



# API

EDUARDO FELIPE POOT CHAIREZ

# BACKEND VS FRONTEND

- **Frontend:** es la parte de una aplicación o sitio web con la que el usuario interactúa directamente, es decir, todo lo visible en la pantalla como botones, formularios, textos, colores y animaciones, además de la experiencia de uso.
- **Backend:** se ocupa de la lógica del sistema, la validación de datos, la seguridad, la autenticación de usuarios y la comunicación con la base de datos. El backend procesa las solicitudes que llegan desde el frontend y devuelve respuestas.

# ¿QUE ES UNA API?

Es un conjunto de reglas y mecanismos que permiten que dos sistemas o aplicaciones se comuniquen entre sí. Funciona como un intermediario que recibe solicitudes, las procesa y devuelve respuestas.

# ENDPOINT

Los endpoints son las rutas o direcciones concretas a las que se hacen las solicitudes para realizar una acción determinada dentro de una API.

# HTTP (HYPERTEXT TRANSFER PROTOCOL)

Es un protocolo de comunicación que permite el intercambio de información entre un cliente y un servidor en la web. Define las reglas sobre cómo se envían las solicitudes (requests) y cómo se devuelven las respuestas (responses).

# ¿QUÉ ES UNA **API REST**?

Es un estilo de arquitectura para crear servicios web que permite que los sistemas se comuniquen entre sí usando HTTP.

Todo se maneja mediante URL y métodos HTTP:

- **GET:** obtiene información
- **POST:** envía datos para crear un nuevo recurso
- **PUT:** actualiza un recurso completo
- **PATCH:** actualiza solo una parte del recurso
- **DELETE:** elimina información

# JSON

## (JAVASCRIPT OBJECT NOTATION)

Es un formato ligero de intercambio de datos que se usa para representar información de forma estructurada y fácil de leer, tanto para personas como para programas.

# EJEMPLO

Para crear un nuevo usuario:

**POST** <https://miservidor.com/api/usuarios>

Se envia al servidor:

```
{  
  "nombre": "Carlos Ramírez",  
  "email": "carlos@gmail.com",  
  "password": "123456",  
  "rol": "Alumno"  
}
```



Para devolver los usuarios registrados:

**GET** <https://miservidor.com/api/usuarios>

Se recibe del servidor:

```
[  
  {  
    "id": 1,  
    "nombre": "Carlos Ramírez",  
    "email": "carlos@gmail.com",  
    "rol": "Alumno",  
  }  
]
```

# CÓDIGOS DE ESTADO HTTP

- 200 - **Éxito**
- 400 - **Error del cliente**
  - 404 - **Not Found**
- 500 - **Error del servidor**