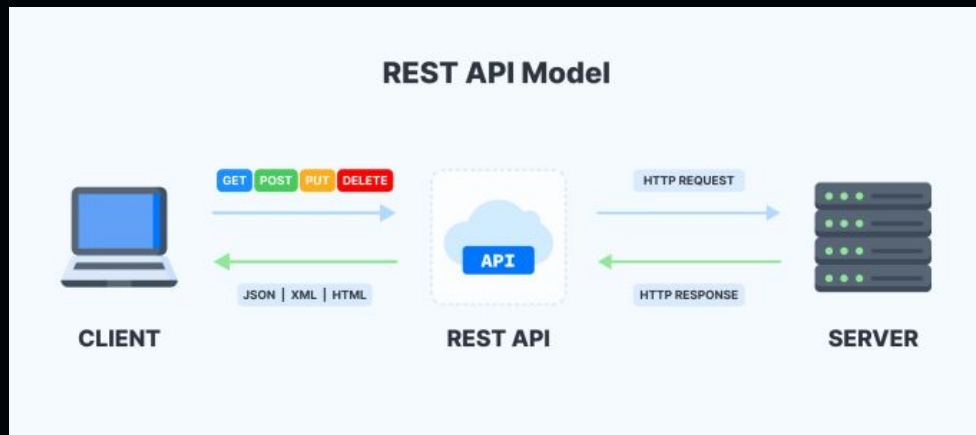


Clase N° 12: NextJS - API



API REST

Una API REST (Representational State Transfer) es una interfaz que permite la comunicación entre diferentes sistemas utilizando el protocolo HTTP. Sigue principios específicos para garantizar que las interacciones sean simples, escalables y estandarizadas.

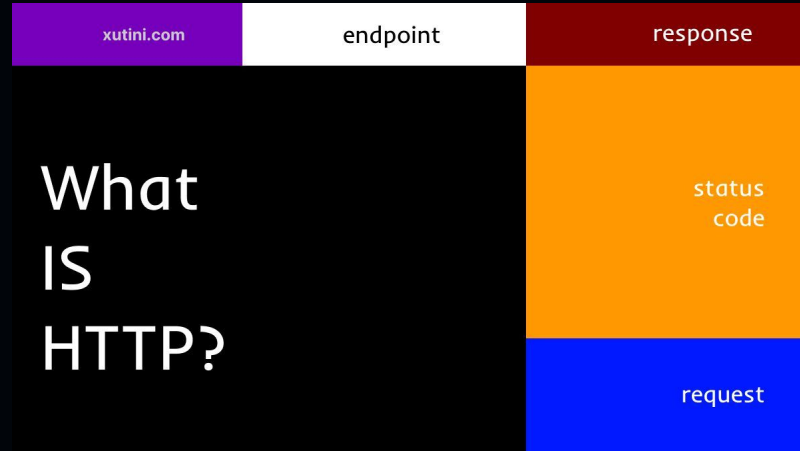


Protocolo de comunicación HTTP

Funciona como un conjunto de reglas que permite a los navegadores y servidores intercambiar información. El cliente (navegador) envía una solicitud HTTP al servidor para obtener datos, y el servidor responde con una respuesta HTTP que incluye el código de estado y los datos solicitados

Todas las solicitudes se componen de los siguientes datos:

- Host
- Endpoint
- Request
- Response
- Status Code



API y Next.JS

- **Next** permite que tanto el código del frontend como el de la API estén en el mismo proyecto, lo que facilita el despliegue. (arquitectura monolítica).
- Las API se definen como funciones del lado del servidor ubicadas dentro de la carpeta **app/api**.

Usos comunes

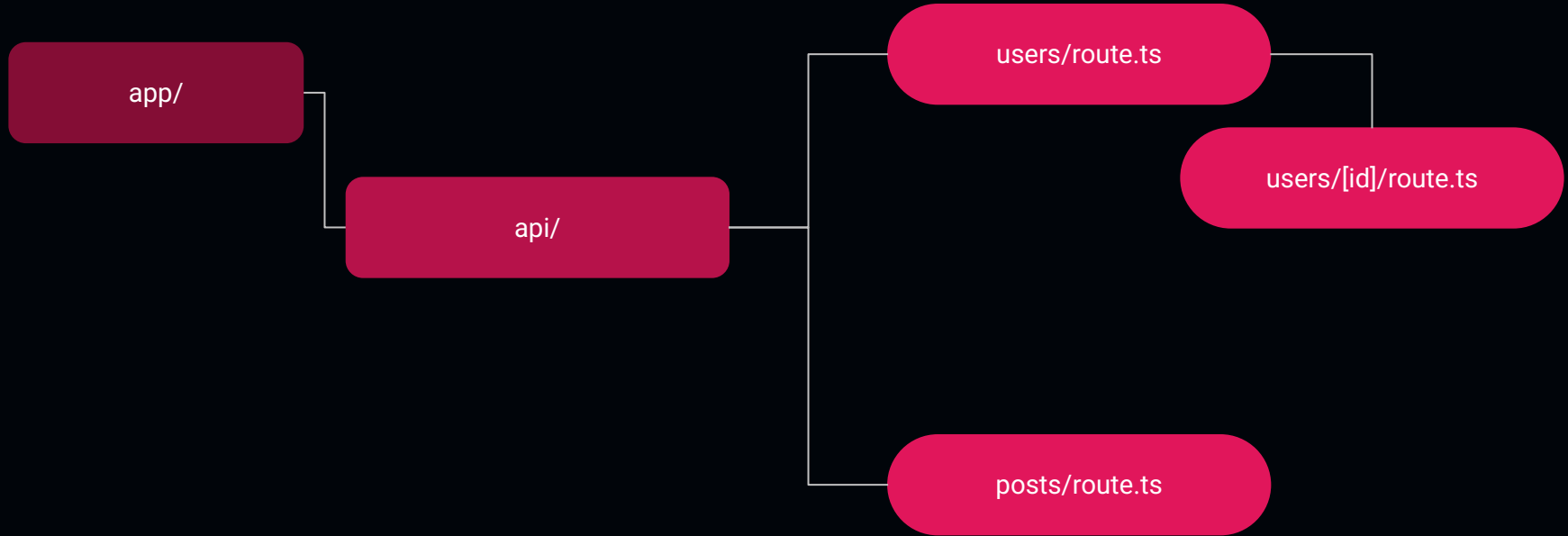
- GET: Obtener datos (ej., lista de usuarios).
- POST: Crear nuevos recursos (ej., registrar un usuario).
- PUT: Actualizar un recurso completo (ej., modificar un perfil).
- PATCH: Actualizar un recurso parcialmente.
- DELETE: Eliminar un recurso (ej., borrar un usuario).

Manejo de Rutas API Dinámicas

- Las rutas dinámicas permiten manejar datos específicos basados en parámetros incluidos en la URL.
- Seguimos usando la convención de [param]
- `app/api/users/[id]/route.ts`

Ejemplos

→ La estructura de carpetas y nomenclatura de archivos es:



Ejemplos

→ Vamos al código para ver un ejemplo entre todos



Momento de práctica:

- Crear una API con las rutas básicas para manejar productos:
 - ◆ GET
 - ◆ POST
 - ◆ PUT
 - ◆ DELETE
- Implementar la lógica dentro de `app/api/products/route.js`
- Enviar respuestas JSON con mensajes apropiados para cada caso.
- Validar el método HTTP utilizado.