

Endpoint.

Un endpoint es básicamente una URL específica de un servidor que expone algún recurso o funcionalidad.

Es el “punto de entrada” donde los clientes (como navegadores o aplicaciones móviles) pueden acceder a datos o servicios.

Ejemplos:

`https://api.ejemplo.com/usuarios` → Devuelve la lista de usuarios.

`https://api.ejemplo.com/usuarios/123` → Devuelve la información del usuario con id 123.

Características:

- Cada endpoint representa un recurso o acción específica.
- Generalmente se asocia a un método HTTP: GET, POST, PUT, DELETE.

Request.

Un request es la petición que hace el cliente al endpoint.

Contiene información como:

- La URL del endpoint.
- El método HTTP (GET, POST, etc.).
- Headers (información extra como tipo de contenido o autorización).
- Body (datos que se envían, típicamente en POST o PUT).

Ejemplo en HTTP:

POST `https://api.ejemplo.com/usuarios`
Content-Type: `application/json`

```
{  
  "nombre": "Ana",  
  "edad": 28  
}
```

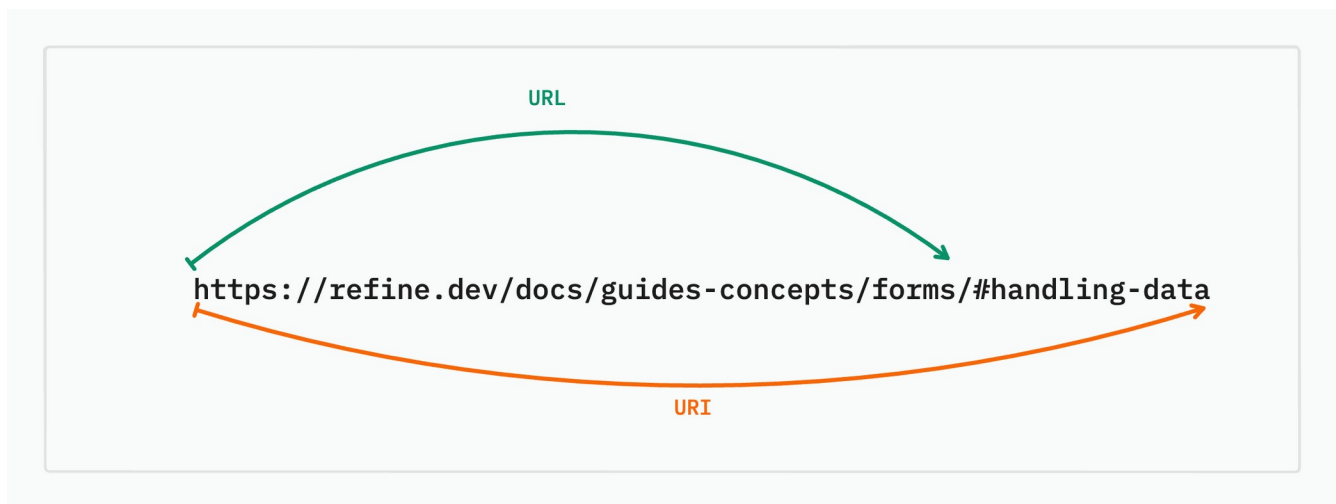
Aquí, el request está intentando crear un nuevo usuario en el endpoint `/usuarios`.

Diferencias clave.

Concepto	Qué es	Función
Endpoint	URL en el servidor	Define dónde se accede a un recurso o servicio
Request	Petición del cliente	Es lo que envías al endpoint para interactuar con él

URI (Uniform Resource Identifier).

Aspecto	URL	URI
Definición	Un tipo de URI que identifica un recurso y también proporciona un medio para localizarlo.	Una cadena de caracteres para identificar un nombre o un recurso en Internet.
Objetivo	Se utiliza principalmente para localizar recursos en la web.	Se utiliza para identificar un recurso, que puede ser una URL (que localiza el recurso) o una URN (que nombra el recurso).
Componentes	Incluye principalmente un protocolo (como HTTP, HTTPS), un dominio y posiblemente una ruta y parámetros de consulta.	Puede ser una URL, una URN o una combinación, y no necesariamente tiene que especificar cómo localizar el recurso.
Ejemplo	<code>https://www.mywebsite.com/page</code>	<code>https://www.mywebsite.com/page</code> (como URL) o <code>urn:isbn:0451450523</code> (como URN)
Uso	Se utiliza directamente en los navegadores web para acceder a páginas web.	Se utiliza en varios sistemas (como espacios de nombres XML, recursos RDF) para identificación, no siempre para recuperación.
Especificaciones del protocolo	Debe especificar un protocolo (HTTP, FTP, etc.) para acceder al recurso.	No es necesaria la especificación del protocolo.
Recuperación de recursos	especifica la capacidad de recuperar el recurso que identifica.	Puede o no implicar la recuperabilidad del recurso.
Normalización	Un tipo específico de URI según los estándares definidos por el Grupo de trabajo de ingeniería de Internet (IETF).	Un formato generalizado estandarizado por la IETF, que cubre más posibilidades que solo las URL.
Mutabilidad	Generalmente mutable y puede cambiar con el tiempo.	Puede ser mutable o inmutable, dependiendo de si es una URL o una URN.
Semejanza	Todas las URL son URI.	Incluye URL como un subconjunto.



Path.

El path es la ruta dentro del servidor que apunta al recurso exacto.

Es la parte de la URL que sigue al dominio y antes de los parámetros (?).

No incluye protocolo ni dominio.

Ejemplo:

URL completa: `https://ejemplo.com/usuarios/123`

Path: `/usuarios/123`

URL completa: `https://ejemplo.com:443/usuarios/123?activo=true`

URI: `https://ejemplo.com/usuarios/123` (identifica el recurso)

URL: `https://ejemplo.com/usuarios/123` (dónde y cómo acceder)

Path: `/usuarios/123` (ruta del recurso en el servidor)

Test Drive Development:

Es una técnica de desarrollo de software donde las pruebas unitarias se escriben antes que el código de producción, siguiendo un ciclo de tres pasos: Rojo (escribir una prueba que falle), Verde (escribir el código mínimo para que pase), y Refactorizar (mejorar la estructura del código sin cambiar su funcionalidad).