecondPageState extends State<SecondPage> {
  final List<String> images = [
    "assets/images/anillo.jpeg",
    "assets/images/condon.jpeg",
    "assets/images/dulce.jpeg",
    "assets/images/image2.jpeg",
    "assets/images/vibrador.jpeg",
    "assets/images/image1.jpeg",
  ];

  late List<int> quantities;

  @override
  void initState() {
    super.initState();
    quantities = List.filled(images.length, 0);
  }
```

Imagen 14 Código Lista de Imágenes

Y por último tenemos una imagen en la cual mostramos un widget con un PrimaryButtom, el cual tendrá uniformidad tanto en color como en forma.

```dart
class PrimaryButtom extends StatelessWidget {
  final String textoButtom;       // Texto que mostrará el botón
  final VoidCallback onPressed;   // Función que ejecutará al presionarlo

  const PrimaryButtom({
    super.key,
    required this.textoButtom,
    required this.onPressed, required TextStyle style,
  });

  @override
  Widget build(BuildContext context) {
    return ElevatedButton(
        style: ButtonStyle(
          backgroundColor: WidgetStateProperty.all<Color>(
          Theme.of(context).colorScheme.inversePrimary,
        ),
        ), // ButtonStyle

        onPressed: onPressed, // <- usa la función que llega por parámetro

        child: Text(textoButtom),
        // <- usa el texto que llega por parámetro
```

Imagen 15 PrimaryButtom

Conclusión

El desarrollo de la aplicación en Flutter para la tienda de productos eróticos ha permitido sentar una base sólida para una solución digital moderna, multiplataforma y adaptable a las necesidades del mercado. Hasta el momento se han alcanzado avances importantes en la estructura, diseño y experiencia de usuario, demostrando la eficacia de Flutter para crear interfaces atractivas, rápidas y seguras.

No obstante, la aplicación continúa en proceso de desarrollo, lo que abre la oportunidad de implementar nuevas funcionalidades, optimizar la integración con métodos de pago y reforzar aspectos relacionados con la seguridad y la privacidad del cliente. Este proyecto refleja no solo el potencial de Flutter como herramienta tecnológica, sino también el compromiso con la innovación en un sector que requiere confianza, accesibilidad y discreción en sus servicios.

Referencias

Vázquez Rodríguez, V. (2019). Desarrollo de aplicaciones móviles multiplataforma con Flutter.

Escandell Montiel, D. (2016). SDK.

Enriquez, J. G., & Casas, S. I. (2014). Usabilidad en aplicaciones móviles. *Informes Científicos Técnicos - UNPA*, *5*(2), 25–47. https://doi.org/10.22305/ict-unpa.v5i2.71

Bermúdez León, M. J. (2021). Importancia de las aplicaciones móviles.

GUADALUPE, G. M. M. (2015). Usos y tipos de aplicaciones móviles. *USOS Y TIPOS DE APLICACIONES MÓVILES*.