

APLICACIÓN MÓVIL UTILIZANDO FLUTTER & API RESTFUL IMPLEMENTADA EN PYTHON

Descripción general del proyecto

Se propone el desarrollo de una **aplicación móvil utilizando Flutter**, que consumirá servicios desde una **API RESTful implementada en Python (Flask)**. Esta API se conectará con una base de datos MySQL cuya estructura dependerá del enfoque específico de cada proyecto.

Tecnologías a utilizar

- **Frontend:** Flutter
- **Backend:** Python - Flask
- **Base de datos:** MySQL (estructura variable según proyecto)
- **API REST:** Comunicación mediante JSON

Diseño y experiencia de usuario (UI/UX)

- Se implementará un diseño atractivo y funcional en Flutter.
- Se utilizarán **paletas de colores armoniosas** y consistentes, siguiendo principios de diseño modernos.
- Las vistas estarán organizadas de forma clara, con componentes reutilizables y adaptativos.
- Se asegurará una navegación intuitiva entre pantallas como login, dashboard, tareas, y más.

Documentación y consumo de API desde Flutter

Cada **ENDPOINT** desarrollado contará con su respectiva documentación, que incluirá:

- **Ruta del endpoint**
- **Método HTTP utilizado** (GET, POST, PUT, DELETE, etc.)
- **Parámetros requeridos** (en el path, query o cuerpo)
- **Tipo de datos esperados**
- **Formato de respuesta**

Organización del proyecto

El desarrollo estará coordinado por un **líder de proyecto**, quien tendrá las siguientes responsabilidades:

- **Asignar tareas** a los distintos miembros del equipo (backend, frontend, diseño, QA).
- **Coordinar entregas** parciales y revisión de avances.
- **Validar el cumplimiento de los endpoints y criterios funcionales.**
- Supervisar la elaboración de la **documentación técnica y de usuario.**

Entrega esperada

- Aplicación móvil funcional desarrollada en Flutter.
- API REST en Python con endpoints probados y documentados.
- Documentación técnica clara y completa para cada endpoint.
- Guía básica de consumo desde Flutter.
- Código limpio y organizado en un repositorio compartido.

Resumen de operaciones

REQUIERE AUTENTICACION	ENDPOINT	METODO HTTP	PERMISOS
No	/listadoUsuarios	GET	LECTURA
No	/editarUsuario/{id}	PATCH	ESCRITURA

La tabla anterior contiene ejemplos, deberá adaptar a su proyecto

Plantilla de documentación de Endpoints - API REST (Python).

Nombre del Endpoint

Describir brevemente la funcionalidad del endpoint.

Método HTTP

GET | POST | PUT | DELETE

Ruta

/ruta/ejemplo/{parametro}

Descripción

Explica qué hace este endpoint. Ejemplo: "Devuelve todas las tareas asignadas a un usuario."

Parámetros de entrada

A continuación, se detallan los parámetros que deben proporcionarse para establecer criterios de búsqueda específicos. Es obligatorio que todos los parámetros de entrada sean incluidos en la solicitud correspondiente.

Requerido	Argumento	Tipo	Descripción

Códigos de respuesta

Código	Descripción
201	Usuario creado exitosamente
400	Datos inválidos o campos incompletos
409	Correo o identificación ya registrados
500	Error interno del servidor

Ejemplo para registro de usuario:

Nombre del Endpoint

Registro de nuevo usuario

Método HTTP

POST

Ruta

/usuarios/registro

Descripción

Este **ENDPOINT** permite registrar un nuevo usuario en el sistema. El cliente debe enviar los datos requeridos en el cuerpo de la solicitud en formato JSON.

Argumento	Tipo	Requerido	Descripción
nombre	String	si	Se refiere al nombre del usuario.
Identificación	Integer	si	Se refiere al número documento de identidad del usuario.
correo	String	si	Correo electrónico del usuario. Debe ser una dirección válida.
fecha_nacimiento	String	si	Fecha de nacimiento del usuario en formato YYYY-MM-DD.
telefono	String	si	Número de teléfono de contacto del usuario.

La tabla anterior contiene ejemplos, deberá adaptar a su proyecto

Ejemplo de solicitud (Request)

```
POST /usuarios/registro HTTP/1.1
Host: api.midominio.com
Content-Type: application/json

{
  "nombre": "Ana Martínez",
  "identificacion": 123456789,
  "correo": "ana.martinez@email.com",
  "fecha_nacimiento": "1990-08-15",
  "telefono": "+573001112233"
}
```

Respuesta exitosa (201 Created)

```
{
  "mensaje": "Usuario registrado correctamente",
  "id_usuario": 42
}
```

Ejemplo para listado de usuarios:

Nombre del Endpoint

Listado de usuarios

Método HTTP

GET

Ruta

/usuarios

Descripción

Este **ENDPOINT** devuelve una lista de todos los usuarios registrados en el sistema. Puede incluir filtros opcionales mediante parámetros de consulta (query params), como por ejemplo por estado o rol.

Parámetros Opcionales (Query Params)

Parámetro	Tipo	Descripción
rol	String	Filtra usuarios por rol (admin, usuario, etc.)
activo	Boolean	Filtra usuarios activos (true) o inactivos

La tabla anterior contiene ejemplos, deberá adaptar a su proyecto

Ejemplo de solicitud (Request)

```
GET /usuarios?rol=usuario&activo=true HTTP/1.1
Host: api.midominio.com
```

Respuesta exitosa (201 Created)

```
[
  {
    "id": 1,
    "nombre": "Carlos Pérez",
    "correo": "carlos.perez@email.com",
    "telefono": "+573001234567",
    "activo": true
  },
  {
    "id": 2,
    "nombre": "Lucía Gómez",
    "correo": "lucia.gomez@email.com",
    "telefono": "+573004567890",
    "activo": true
  }
]
```