



MÉTODOS HTTP

Los métodos HTTP son verbos utilizados para indicar la acción que se desea realizar en un recurso específico en la web. Aquí tienes un resumen de los métodos HTTP más comunes:

GET:

- **Características:**
 - No tiene cuerpo en la solicitud.
 - Es seguro y no idempotente.
 - Recupera datos del servidor.
- **Uso:** Obtener información de un recurso.
- **Ejemplo:** Obtener información de un usuario

```
http
```

[Copiar código](#)

```
GET /usuarios/123
```

POST:

- **Características:**
 - Tiene cuerpo en la solicitud.
 - No es seguro y no idempotente.
 - Envía datos al servidor, comúnmente para crear un nuevo recurso.
- **Uso:** Enviar datos para crear un nuevo recurso.
- **Ejemplo:** Crear un nuevo usuario.



```
http Copiar código

POST /usuarios
Content-Type: application/json

{
  "nombre": "Juan Pérez",
  "email": "juan@example.com"
}
```

PUT:

- **Características:**
 - Tiene cuerpo en la solicitud.
 - No es seguro pero es idempotente.
 - Actualiza completamente un recurso existente.
- **Uso:** Actualizar un recurso existente.
- **Ejemplo:** Actualizar la información de un usuario.

```
http Copiar código

PUT /usuarios/123
Content-Type: application/json

{
  "nombre": "Juan Pérez",
  "email": "juan.perez@example.com"
}
```

PATCH:

- **Características:**
 - Tiene cuerpo en la solicitud.
 - No es seguro pero es idempotente.



- Realiza actualizaciones parciales en un recurso existente.
- **Uso:** Actualizar parcialmente un recurso.
- **Ejemplo:** Actualizar solo el correo electrónico de un usuario.

```
http                                                                    Copiar código

PATCH /usuarios/123
Content-Type: application/json

{
  "email": "juan.perez@example.com"
}
```

DELETE:

- **Características:**
 - Puede tener o no cuerpo en la solicitud.
 - No es seguro pero es idempotente.
 - Elimina un recurso.
- **Uso:** Eliminar un recurso existente.
- **Ejemplo:** Eliminar un usuario

```
http                                                                    Copiar código

DELETE /usuarios/123
```

HEAD:

- **Características:**
 - No tiene cuerpo en la solicitud ni en la respuesta.
 - Es seguro e idempotente.
 - Similar a GET, pero solo obtiene los encabezados de la respuesta.
- **Uso:** Verificar si un recurso existe y obtener metadatos sin descargar el recurso.



- **Ejemplo:** Obtener los metadatos de un usuario

```
http
```

[Copiar código](#)

```
HEAD /usuarios/123
```

OPTIONS:

- **Características:**
 - Puede tener o no cuerpo en la solicitud.
 - Es seguro e idempotente.
 - Describe las opciones de comunicación para el recurso.
- **Uso:** Obtener métodos permitidos y otras opciones de comunicación.
- **Ejemplo:** Obtener métodos permitidos para un recurso de usuarios.

```
http
```

[Copiar código](#)

```
OPTIONS /usuarios
```