# Proyecto Flutter
# Tienda de Fortnite

Doriana Angélica Da Costa Magello
Programación multimedia y dispositivos móviles

Índice

# Introducción

En el universo de Fortnite, la personalización es clave, los jugadores no solo se sumergen en emocionantes batallas, sino que también tienen la oportunidad de expresar su estilo único a través de una amplia gama de cosméticos disponibles en la tienda del juego. Desde skins impresionantes hasta emocionantes emotes, mochilas y picos, la tienda de Fortnite ofrece un sinfín de opciones para que los jugadores definan su identidad en el campo de batalla virtual.

Sin embargo, detrás de la apariencia visualmente de estos cosméticos, se encuentra un sistema de rareza que determina su valor. Cada artículo cosmético en Fortnite se clasifica en una de varias rarezas, desde lo común hasta lo legendario, y el precio de cada uno refleja su exclusividad y deseabilidad.

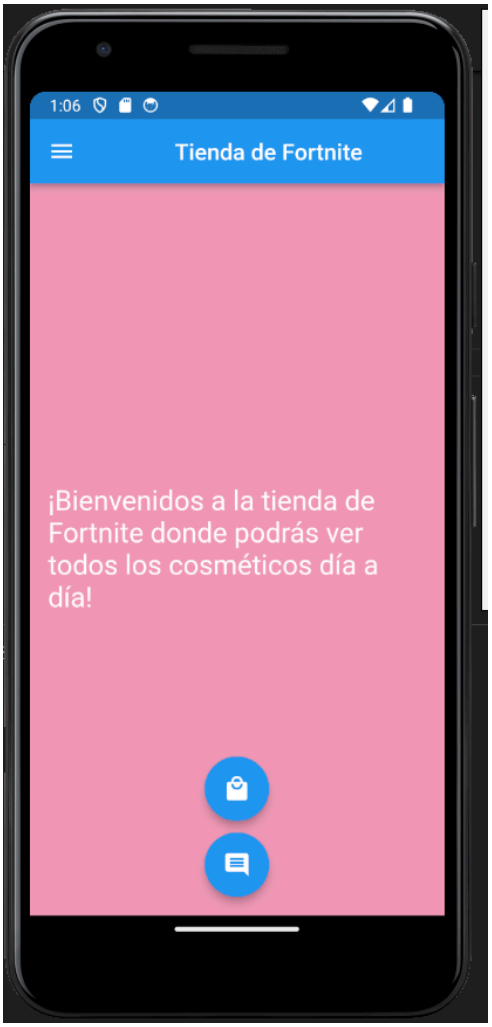
Ofreciendo a los usuarios una interfaz intuitiva y atractiva para explorar y adquirir estos emocionantes cosméticos, cada artículo se presenta junto con su precio correspondiente.

Para la persistencia se utilizó Firebase.

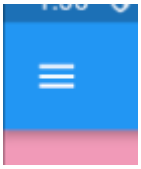
Interacción con la aplicación

**Pantalla Principal:**

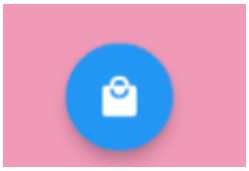
Nos da la bienvenida a la tienda de Fornite dando una pequeña explicación que podremos ver todos los cosméticos día a día, ya que la tienda cambia todos los días a la hora 1:00 am Española.




Botón de hamburguesa:  Nos permite acceder al drawer personalizado




Botón de la bolsita: Accedemos a la tienda



**Pantalla de la Tienda:**

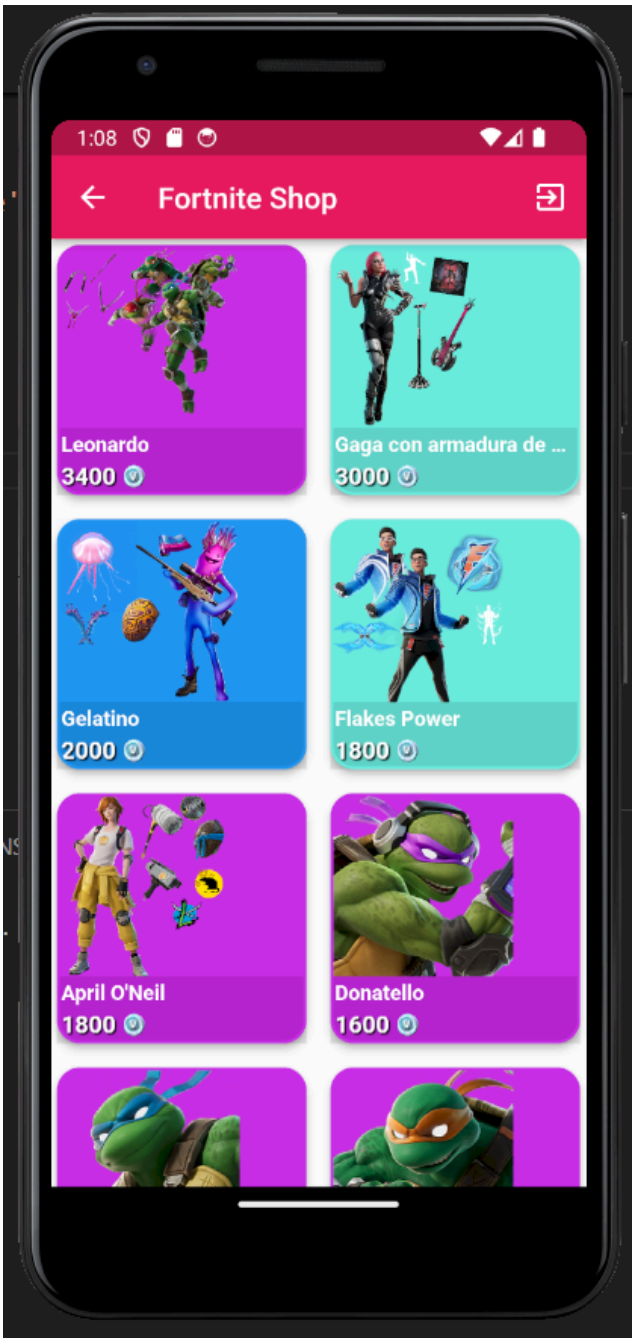
Se observa la tienda de Fortnite, con todos los cosméticos y su precio, ya sean packs, skins, mochilas, picos, camuflajes, emotes. Fortnite no solo saca al mercado cosméticos de la propia empresa sino que también hay de cantantes famosos o colaboraciones.

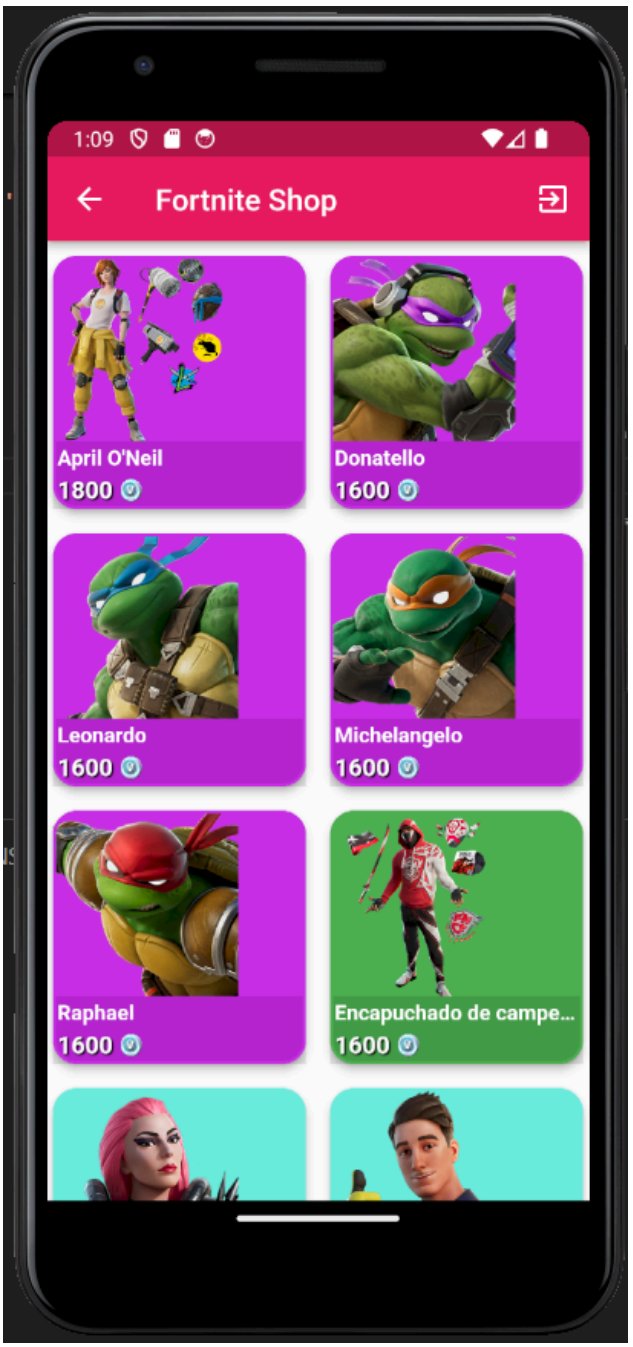
Origen del fondo de las skins: cada cosmético de Fortnite tiene una rareza con un color en especial, dependiendo de esa rareza es su precio.

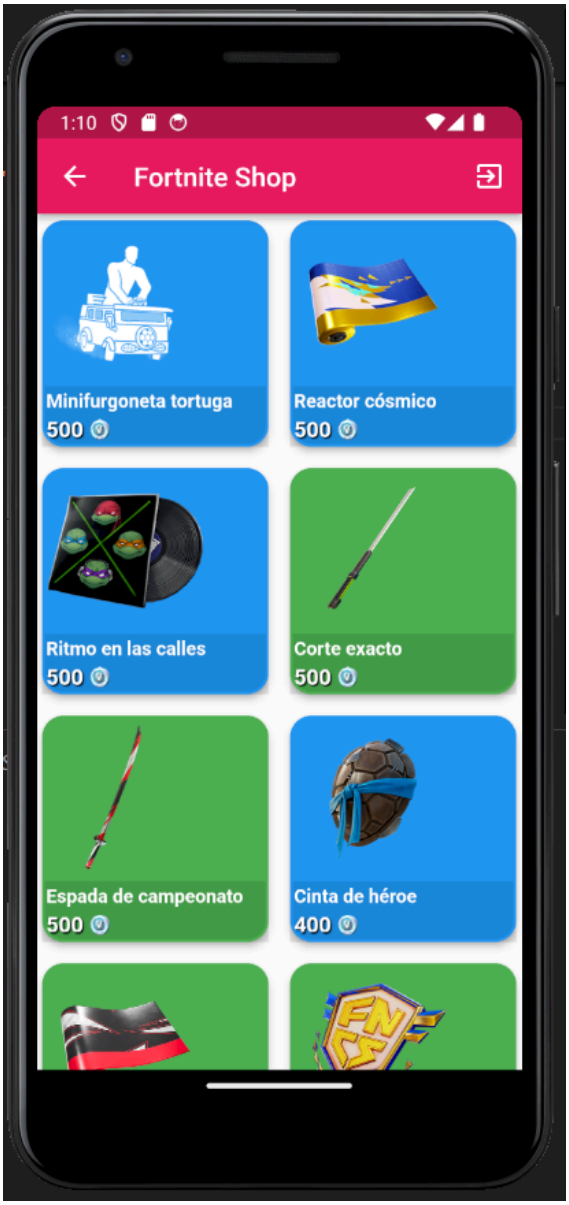
Las rarezas son:

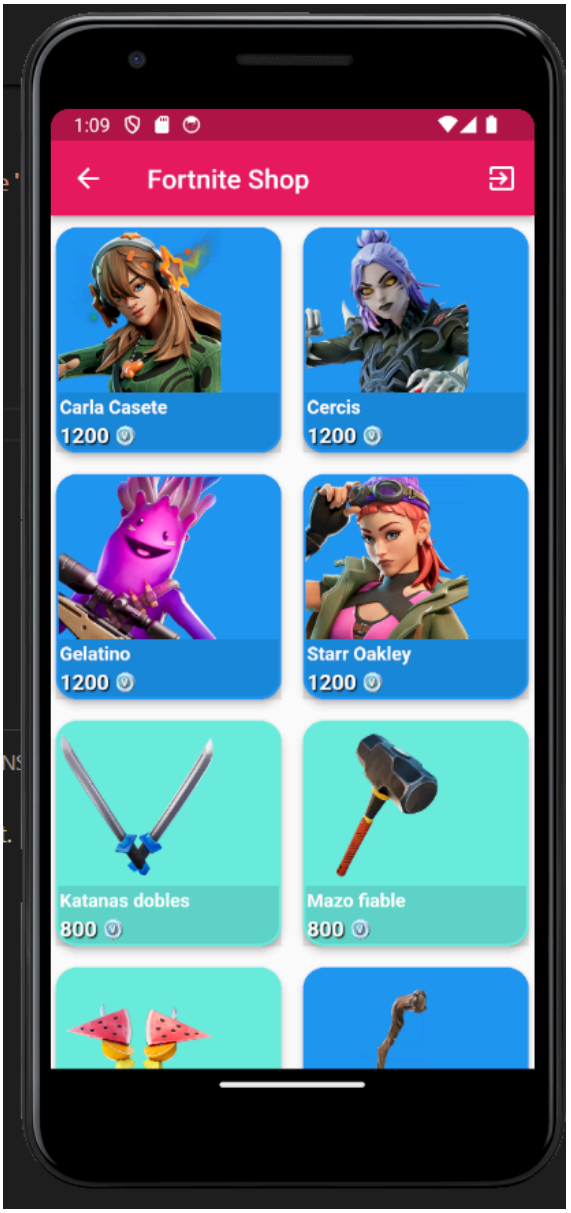
- Poco común: color gris y tiene un precio más económico de todos.
- Común: color verde, su precio es de 200 pavos para mochilas, camuflajes, emotes, picos y para skins sería de 800 pavos.
- Raro: color azul, emotes con valor a 500 pavos, skins con valor de 1200 pavos.
- Épico: color morado, emotes con valor de 800, skins con valor de 1500 pavos, picos con valor de 1200 pavos
- Legendario: color naranja, skins con valor de 2000 pavos
- Serie Ídolos: color turquesa, emotes de 500 pavos, picos de 800 pavos, skins con valor de 1500 pavos

Hay más rarezas como la de los videojuegos, la serie de sombra, la serie oscura, de colaboraciones con otras marcas como marvel, dc, star wars, nike, ariana grande, entre muchas otras.. Fortnite es un videojuego con un gran nivel de colaboraciones.

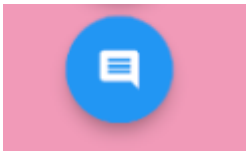

1:08
Fortnite Shop
Leonardo
3400
Gaga con armadura de ...
3000
Gelatino
2000
Flakes Power
1800
April O'Neil
1800
Donatello
1600


1:09
Fortnite Shop
April O'Neil
1800
Donatello
1600
Leonardo
1600
Michelangelo
1600
Raphael
1600
Encapuchado de campe...
1600

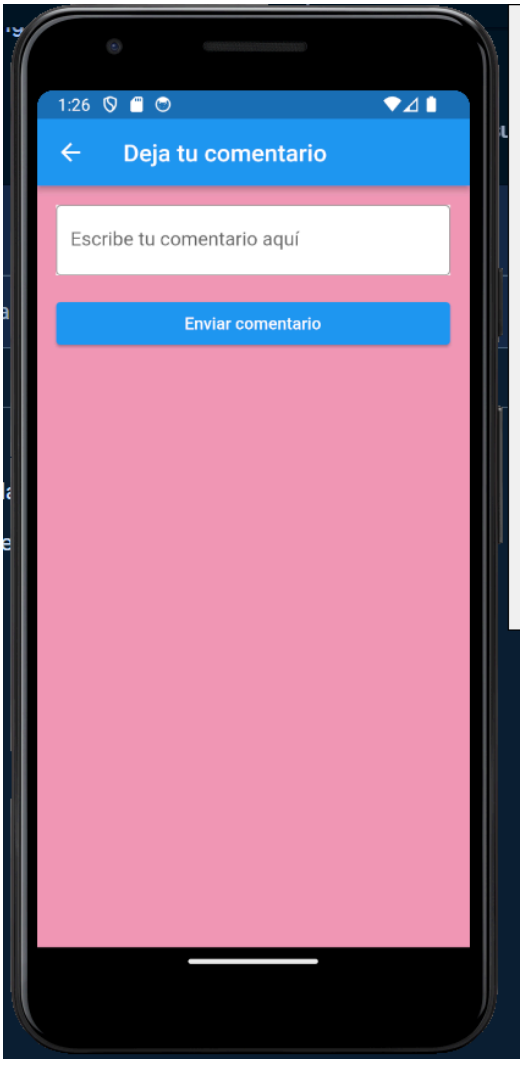

1:10
Fortnite Shop
Minifurgoneta tortuga
500
Reactor cósmico
500
Ritmo en las calles
500
Corte exacto
500
Espada de campeonato
500
Cinta de héroe
400

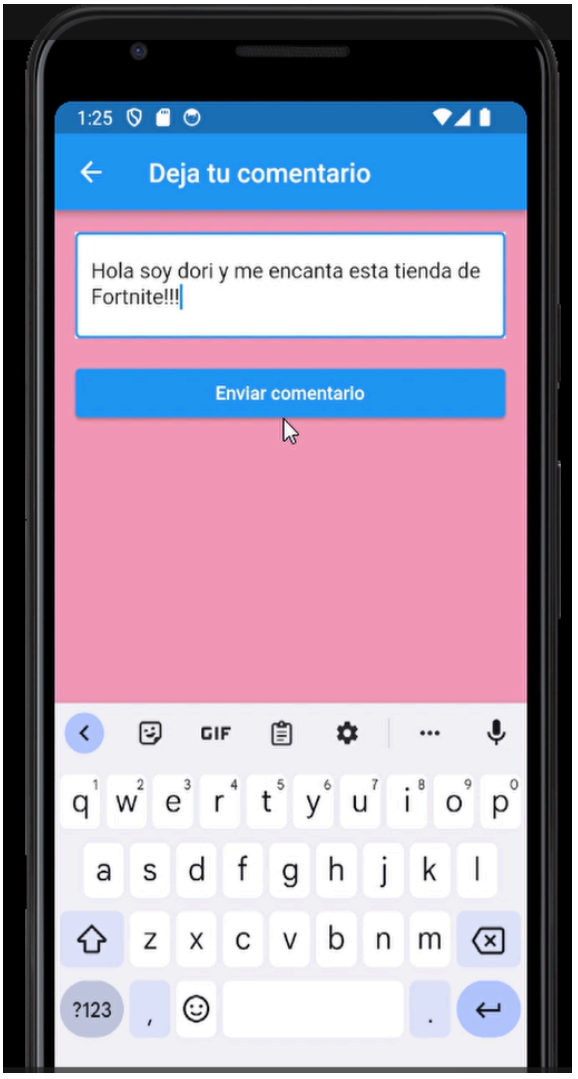

1:09
Fortnite Shop
Carla Casete
1200
Cercis
1200
Gelatino
1200
Starr Oakley
1200
Katanas dobles
800
Mazo fiable
800

Botón del mensajito: Accedemos a añadir un comentario.

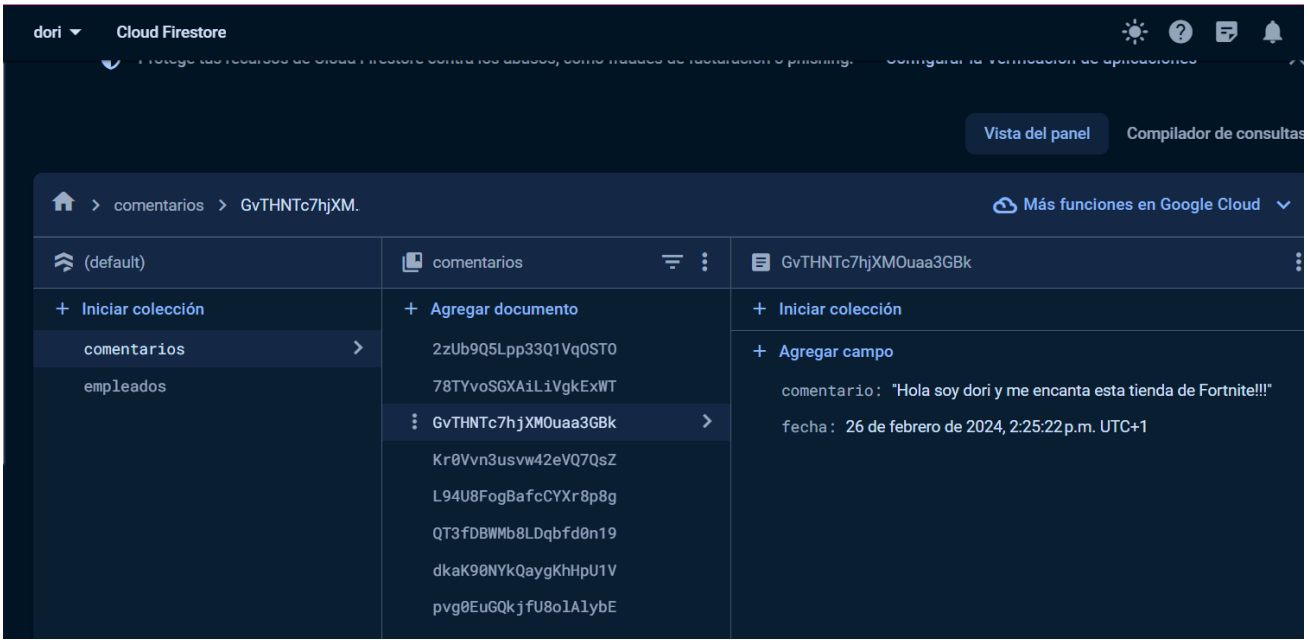


**Pantalla para añadir un comentario:**
Permite añadir un comentario y se guarda en la base de datos.

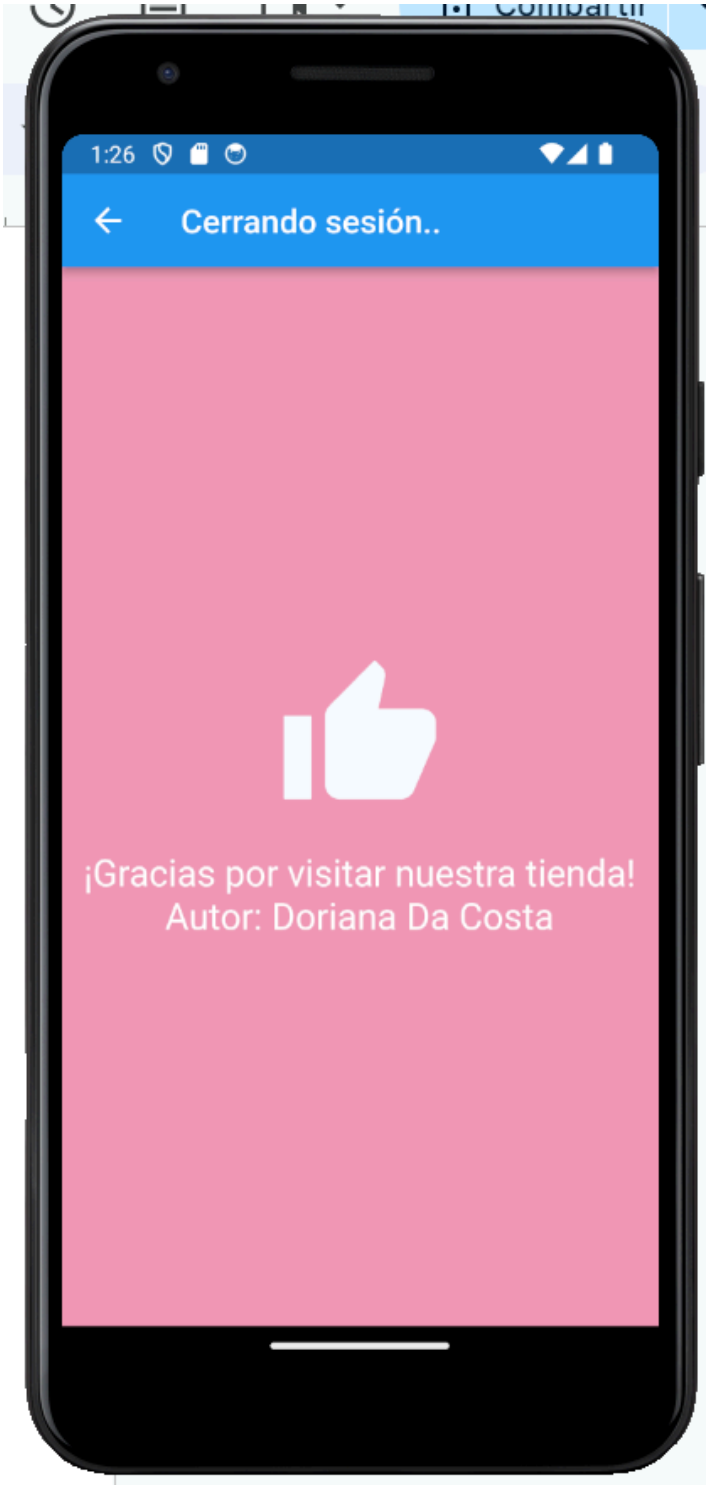



1:25
Deja tu comentario
Escribe tu comentario aquí
Enviar comentario
Comentario agregado correctamente
Aceptar


dori
Cloud Firestore
Vista del panel
Compilador de consultas
comentarios
GvTHNTc7hjXM.
Más funciones en Google Cloud
(default)
comentarios
GvTHNTc7hjXMOuaa3GBk
Iniciar colección
Agregar documento
Iniciar colección
comentarios
empleados
2zUb9Q5Lpp33Q1VqOSTO
78TYvoSGXAiLiVgkExWT
GvTHNTc7hjXMOuaa3GBk
Kr0Vvn3usvw42eVQ7QsZ
L94U8FogBafcCYXr8p8g
QT3fDBWMb8LDqbfd0n19
dkaK90NYkQaygKhHpU1V
pvg0EuGQkjfU8olAlybE
Agregar campo
comentario: "Hola soy dori y me encanta esta tienda de Fortnite!!!"
fecha: 26 de febrero de 2024, 2:25:22 p.m. UTC+1

**Pantalla final:** Se cierra sesión, nos despedimos con un agradecimiento por visitar nuestra tienda, y el nombre del autor



Explicacion del codigo:

**Clase main:**

La aplicación importa las bibliotecas necesarias de Flutter, incluyendo widgets y herramientas de Material Design, así como lógica específica contenida en otros archivos, como el archivo inicio.dart.

Punto de entrada principal (main()): Aquí se establece el punto de entrada principal de la aplicación Flutter. La función main() llama a runApp() con una instancia de MyApp para iniciar la aplicación.

Clase MyApp: Esta clase representa la aplicación Flutter. Se define como un StatelessWidget, lo que indica que su contenido no cambia durante el ciclo de vida de la aplicación. En el método build(), se define la estructura de la interfaz de usuario de la aplicación utilizando un MaterialApp, con la pantalla de inicio (Inicio()) como la página principal.

**Clase inicio:**

**P**antalla de inicio de la aplicación, que presenta una barra de aplicación con el título "Tienda de Fortnite" y un menú lateral. La pantalla de inicio muestra un mensaje de bienvenida y un botón flotante que lleva a los usuarios a la tienda de Fortnite cuando se presiona.

Importaciones: Importamos las bibliotecas necesarias de Flutter, así como los archivos drawer.dart y tienda.dart, que contienen la lógica para el menú lateral y la pantalla de la tienda.

Definimos una clase llamada Inicio que extiende StatelessWidget. Este widget representa la pantalla de inicio de la aplicación.

Constructor: La clase Inicio tiene un constructor que toma una clave opcional.

Método build: Este método es necesario en cualquier widget Flutter. Retorna un árbol de widgets que definen la interfaz de usuario.

MaterialApp: Retorna un MaterialApp, el widget raíz de una aplicación Flutter. Oculta la bandera de modo de depuración y establece el título de la aplicación.

Scaffold: Retorna un Scaffold, un layout de alto nivel de Material Design que implementa las estructuras básicas de diseño visual de la aplicación.

AppBar: Define una AppBar que es la barra de aplicaciones en la parte superior de la pantalla, con el título "Tienda de Fortnite".

Menú lateral: Establece el menú lateral de la AppBar utilizando la lógica definida en el archivo drawer.dart.

Cuerpo de la pantalla: Define el cuerpo de la pantalla de inicio como un contenedor con un color de fondo y un texto centrado que da la bienvenida a los usuarios a la tienda de Fortnite.

Botón de acción flotante: Define un botón flotante que, al ser presionado, navega a la pantalla de la tienda definida en el archivo tienda.dart.

**Clase drawer:**

El código define la clase MenuLateral, que representa el menú lateral de la aplicación Flutter. Este menú se utiliza para mostrar opciones de navegación o acciones adicionales al usuario.

La clase MenuLateral extiende StatelessWidget, lo que significa que su contenido no cambia durante el ciclo de vida de la aplicación. En su método build(), se define la interfaz de usuario del menú lateral.

Dentro del método build(), se crea un Drawer, que es un panel deslizable utilizado típicamente para mostrar opciones de navegación. Dentro del Drawer, se utiliza un ListView para mostrar una lista de elementos.

El primer elemento de la lista es un UserAccountsDrawerHeader, que proporciona un encabezado con información del usuario, como el nombre y el correo electrónico. Se utiliza un fondo de imagen para el encabezado, que se configura mediante DecorationImage.

Después del encabezado del usuario, se añade un elemento de lista personalizado mediante el widget Ink. Este elemento de lista tiene un fondo de tinta personalizado y contiene un ListTile con el título "Tienda de Fortnite".

**Clase tienda:**

Importa las bibliotecas necesarias de Flutter y http para realizar solicitudes HTTP y convertir las respuestas en objetos JSON.

Clase RarityData:
Define una clase RarityData que representa la rareza de los artículos en la tienda. Contiene información como el valor de visualización, el valor de fondo y el color asociado.

Clase FortniteShopItem:
Define una clase FortniteShopItem que representa un artículo en la tienda de Fortnite. Contiene información como la imagen de fondo, el nombre, el precio, la rareza, etc.

Se implementa un método fromJson para convertir los datos JSON en objetos FortniteShopItem.

Clase Tienda:
Define una clase Tienda que representa la pantalla principal de la aplicación.
Se establece la AppBar con un título y un botón de retroceso.
También se define un botón de salida que lleva a la pantalla de cierre de sesión.

Clase MyHomePage:
Define una clase MyHomePage que es un StatefulWidget que representa el cuerpo de la tienda.

En el método initState(), se inicializa la lista de artículos de la tienda y se realiza una solicitud HTTP para obtener los datos de la tienda actual.

El método build() retorna un GridView que muestra los artículos de la tienda en forma de tarjetas.
Cada tarjeta muestra la imagen del artículo, el nombre, el precio y la rareza.

Método main():
En el método main(), se ejecuta la aplicación llamando al widget Tienda.

**Clase de añadir un comentario en la base de datos:**

AddComentarioBbdd Class: Muestra una barra de navegación (AppBar) con un título y un CommentForm Widget dentro de un contenedor de color.

CommentForm Class: Esta clase define un formulario para escribir comentarios. Es un StatefulWidget porque el estado del widget (el texto del comentario) puede cambiar. En su método build(), muestra un campo de texto (TextField) para que el usuario escriba un comentario y un botón elevado (ElevatedButton) para enviar el comentario.

_CommentFormState Class: Esta clase define el estado para el formulario de comentarios. Contiene un controlador de texto para manejar el texto del comentario, un método _submitComment() para guardar el comentario en Firestore y mostrar un diálogo de confirmación, y un método build() que muestra el campo de texto y el botón.

guardarComentario() Function: Esta función toma un comentario como parámetro y lo guarda en Firestore en la colección "comentarios", junto con la fecha actual. Imprime un mensaje de éxito o error en la consola según el resultado de la operación.

## Clase final:

El código define una clase llamada pantallaFinal, que representa la pantalla de cierre de sesión de la aplicación. Esta pantalla se muestra cuando el usuario cierra sesión en la aplicación.

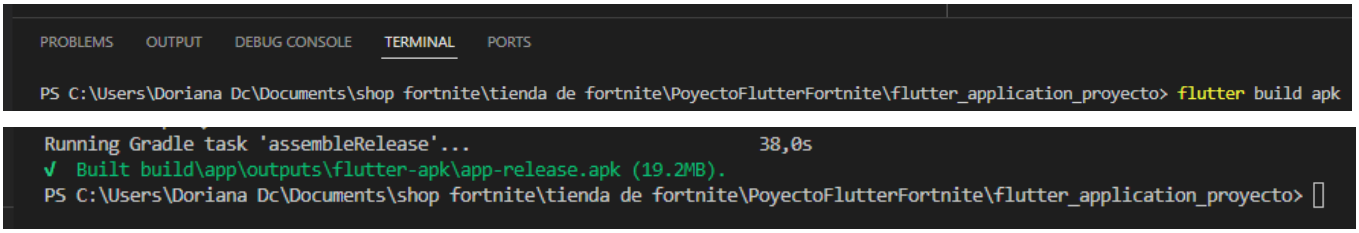
La clase pantallaFinal extiende StatelessWidget, lo que significa que su contenido no cambia durante el ciclo de vida de la aplicación. Dentro de esta clase, se implementa el método build() que define la interfaz de usuario de la pantalla.

El cuerpo de la pantalla está contenido en un Container, que tiene un color de fondo rosa. Dentro de este contenedor, hay un Center que alinea el contenido al centro vertical y horizontalmente.

Dentro del Center, se encuentra un Column que contiene los elementos de la pantalla: un icono de "pulgar hacia arriba" de gran tamaño, seguido de dos líneas de texto. El primer texto agradece al usuario por visitar la tienda, mientras que el segundo texto indica el nombre del autor de la aplicación.

Creamos la APK:

Ejecutamos en la terminal el comando: flutter build apk




Ya la tenemos en el proyecto: