

Módulo 2: Fundamentos en el Diseño de Aplicaciones Móviles

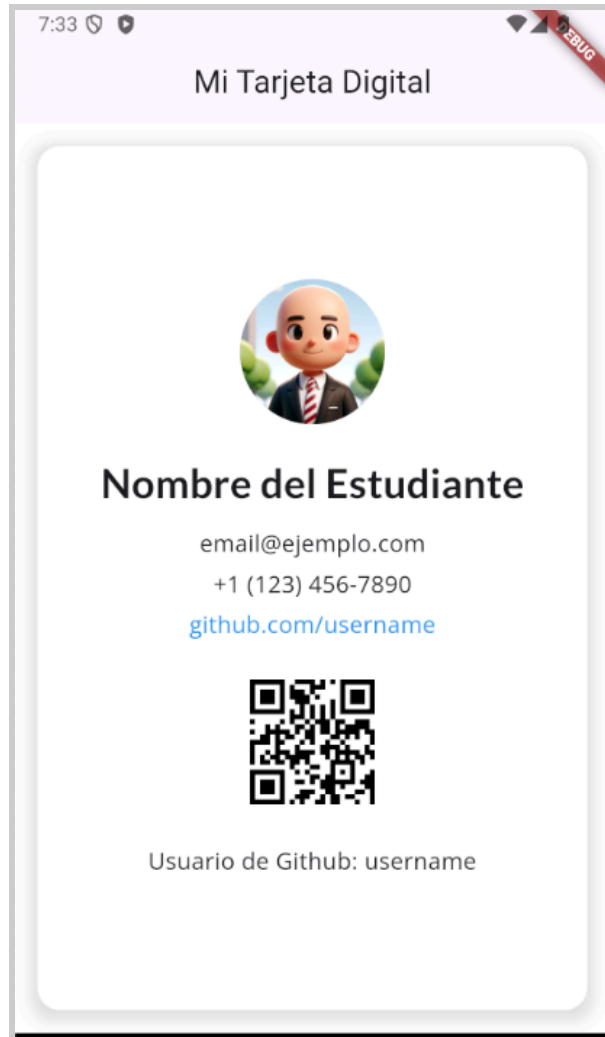
Actividad Especial: Creación de una V-Card con Flutter

Objetivos de la Actividad

- Crear una aplicación simple en Flutter que despliegue una V-Card con información personal estática.
- Utilizar widgets básicos de Flutter como Text, Image, Column, Row, SizedBox y Expanded para estructurar el diseño.
- Implementar tipografía personalizada utilizando la biblioteca Google Fonts.
- Incorporar una imagen de perfil y un código QR que enlace a la cuenta de Github del estudiante.

Introducción

En esta actividad, aprenderás a construir una V-Card (tarjeta de presentación virtual) en Flutter. Este tipo de aplicación es excelente para explorar los fundamentos del diseño de interfaces simples y atractivas. La aplicación que desarrollaremos contará con una pantalla estática donde se presentarán detalles del usuario, una imagen de perfil y un QR que redirige al perfil de Github del estudiante.



La imagen presenta una captura de la pantalla de la aplicación.

Creado por Javier A. Dastas (2024).

Desarrollo del Tema Central

1. Preparación del Entorno

- Asegúrate de tener Visual Studio Code instalado y configurado para el desarrollo en Flutter.
- Crea un nuevo proyecto en Flutter desde Visual Studio Code llamado vcard_app.

2. Estructura de la Aplicación

La estructura básica de nuestra aplicación incluirá una única pantalla (`Scaffold`) que contendrá una columna principal con el contenido de la V-Card.

3. Importar Dependencias

Para utilizar fuentes personalizadas, añadiremos la biblioteca `google_fonts` a nuestro proyecto:

Abre el archivo `pubspec.yaml` que está en tu proyecto y agrega la dependencia de `google_fonts` al archivo:

`dependencies:`

`flutter:`

`sdk: flutter`

`google_fonts: ^6.2.1`

Importante: Verifica que la línea **`google_fonts: ^6.2.1`** esté indentada al mismo nivel que **`flutter:`**.

4. Creación del Widget Principal

```
import 'package:flutter/material.dart';
```

```
import 'package:google_fonts/google_fonts.dart';
```

```
void main() => runApp(VCardApp());
```

```
class VCardApp extends StatelessWidget {
```

```
  @override
```

```
  Widget build(BuildContext context) {
```

```
    return MaterialApp(
```

```
      title: 'V-Card',
```

```

        theme: ThemeData(primarySwatch: Colors.blue),
        home: VCardScreen(),
    );
}
}

```

5. Diseño de la V-Card

En este diseño utilizaremos una columna para organizar el contenido verticalmente y un `SizeBox` para agregar espaciado entre los elementos.

```

import 'package:flutter/material.dart';
import 'package:google_fonts/google_fonts.dart';

class VCardScreen extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text('Mi Tarjeta Digital'),
        centerTitle: true,
      ),
      backgroundColor: Colors.white,
      body: Center(
        child: Padding(
          padding: const EdgeInsets.all(16.0),
          child: Container(
            decoration: BoxDecoration(
              color: Colors.white,

```

```

borderRadius: BorderRadius.circular(15),
boxShadow: [
  BoxShadow(
    color: Colors.grey.withOpacity(0.3),
    spreadRadius: 5,
    blurRadius: 7,
    offset: Offset(0, 3), // Desplazamiento de la
sombra
  ),
],
),
child: Padding(
  padding: const EdgeInsets.all(20.0),
  child: Column(
    mainAxisAlignment: MainAxisAlignment.center,
    children: [
      CircleAvatar(
        radius: 50,
        backgroundImage:
AssetImage('assets/profile.jpg'), // Imagen del perfil
      ),
      SizedBox(height: 20),
      Text(
        'Nombre del Estudiante',
        style: GoogleFonts.lato(fontSize: 28,
fontWeight: FontWeight.bold),
      ),
    ],
  ),
),

```

```

        SizedBox(height: 10),
        Text(
          'email@ejemplo.com',
          style: GoogleFonts.openSans(fontSize: 16),
        ),
        SizedBox(height: 5),
        Text(
          '+1 (123) 456-7890',
          style: GoogleFonts.openSans(fontSize: 16),
        ),
        SizedBox(height: 5),
        Text(
          'github.com/username',
          style: GoogleFonts.openSans(fontSize: 16,
color: Colors.blue),
        ),
        SizedBox(height: 20),
        // Código QR (puede ser una imagen estática o
generada por un paquete QR)
        Image.asset(
          'assets/qrcode.png',
          width: 100,
          height: 100,
        ),
        SizedBox(height: 20),
        // Botón de Expansión (solo como ejemplo
visual)

```

```

Row(
  mainAxisAlignment: MainAxisAlignment.center,
  children: [
    Expanded(
      child: Text(
        'Usuario de Github: username',
        style: GoogleFonts.openSans(fontSize:
16),

        textAlign: TextAlign.center,
      ),
    ),
  ],
),
],
),
],
),
),
),
),
),
),
);
}
}

```

6. Preparar los Recursos

1. Crea una carpeta en tu proyecto llamada “assets”.
2. Coloca una foto de perfil en `assets/profile.jpg`.
3. Agrega una imagen con código QR en `assets/qrcode.png`. Este QR debe redirigir a tu cuenta de Github.

Actualiza el archivo `pubspec.yaml` para incluir los recursos:

```
flutter:
```

```
  assets:
```

- `assets/profile.jpg`
- `assets/qrcode.png`

Recuerda que los nombres de las imágenes corresponden a:

- `profile.jpg`: Una foto o imagen de algún Avatar para tu perfil de usuario. Puedes generar un Avatar o imagen utilizando ChatGPT. De igual manera, puedes darle una foto tuya e indicarle que genera algún Avatar o caricatura.
- `qrcode.png`: Una imagen con código QR que contenga la dirección de URL de tu Github. Puedes utilizar el sitio web <https://www.qrcode-monkey.com/> para generar la imagen de qrcode.

7. Probar la Aplicación

Asegúrate de probar la aplicación en algún dispositivo móvil o emulador de dispositivo móvil configurado en tu editor.

Ejemplos y Explicaciones Detalladas

- **Text Widgets**: Cada elemento de información utiliza el widget `Text` con estilos personalizados para mejorar la estética.

- **Column y Row:** La columna organiza los elementos verticalmente, mientras que la fila dentro de la columna permite centrar el usuario de Github.
- **CircleAvatar:** Crea una imagen circular, ideal para una foto de perfil.
- **SizedBox:** Añade espacio entre los widgets para un diseño limpio y equilibrado.
- **Expanded:** Permite que el widget de usuario se expanda horizontalmente en el centro de la pantalla.
- **Container:** Para contener diferentes elementos y aplicar estilos como márgenes, paddings, y colores de fondo.
- **BoxDecoration:** Para decorar elementos, por ejemplo, aplicando bordes redondeados al Container de la foto de perfil.

Explicación del BoxDecoration

- **color:** Establece el color de fondo del Container (en este caso, blanco).
- **borderRadius:** Aplica bordes redondeados para dar un aspecto moderno y suave.
- **boxShadow:** Añade la sombra. Los parámetros principales son:
 - **color:** Color de la sombra (gris claro con opacidad del 30%).
 - **spreadRadius:** Controla la expansión de la sombra.
 - **blurRadius:** Controla el desenfoque de la sombra.
 - **offset:** Define la dirección de la sombra, en este caso, hacia abajo y ligeramente a la derecha.

Este diseño le da un aspecto moderno y elegante a la tarjeta, con una sombra sutil para mejorar la apariencia.

Relación con Otros Conceptos o Lecciones

Este tutorial sienta las bases para futuras aplicaciones móviles con diseño responsive y mayor interactividad, ideales para introducir las prácticas de diseño de interfaz y experiencia de usuario en Flutter.

Resumen

Hemos desarrollado una aplicación en Flutter que muestra una V-Card personal con nombre, correo electrónico, teléfono, usuario de Github, una imagen de perfil y un código QR. Utilizamos widgets básicos como `Text`, `Image`, `Column`, `Row`, `SizedBox` y `Expanded`, y añadimos fuentes personalizadas con Google Fonts para darle un toque profesional.

Actividad Adicional

1. Cambia el color de fondo de la aplicación y personaliza los colores del texto y el diseño.
 - a. Para cambiar el color de fondo de la aplicación debes utilizar la propiedad “backgroundColor” en el widget “Scaffold”.

Lista de colores recomendados para un Diseño Moderno:

1. **Blanco Suave** - Color(0xFFFF5F5F5)
2. **Azul Pizarra Claro** - Color(0xFFE0F7FA)
3. **Gris Oscuro** - Color(0xFF263238)
4. **Rosa Claro** - Color(0xFFFFCDD2)
5. **Verde Menta** - Color(0xFFD1F2EB)
6. **Amarillo Pastel** - Color(0xFFFFF9C4)
7. **Lavanda** - Color(0xFFE1BEE7)
8. **Gris Cálido** - Color(0xFFE0E0E0)
9. **Aqua Suave** - Color(0xFFB2EBF2)
10. **Coral Ligero** - Color(0xFFFF8A65)