

Aplicación APP Multiplataforma

Programación en Dispositivos Móviles

Presentado por:

Edna Carolina García Holguín

Universidad Internacional de Valencia

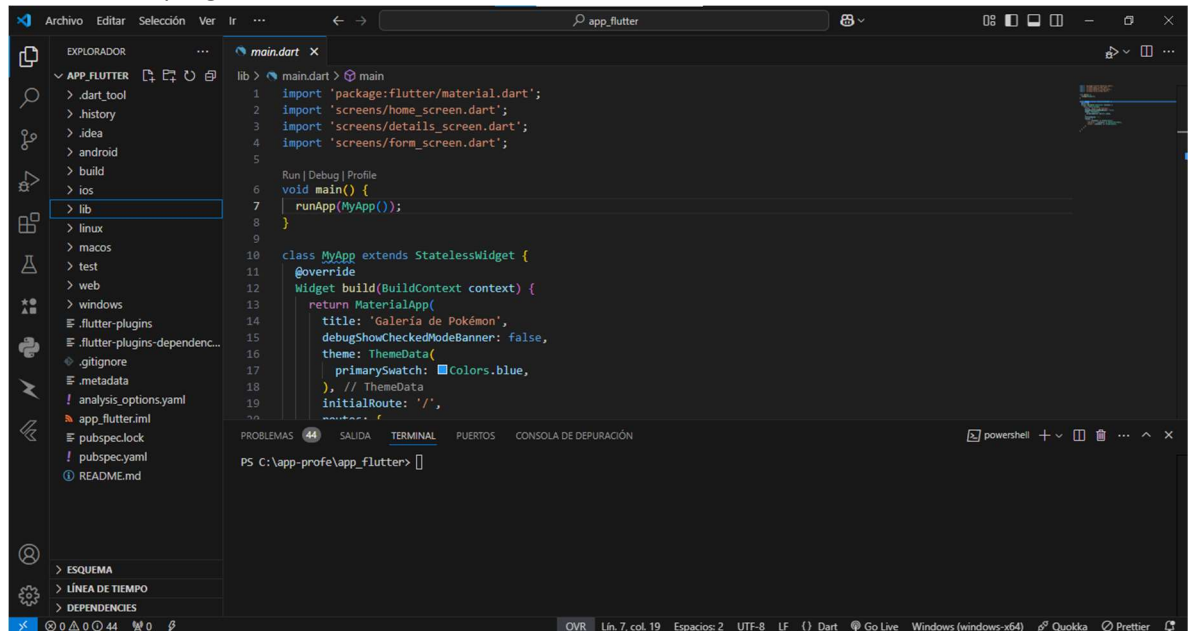
Explicación del Aplicativo:

La base de desarrollo del aplicativo móvil se realizó en un framework de desarrollo de aplicaciones de código abierto llamado flutter, el cual fue creado por Google sirve para el desarrollo de aplicaciones nativas para dispositivos móviles, web y de escritorio a partir de una única base de código, permite la creación de interfaces de usuario interactivas y altamente personalizadas mediante el uso del lenguaje de programación dart.

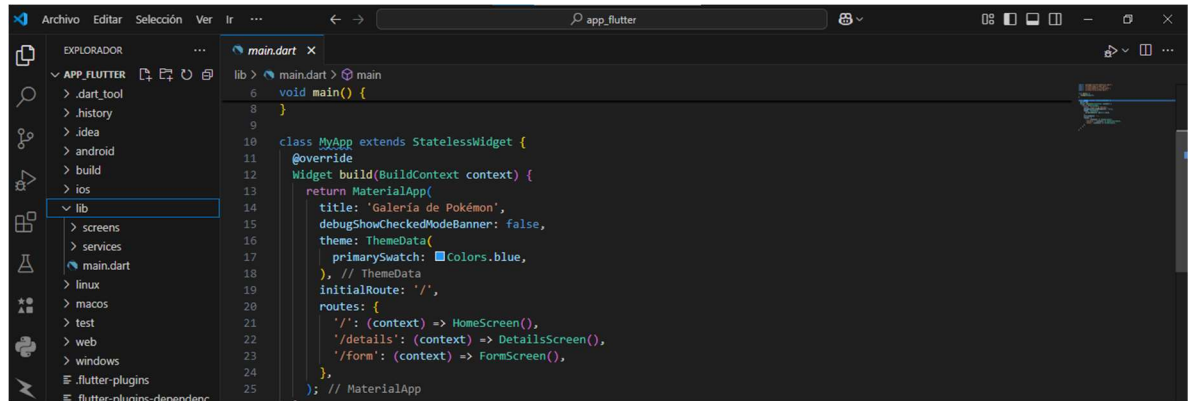
Para el desarrollo de la aplicación se instaló y se configuro el framework flutter, se descargó el instalador de flutter para Windows el cual nos descarga un archivo comprimido, el cual lo ubicamos en el directorio raíz de la máquina, continuando con la configuración de flutter realizamos el direccionamiento de la solución al path, después se descarga y se instala el GIT que es un requerimiento de flutter, para la ejecución de los modelos o el emulador de Android para que pueda ejecutar el programa o software, continuando para el desarrollo de la aplicación en Flutter se trabajó con el editor de trabajo visual studio code el cual se le adicionaron la extensiones de flutter.

La aplicación se llama pokemon, esta aplicación sirve para buscar el pokemon que se está preguntando, nos trae la imagen y los datos de la caricatura.

1. Creación del programa en visual studio code.



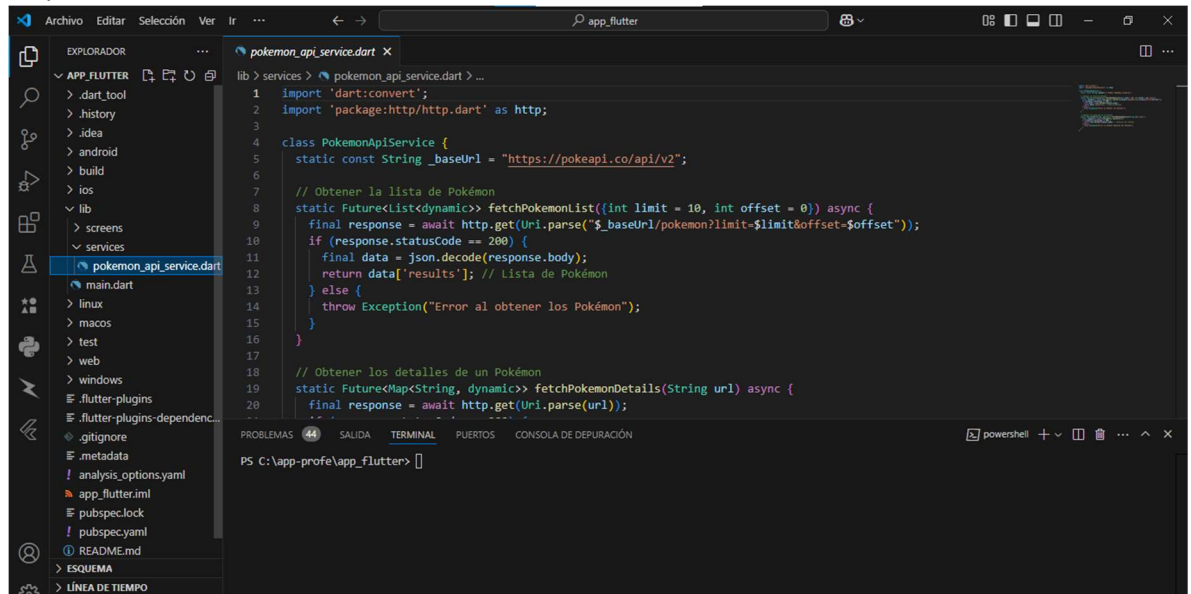
2. Buscamos la carpeta lib y buscamos el archivo main.dart donde se encuentra la configuración inicial, la estructura del proyecto el cual sirve para la ejecución del proyecto.



The screenshot shows an IDE with the Explorer panel on the left displaying the project structure. The 'lib' directory is expanded, showing 'main.dart' selected. The main.dart file is open in the editor, showing the following Dart code:

```
lib > main.dart > main
6 void main() {
8 }
9
10 class MyApp extends StatelessWidget {
11   @override
12   Widget build(BuildContext context) {
13     return MaterialApp(
14       title: 'Galería de Pokémon',
15       debugShowCheckedModeBanner: false,
16       theme: ThemeData(
17         primarySwatch: Colors.blue,
18       ), // ThemeData
19       initialRoute: '/',
20       routes: {
21         '/': (context) => HomeScreen(),
22         '/details': (context) => DetailsScreen(),
23         '/form': (context) => FormScreen(),
24       },
25     ); // MaterialApp
```

3. Para el desarrollo de la aplicación pokemon utilice un api, esta se encuentra ubicada en la carpeta services.



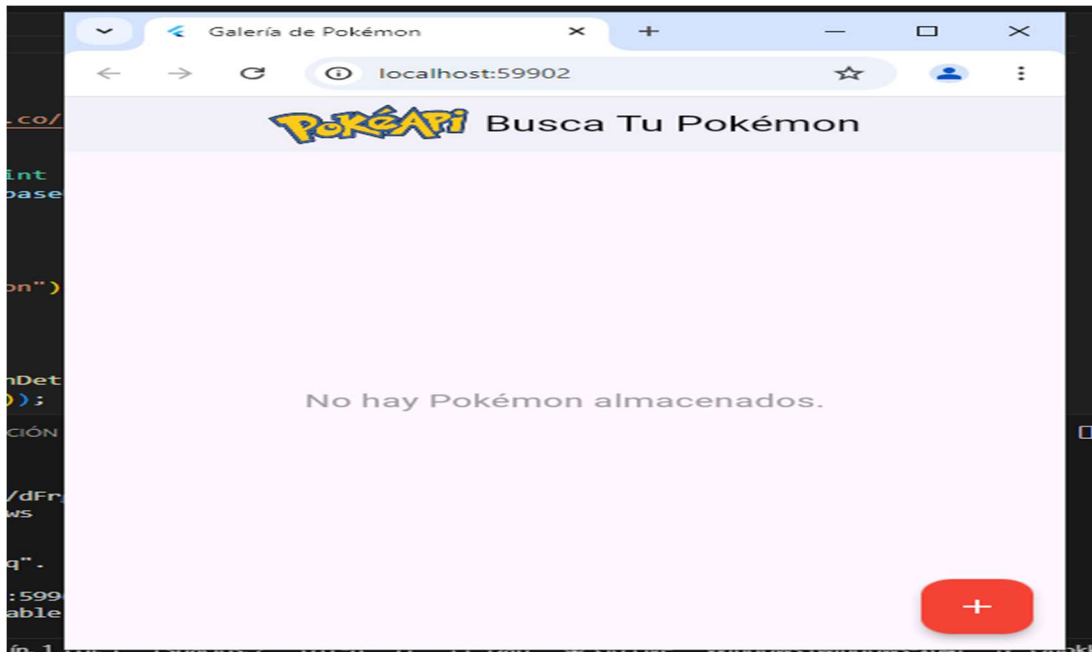
The screenshot shows the same IDE with the Explorer panel on the left. The 'services' directory is expanded, showing 'pokemon_api_service.dart' selected. The file is open in the editor, showing the following Dart code:

```
lib > services > pokemon_api_service.dart > ...
1 import 'dart:convert';
2 import 'package:http/http.dart' as http;
3
4 class PokemonApiService {
5   static const String _baseUrl = "https://pokeapi.co/api/v2";
6
7   // Obtener la lista de Pokémon
8   static Future<List<dynamic>> fetchPokemonList({int limit = 10, int offset = 0}) async {
9     final response = await http.get(Uri.parse("${_baseUrl}/pokemon?limit=$limit&offset=$offset"));
10    if (response.statusCode == 200) {
11      final data = json.decode(response.body);
12      return data['results']; // Lista de Pokémon
13    } else {
14      throw Exception("Error al obtener los Pokémon");
15    }
16  }
17
18  // Obtener los detalles de un Pokémon
19  static Future<Map<String, dynamic>> fetchPokemonDetails(String url) async {
20    final response = await http.get(Uri.parse(url));
21    ...
22  }
```

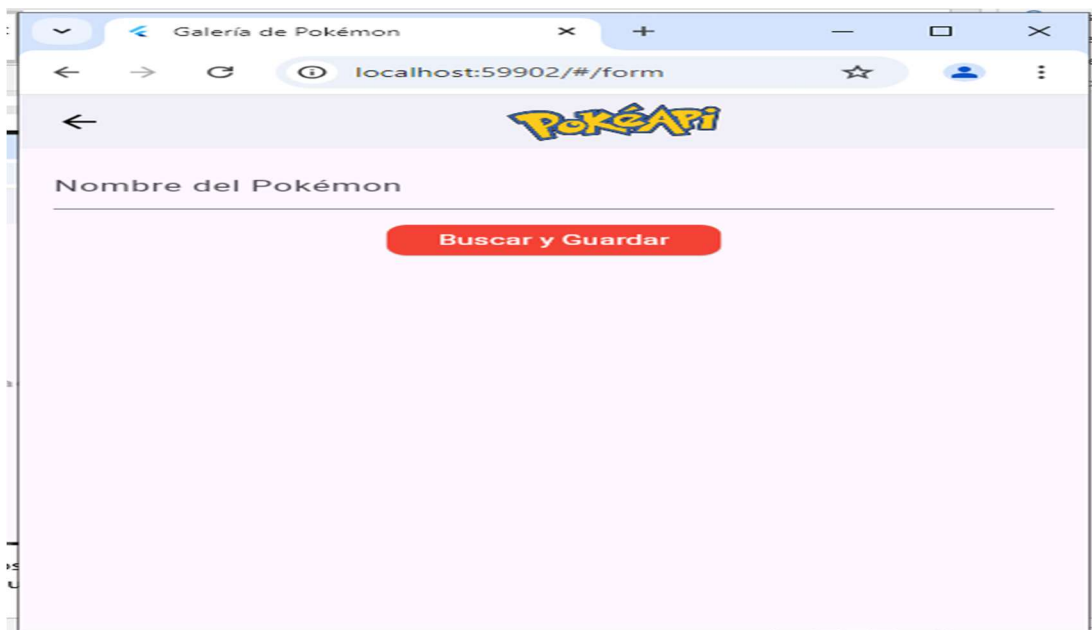
At the bottom of the IDE, the Terminal panel is visible, showing the command prompt: `PS C:\app-profe\app_flutter>`

Ejecución del programa

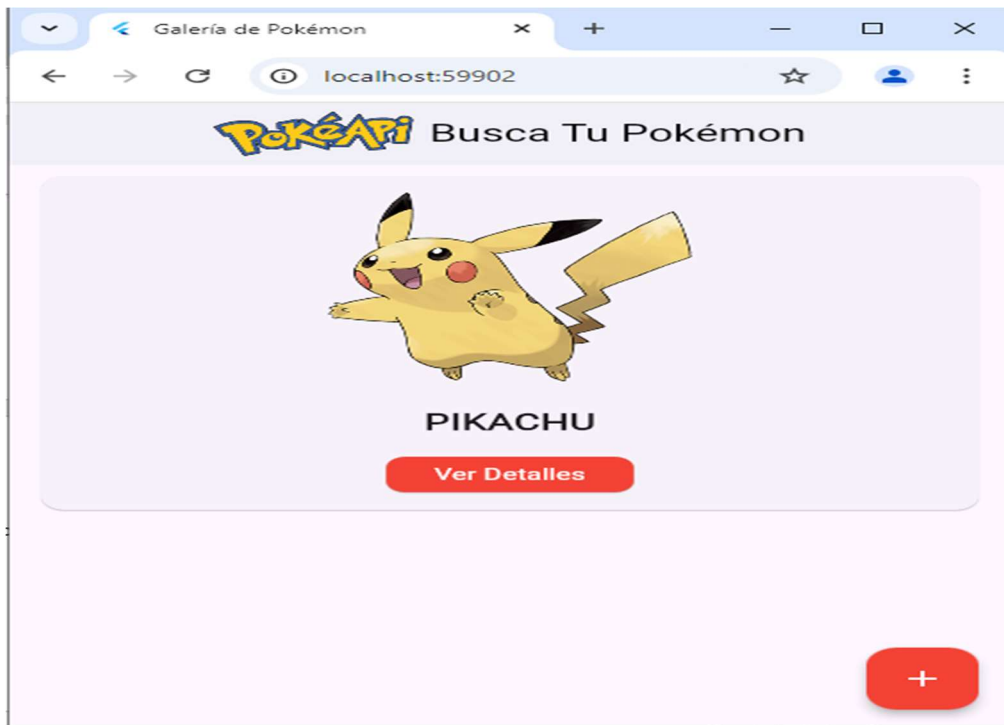
La primera página que encontramos en nuestro programa de pokemon, nos indica que no hemos buscado ningún pokemon y no está almacenado.



Para buscar un pokemon vamos al botón del mas y nos aparece una segunda pantalla, con input donde se debe digitar el nombre del dibujo y el botón buscar y guardar.



Al buscar uno de los dibujos en el input, en este caso buscamos el pokemon pikachu, nos trae la imagen de la caricatura y botón para ver los detalles del mismo.



Para ver los detalles lo mostramos en otra pantalla

