Instituto Tecnológico y de Estudios Superiores de Occidente.

ITESO

Universidad Jesuita de Guadalajara

Practia 01

Materia: PROGRAMACION DE DISPOSITIVOS MOVILES

Profesor: Francisco Javier Camacho Gil

Fecha: 12 de febrero de 2021

Autor(es): Armando Emmanuel Correa Amorelli

Introducción

El objetivo de esta práctica es realizar una aplicación con diseño basado en los lineamientos de material design y siguiend los mockups otorgados.

Desarrollo

Para el desarrollo de esta actividad, se partío de una estructura proporcionada por el profesor. Y se analizó para así poder decidir que se realizaría primero, ya que faltaba una cantidad considerable de trabajo a realizar.

Dicho esto, y considerando la tematica de la aplicación, se tomó la decisión de empezar la implementación de la lista de bebidas calientes. Ya que esta misma estructura se puede repetir apra lospostres y los granos.

Vista de bebidas calientes

Para la vista de bebidas calientes, se optó por seguir el mismo diseño de la ventana principal. Con la diferencia de haberle añadido el tipo de de bebida caliente y el precio unitario, como se puede observar en la Figura 1.

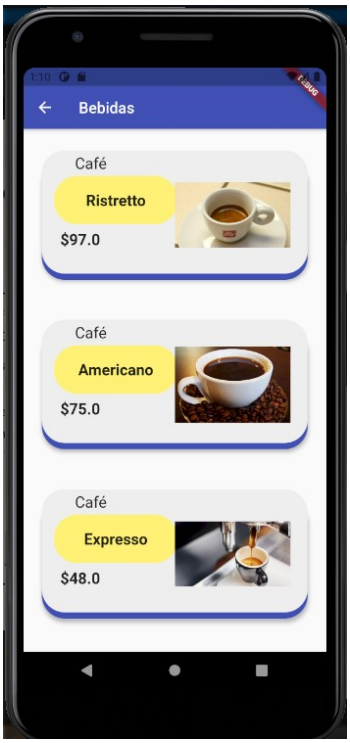


Figura 1: Lista de bebidas calientes

Una vez terminada esta vista, se implementó el mismo diseño para postres y granos.

Drawer

Despues de terminar la lista de bebidas, se implementó un drawer que nos permite visualizar el perfin sin necesidad de cambiar de ventana, cómo se puede observar en la Figura 2

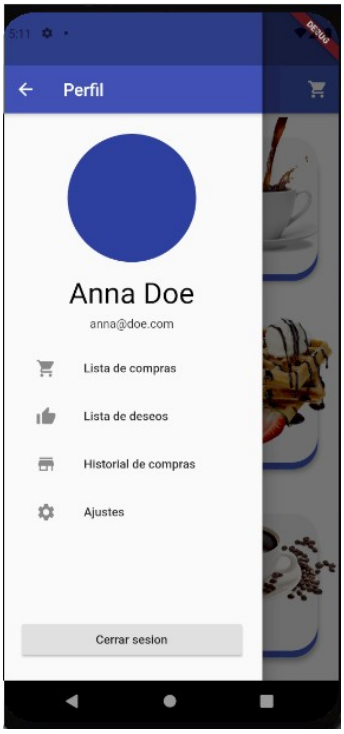


Figura 2: Drawer de perfil

Slplash screen

Tras haber terminado las listas de productos e implementar el drawer, se decidió empezar la implemetación del Splash screen (Figura 3), ya que este no se había realizado de manera previa en ninguna actividad de clase.
El resultado al que se llegó fue el utilizar el plugin splashscreen, el cual nos permite añadir una ventana de carga, con una duración definida por tiempo o un evento futuro. Aunque por el momento no se le pueda utilizar al máxmo, su implementación por tiempo nos permite obtener un prototipo más apegado a la realidad que con cualquér otro metodo del cual se tuviera conocimiento

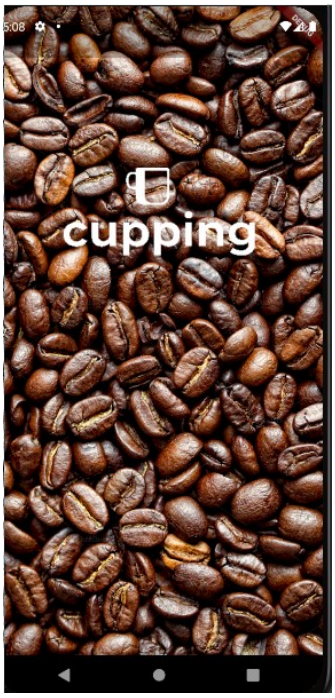


Figura 3: Splash Screen

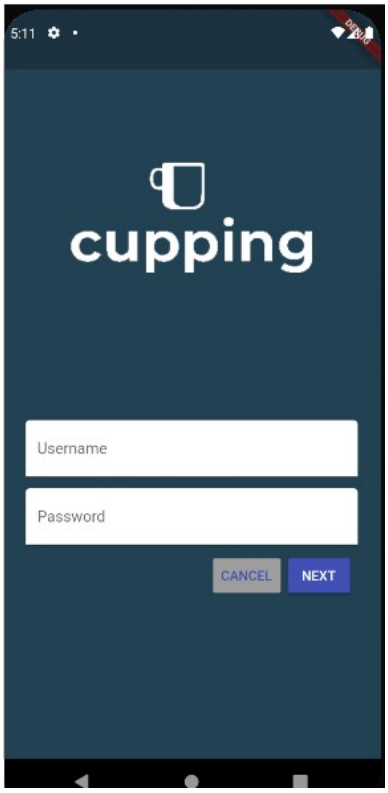
Ventana de inicio, LogIn y Registro

Ya implementado el Slash Screen, se diseñó la ventana de inicio, la cual cuenta con dos botones, los cuales nos dejan elegir entre registrarnos (nuevos usuarios) o ingresar (usuariso existentes)

Figura 4: ventana de inicio

La primera de estas cuenta con 3 campos de ingreso de texto, siendo estos en orden: Nombre completo, Email y contraseña. Seguidos de un checkbox para inicar la aceptación de terminos y condiciones de uso. Y por ultimo un unico boton para terminar el registro(Figura 6).

Mientras que la segnda de estas, solo cuenta con dos campos de texto, el de nombre de usuari y el de contraseña. Junto con un botón para continuar y otro para cancelar el inicio de seción(Figura 5).



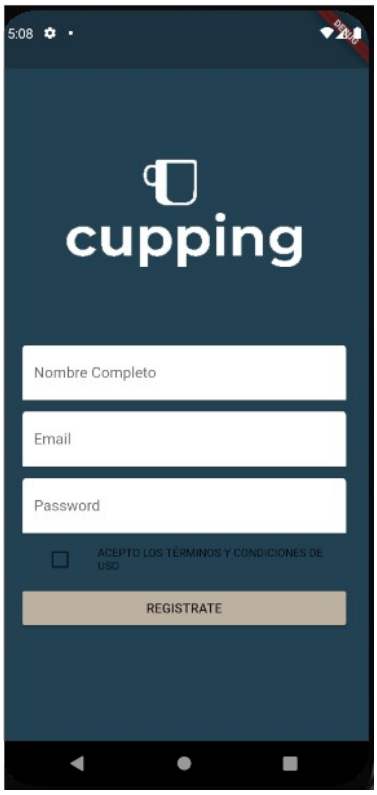


Figura 5: inicio de seción *Figura 6: Registro*

Snackbar "Proximamente"

Una vez terminadas estas ventanas, se implementó un snackbar que despliega el mensaje "Proximamente!!!", para que sea mostrado cada que se pulse en algúna proxima categoría.
Para realizar esto se tuvo que implementar un scafold key de la siguiente manera

```
final _scaffoldKey = GlobalKey<ScaffoldState>();
```

Esto debido a que el scaffols se encontraba inaccesible desde el contexto.

Detalles del producto

Lo siguiente que se realizó fue el diseño de la ventana en la cual se despliegaría el producto selecionado por el usuario. Esta despliega la imagen del producto, seguida de su Nombre y la descripción del mismo. De igual manera muestra 3 selectores de tipo chip, los cuales alternan su estado entre ellos, dependiendo de cual esté seleccionado. Esto se logró de la siguiente manera

```
onSelected: (bool sel) {
    setState(() {
        if (sel != _chico && true == sel) {
            widget.drink.productSize = ProductSize.CH;
```

```
            widget.drink.productPrice =
widget.drink.productPriceCalculator();
            _chico = sel;
            _mediano = !sel;
            _grande = !sel;
        }
    });
}
```

Otro elemento que alterna su estado es el icono de “favoritos” que se encuentra el na parte superior derecha de la carta del producto.

Figura 7: vista detallada del producto

Paso de argumentos entre ventanas

El siguente objetivo a cumplir fue el paso de datos entre ventanas, conforme se fuera navegando, primero se intentento ingresar distintos tipos de objetos dentro del elemento `arguments:` Lo cual no funcionó, por lo que se opotó por realizar la implementación

demostrada dentro de la documentación de Flutter, lo cual era crear una nueva clase que contenga todos los elementos deseados para su transmición como un unico objeto. La clase creada es la siguiente:

```
class ProductCart{
  List<ProductHotDrinks> drinks;
  List<ProductGrains grains;
  List<ProductDesserts> desserts;

  ProductCart({
    this.drinks,
    this.grains,
    this.desserts,
  });
}
```

Carrito de compras y ventana de pago

Lo ultimo que se realizó fue la implementación del carrito de compra, al cuál se le añadió un condicional y una variable local de tipo bool, el cual nos permite cambiár la pagina por completo dependiendo si hay o no productos en el carrito de compras.

El único problema que se encontró en este diseño fue el paso de datos ya que la clase creada previamente es distinta a la clase que recibe la ventana de carrito. Por lo que se tuvo que implementar una finción que realizara el vaciado de las listas contenidas en la clase ProductCart ha una unica lista que recibe la ventana del carrito.

Figura 8: carrito de compras vacio

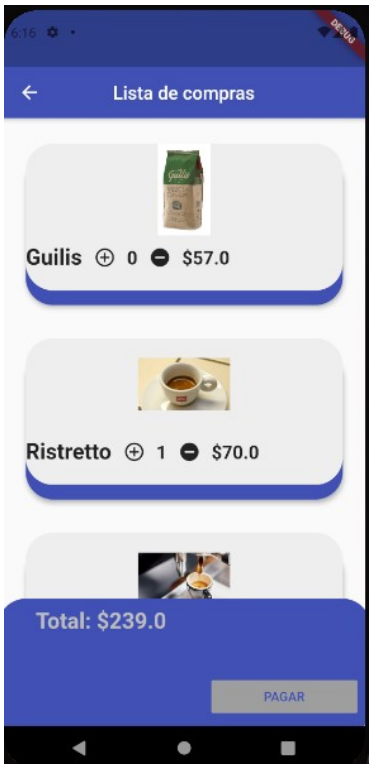


Figura 9: Carrito de compras con productos

Una ves terminado el carrito de compra se procedió a realizar la ventana de pagos, la cual despliega las 3 opciones de pago aceptadas, e total y la imagen y nombre de productos(por el momento solo muestra 1 producto por cuestiones de tiempo).

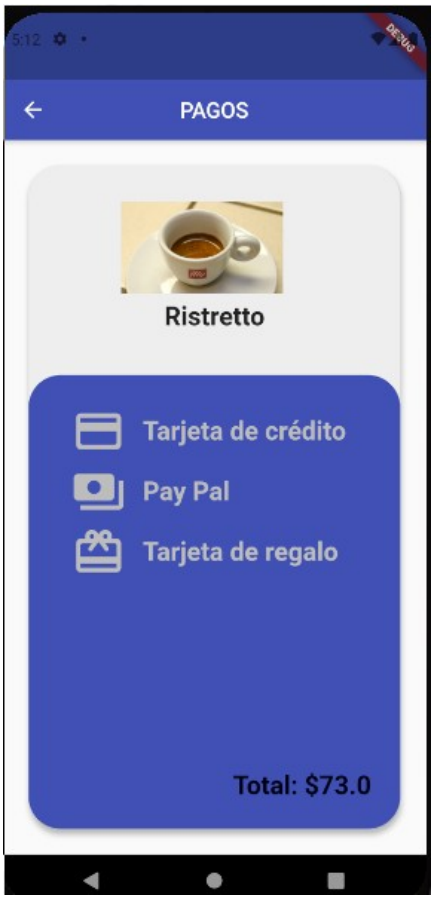


Figura 10: Ventana de pagos

Conclusiones

Mediante la realización de esta practica, se puede percatar de dificultad que conlleva el diseñar interfaces de usuario. Ya que, en lo personal, se estaba más preocupado por la comunicación entre ventanas que por el diseño de la UI, pero resulto esta ultima ser más complicada que las demás ya que se necesita estar bien documentado en cuanto a las capacidades de cada widget y sus limitaciones para poder realizar un diseño atractivo y simple de entender.

Observaciones

Se aprendió la importancia de definir una key para el scaffold cuando se anidan varios widgets distintos. También que no siempre se puede extraer un conjuntos de widgets como otro unificado.

En cuanto a las dificultades presentadas, se pueden resumir en 2, el utilizar una imagen de tipo network como fondo para el spashScreen y la correcta conjunción de widgets para

la creación de UI.
El primero se soluciono mediante la adición de “.image” al final del “Image.network”, mientras que el segundo no quedó completamente aprendido, ya que es cuestión de experiencia el poder utilizar las herramientas visuales al máximo.

Repositorio
Video

Bibliografía

[1]"Add a Drawer to a screen", *Flutter.dev*, 2021. [Online]. Available: https://flutter.dev/docs/cookbook/design/drawer#2-add-a-drawer. [Accessed: 12- Feb- 2021].

[2]"Login Page UI in Flutter » Flutter Overflow - Flutter Login", *Flutter Overflow*, 2021. [Online]. Available: https://flutteroverflow.com/login-page-ui-in-flutter/. [Accessed: 12- Feb- 2021].

[3]"splashscreen | Flutter Package", *Dart packages*, 2020. [Online]. Available: https://pub.dev/packages/splashscreen. [Accessed: 12- Feb- 2021].

[4]F. ImageProvider, M. Peterson and D. S., "Flutter Image object to ImageProvider", *Stack Overflow*, 2021. [Online]. Available: https://stackoverflow.com/questions/58870443/flutter-image-object-to-imageprovider. [Accessed: 12- Feb- 2021].

[5]L. Mbele, "Scaffold.of() called with a context that does not contain a Scaffold", *Stack Overflow*, 2021. [Online]. Available: https://stackoverflow.com/questions/51304568/scaffold-of-called-with-a-context-that-does-not-contain-a-scaffold. [Accessed: 12- Feb- 2021].

[6]"Send data to a new screen", *Flutter.dev*. [Online]. Available: https://flutter.dev/docs/cookbook/navigation/passing-data. [Accessed: 12- Feb- 2021].

[7]"Return data from a screen", *Flutter.dev*. [Online]. Available: https://flutter.dev/docs/cookbook/navigation/returning-data. [Accessed: 12- Feb- 2021].

Criterio de evaluación ponderado.

	Descripción	Puntos alcanzables	Puntos obtenidos	Observaciones
UI (layouts, themes, fonts)	UI similar a la solicitada en el mockup de la primera propuesta (imágenes, texto, themes, appbar, listas).	**25 pts**	20	Se restan **8 pts** en caso de no utilizar theme.
Manejo de estados	Se cambia el botón de like, al igual que los textos de cantidad y costo total, además, se maneja el estado para mostrar buena interacción con el usuario.	**20 pts**	18	Se restan **16 pts** en caso de no agregar y quitar productos, así como actualizar el total en **el carrito**.
Navegación	Navegación similar a la solicitada (en caso de ser necesario pasar y recuperar datos).	**30 pts**	30	
Funcionalidad	Basarse en recursos como mockups de muestra de la app además de explicación del profesor.	**20 pts**	20	
Video	Link de video con su app funcionando.	**5 pts**	5	Si la app no tiene el funcionamiento correcto, únicamente se obtiene **1 pt**