

DENO:

ctrl + may + P

> Dena:

deno init → en terminal

API rest → programa que tiene end points y que atiende a peticiones que le llegan a través del protocolo http.

End points → ruta `www.nebrija.es/alumno` → ruta

ctrl + ñ → abrir terminal

método (PUT, GET, POST, DELETE) → cómo hace la petición

payload: contenido de la petición

función recibe un parámetro de tipo request con toda la información sobre la información que me llega, y me va a dar una respuesta.

```
const handler = (req: Request): Response => {  
  return new Response("Hola mundo");  
}
```

en el deno.json (para tener todos los permisos)

Deno.serve({ port: 3000 }, handler); → función que gestiona todas las peticiones

deno task dev → en terminal para lanzar el código.

const url = new URL(req.url);
sacamos la url ✓ Dentro del handler

console.log('La url es: ' + url.host); → Para saber la ruta.

return new Response(url.host);

La ruta no está incluida en el host. Para ello añado:

const path = url.pathname;

return new Response('' + url.host + '' + path);

Como devolver una cosa u otra en función de la ruta.

```
const users = [{ name: "Alberto", email: "a@gmail.com" },  
  { name: "Luis", email: "l@gmail.com" }];
```

```
const handler = (req: Request): Response => {  
  const url = new URL(req.url);  
  const path = url.pathname;  
  if (path === "/users") {  
    return new Response("Estas llamando a /users");  
  } else if (path === "/dates") {  
    return new Response("Estas llamando a dates");  
  }  
  
  return new Response("La ruta " + path + " no existe", { status: 404 });  
};  
Deno.serve({ port: 3000 }, handler);
```

return new Response(JSON.stringify(users));

↳ no se puede poner aquí directamente users porque las respuestas tienen que ser texto, no un objeto. Empleo `JSON.stringify` para que me lo convierta en texto.

0.0.0:3000/users?name=Alberto → parámetro de búsqueda

const searchParams = url.searchParams

```
const handler = (req: Request): Response => {  
  const url = new URL(req.url);  
  const path = url.pathname;  
  const searchParams = url.searchParams;  
  if (path === "/users") {  
    if (searchParams.get("name")) {  
      const name = searchParams.get("name");  
      const u = users.filter((elem) => elem.name === name);  
      return new Response(JSON.stringify(u));  
    }  
  
    return new Response(JSON.stringify(users));  
  }  
  
  return new Response("La ruta " + path + " no existe", { status: 404 });  
};  
Deno.serve({ port: 3000 }, handler);
```

→ Si busco Alberto, me salen los datos de Alberto, pero si busco a Ana que no está, me devuelve un array vacío. Si hubiera dos Albertos, te mostraría ambos.

③ Devolver solo el usuario Alberto y solo Luis.

④ Tener tres posibilidades, que filtre por nombre, por edad o por ambas.