

## Solicitud de información a API RESTful

usado: <https://dattebayo-api.onrender.com/characters>

método usado: http.get()

api\_services.dart:

```
import 'dart:convert';
import 'package:http/http.dart' as http;
import 'naruto_character.dart';

class ApiService {
  static const String baseUrl = 'https://dattebayo-api.onrender.com/characters';

  static Future<List<NarutoCharacter>> fetchCharacters() async {
    final response = await http.get(Uri.parse(baseUrl));

    if (response.statusCode == 200) {
      final Map<String, dynamic> jsonData = json.decode(response.body);
      final List<dynamic> jsonList = jsonData['characters'];

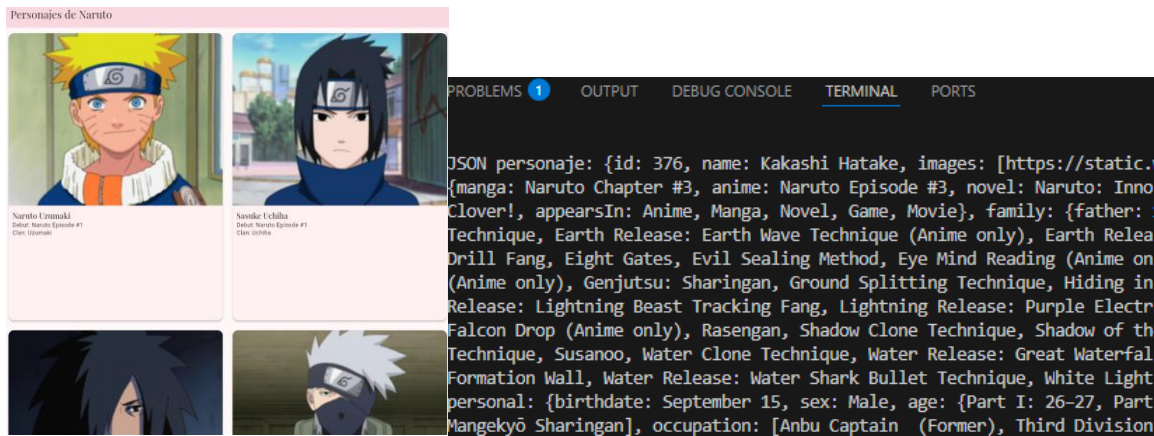
      for (var item in jsonList.take(5)) {
        print('JSON personaje: $item\n');
      }
      return jsonList.map((json) => NarutoCharacter.fromJson(json)).toList();
    } else {
      throw Exception('Error al cargar los personajes');
    }
  }
}
```

En el home page se cumple con:

- mostrar datos
- presentarlos con nombre e imagen
- y manejarlos con FutureBuilder

```
body: FutureBuilder<List<NarutoCharacter>>({
  future: _futureCharacters,
  builder: (context, snapshot) {
    if (snapshot.connectionState == ConnectionState.waiting) {
      return const Center(child: CircularProgressIndicator());
    } else if (snapshot.hasError) {
      return Center(child: Text('Error: ${snapshot.error}'));
    } else if (!snapshot.hasData || snapshot.data!.isEmpty) {
      return const Center(child: Text('No se encontraron personajes.'));
    }

    final characters = snapshot.data!;
    return ListView.builder(
      itemCount: characters.length,
      itemBuilder: (context, index) {
        final char = characters[index];
        return Card(
          margin: const EdgeInsets.symmetric(vertical: 6, horizontal: 12),
          elevation: 2,
          child: ListTile(
            leading: ClipRect(
              borderRadius: BorderRadius.circular(8),
              child: Image.network(
                char.image,
                width: 50,
              ),
            ),
            title: Text(char.name),
          ),
        );
      },
    );
  },
),
```



Validamos con código de respuesta:

```
if (response.statusCode == 200) {
    final Map<String, dynamic> jsonData = json.decode(response.body);
    final List<dynamic> jsonList = jsonData['characters'];

    for (var item in jsonList.take(5)) {
        print('JSON personaje: $item\n');
    }
    return jsonList.map((json) => NarutoCharacter.fromJson(json)).toList();
} else {
    throw Exception('Error al cargar los personajes');
}
```

Transformación y uso del JSON

```
final List<dynamic> jsonList = jsonData['characters'];

for (var item in jsonList.take(5)) {
    print('JSON personaje: $item\n');
}

return jsonList.map((json) => NarutoCharacter.fromJson(json)).toList();
```

Esto transforma el JSON a una lista de objetos NarutoCharacter