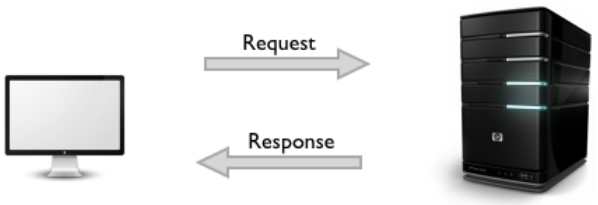
ACCESO A DATOS
UD06 - ANEXO 1: DISEÑO DE APIs: PROTOCOLO HTTP

1. MENSAJES HTTP

En el protocolo HTTP existen 2 tipos de mensajes:

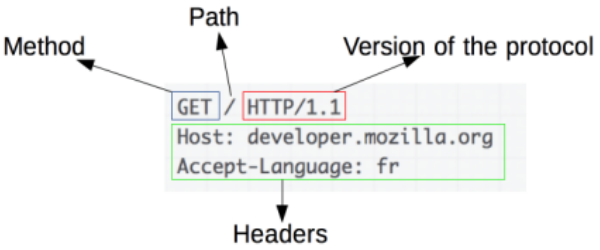
- **Peticiones HTTP (Requests):** Son los mensajes que envía el cliente al servidor para solicitar información o realizar una operación
- **Respuestas HTTP (Responses):** Son las respuestas que el servidor envía al cliente para responder a sus peticiones



1.1. Petición HTTP

Toda petición HTTP está compuesta de:

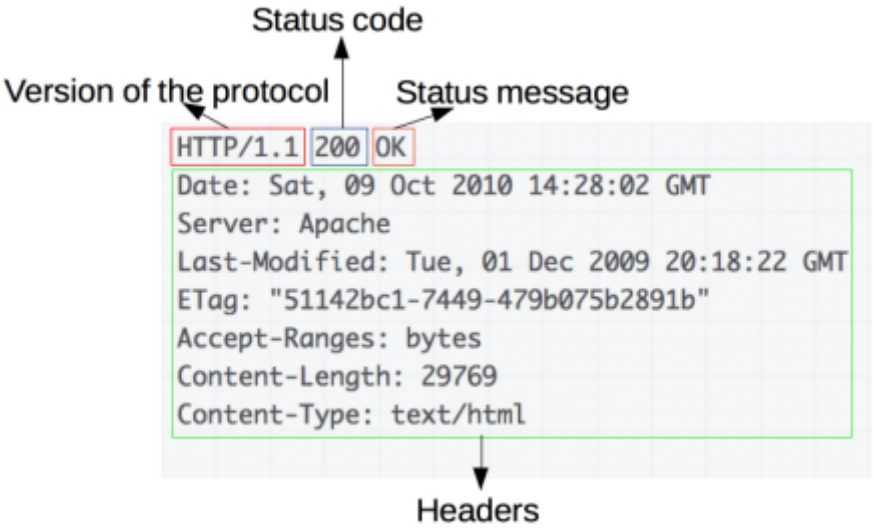
- **Método HTTP:** el tipo de operación que se quiere realizar (registro, consulta, modificación...).
- **Ruta o path:** la ruta completa para identificar el recurso sobre el que se quiere operar.
- **Cabeceras:** información adicional para el servidor que procesa la petición.
- **Cuerpo:** la información que se quiere enviar junto con la petición (sin contenido en operaciones GET). En una operación de registro, sería la información que se almacena.



1.2. Respuesta HTTP

Toda petición HTTP conlleva una respuesta, ya sea por haberse ejecutado ésta correctamente o bien para informar de un error. Estará compuesta de:

- **Código de estado:** indica si la petición tuvo o no éxito y por qué.
- **Mensaje de estado:** una descripción breve sobre el código de estado.
- **Cabeceras:** información adicional para el cliente que realizó la petición.
- **Cuerpo:** la información que se quiere enviar al cliente. En una operación de consulta serían los datos que se solicitaron en la petición.



2. MÉTODOS HTTP

El protocolo HTTP define una serie de métodos que permiten indicar el tipo de operación que se quiere realizar en toda petición HTTP.

La fundación Mozilla mantiene una documentación muy buena sobre todos los métodos que existen en el protocolo.

- Documentación métodos HTTP en mozilla.org

Como durante el curso no veremos todos ellos, dejaremos en este anexo un resumen de lo más importante acerca de los métodos con los que vamos a trabajar.

2.1. Métodos HTTP más utilizados

En el protocolo HTTP, el método define el tipo de operación que se va a realizar.

De entre los métodos existentes en el protocolo, los más utilizados son los que se corresponden con las operaciones CRUD:

Método	Cuándo usar	Ejemplo
GET	Operación de consulta	GET /books o GET /books/2
POST	Operación de registro o acciones	POST /books
DELETE	Operación de borrado	DELETE /books/12
PUT	Operación de modificación completa	PUT /books/12
PATCH	Operación de modificación parcial	PATCH /books/12

Cabe destacar el uso del método POST para todas aquellas operaciones que no se pueden clasificar como una operación CRUD de alta, baja, modificación o consulta de recurso. Por ejemplo, cancelar un préstamo, devolver un pedido, desactivar una tarjeta de crédito…

2.2. Modelado de Recursos/URIs

Las URIs de cada recurso serán cada uno de los métodos que desarrollamos en el proyecto de nuestra API y mapeamos usando las anotaciones @GetMapping, @PostMapping, @DeleteMapping, @PatchMapping o @PutMapping.

Con el objetivo de definir una colección uniforme de operaciones o endpoints, es muy importante perder algo de tiempo para pensar y diseñar cómo deben de ser estas URLs.

El primer paso es tener en cuenta que el método HTTP seleccionado actuará como verbo y que la URL deberá definir al recurso sobre el que ese 'verbo' actúa (luego veremos que no todas las operaciones encajan en este modelo).

A continuación se muestran algunas URLs a modo de ejemplo:

- **Operaciones de consulta:**
  - GET /products
  - GET /products?category=food
  - GET /products?category=food&price=100
  - GET /product/{productId}
  - GET /user/{userId}/orders
  - GET /user/{userId}/order/{orderId}
- **Operaciones de registro:**
  - POST /products
  - POST /user/{userId}/orders
- **Operaciones de borrado:**
  - DELETE /product/{productId}
  - DELETE /user/{userId}/order/{orderId}
- **Operaciones de modificación (registro completo):**
  - PUT /product/{productId}
  - PUT /user/{userId}/order/{orderId}
- **Operaciones de modificación parcial:**
  - PATCH /product/{productId}
  - PATCH /user/{userId}/order/{orderId}

Por ejemplo:

Task	Method	Path
Create a new task	POST	/tasks
Delete an existing task	DELETE	/tasks/{id}
Get a specific task	GET	/tasks/{id}
Search for tasks	GET	/tasks
Update an existing task	PUT	/tasks/{id}

3. CÓDIGOS DE ESTADO

El protocolo HTTP define una serie de códigos de estado que permiten indicar el resultado de toda petición HTTP. Estos códigos proporcionan al cliente que realizó la petición información global sobre el resultado que se devuelve en la respuesta.

La fundación Mozilla mantiene una documentación muy buena sobre todos estos códigos:

- Códigos de estado HTTP

Como durante el curso no veremos todos ellos, dejaremos en este anexo un resumen de lo más importante acerca de los códigos de estado con los que vamos a trabajar.

Existen 5 categorías:

- **1XX Información:** proporcionan información a nivel de protocolo.
- **2XX Éxito:** se utilizan para indicar al cliente que la petición se ha realizado con éxito.
- **3XX Redirección:** sirven para indicar al cliente que debe realizar alguna acción adicional para completar la petición.
- **4XX Error en el cliente:** se utilizan para indicarle al cliente que ha cometido algún tipo de error al realizar la petición.
- **5XX Error en el servidor:** indican que se ha producido algún tipo de error en el servidor mientras se ejecutaba la petición.

3.1. Códigos de estado HTTP más utilizados

Código	Estado	Cuándo usar
200	**OK**	Ok en GET,PUT,PATCH
201	**Created**	Ok en POST (registro)
204	**Not Content**	Ok sin devolver datos en DELETE o GET si no devuelve nada
400	**Bad Request**	Error de invocación del cliente: argumentos no válidos…
401	**Unauthorized**	Usuario/Contraseña incorrectos
403	**Forbidden**	Sin privilegios
404	**Not Found**	Recurso no encontrado (especificado en la entrada)
500	**Internal Server Error**	Error en backend

4. EJERCICIOS RESUELTOS

1. ¿Qué código de estado HTTP devolverá una operación que devuelve la lista de usuarios conectados cuando no haya ninguno? ¿Y qué información devolverá en la respuesta?

Si no hay usuarios conectados en la lista, una buena opción sería devolver un código de estado HTTP 204 (No Content). Este código de estado indica que la solicitud se ha completado correctamente, pero que no hay ningún contenido que devolver.

En la respuesta, puede incluir información adicional en el cuerpo de la respuesta, como un mensaje que indique que no hay usuarios conectados. También puede incluir información adicional en el encabezado de la respuesta, como información de paginación o enlaces a recursos relacionados.

Aquí hay un ejemplo de cómo podría verse la respuesta:

```
HTTP/1.1 204 No Content
Content-Type: application/json

{
  "message": "No hay usuarios conectados"
}
```

2. ¿Qué código de estado HTTP debe devolver una operación que registra un nuevo producto en la base de datos? ¿Debe devolver algo como respuesta?

Un código de estado HTTP adecuado para una operación que registra un nuevo producto en la base de datos sería 201 (Created). Este código de estado indica que la solicitud ha sido procesada correctamente y ha resultado en la creación de un nuevo recurso.

En la respuesta, es común incluir información sobre el recurso creado. Por ejemplo, puede incluir el ID del nuevo producto, el nombre del producto, el precio, etc. También puede incluir enlaces a recursos relacionados, como un enlace al recurso del producto creado.

Aquí hay un ejemplo de cómo podría verse la respuesta:

```
HTTP/1.1 201 Created
Content-Type: application/json
Location: /products/123

{
  "id": 123,
  "name": "Product Name",
  "price": 99.99
}
```

3. ¿Qué código de estado HTTP y respuesta devolverá una operación que utiliza el método HTTP DELETE?

Cuando se utiliza el método HTTP DELETE, una buena opción es devolver un código de estado HTTP 204 (No Content). Este código de estado indica que la solicitud se ha procesado correctamente y que el recurso se ha eliminado correctamente, pero no hay ningún contenido que devolver.

En la respuesta, no es necesario incluir ningún contenido en el cuerpo de la respuesta, ya que se ha eliminado el recurso. También puede incluir información adicional en el encabezado de la respuesta, como enlaces a recursos relacionados.

Aquí hay un ejemplo de cómo podría verse la respuesta:

```
HTTP/1.1 204 No Content
Content-Type: application/json
```

4. ¿Qué código de respuesta HTTP devolverá una operación que sirve para conocer la información de un producto determinado?

Si la operación es capaz de encontrar la información del producto solicitado y devolverla correctamente, un código de estado HTTP adecuado sería 200 (OK). Este código de estado indica que la solicitud se ha completado correctamente y que se ha proporcionado el contenido solicitado en el cuerpo de la respuesta.

En la respuesta, puede incluir información sobre el producto, como su ID, nombre, precio, etc. También puede incluir enlaces a recursos relacionados, como enlaces a imágenes del producto o reseñas del mismo.

Aquí hay un ejemplo de cómo podría verse la respuesta:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "id": 123,
  "name": "Product Name",
  "price": 99.99
}
```

5. ¿Qué código de respuesta HTTP devolverá una operación que sirve para conocer la información de un producto determinado si no lo encuentra?

Si no se encuentra el producto solicitado, una buena opción es devolver un código de estado HTTP 404 (Not Found). Este código de estado indica que el recurso solicitado no se encuentra en el servidor.

En la respuesta, puede incluir información adicional en el cuerpo de la respuesta, como un mensaje que indique que el producto no se ha encontrado. También puede incluir información adicional en el encabezado de la respuesta, como enlaces a recursos relacionados o información de paginación.

Aquí hay un ejemplo de cómo podría verse la respuesta:

```
HTTP/1.1 404 Not Found
Content-Type: application/json

{
  "message": "El producto no se ha encontrado"
}
```

5. BIBLIOGRAFÍA

- https://datos.codeandcoke.com/apuntes:http
- https://code.tutsplus.com/es/tutorials/http-the-protocol-every-web-developer-must-know-part-1--net-31177
- https://ed.team/comunidad/codigos-de-respuesta-http