

# Estructurar e implementar el contrato exterior

---

## Coming Up



- **Estructuración del contrato exterior**
- **Directrices de diseño de URI**
- **Enrutamiento**
- **La importancia de los códigos de estado**
- **Negociación de contenidos**
- **Modelo exterior frente al modelo de entidad**

# Estructurando nuestro contrato de cara al exterior



Identificador de  
recursos

<http://host/api/authors>



Método HTTP

<https://datatracker.ietf.org/doc/html/rfc9110>



Recurso

(representación: tipos  
de medios)

# Estructurando nuestro contrato de cara al exterior



Identificador de  
recursos

<http://host/api/authors>



Método HTTP

<https://datatracker.ietf.org/doc/html/rfc9110>



Carga

(representación: tipos  
de medios)

# Guía para el nombrado de recursos



Sustantivos: cosas, no acciones

- ~~api/getauthors~~
- GET api/authors
- GET api/authors/{authorId}

Transmitir el significado al elegir  
sustantivos

# Guía para el nombrado de recursos



ID SIEMPRE AL FINAL

Cumplir con este principio para que sea predecible

- ~~[api/something/somethingelse/employees](#)~~
- [api/employees](#)
- ~~[api/id/employees](#)~~
- [api/employees/{employeeId}](#)

# Guía para el nombrado de recursos



Representar la jerarquía al nombrar los recursos

- [api/authors/{authorId}/books](#)
- [api/authors/{authorId}/books/{bookId}](#)

# Guía para el nombrado de recursos



Los filtros, las clasificaciones, .... no son recursos

- ~~[api/authors/orderby/name](#)~~
- [api/authors?orderby=name](#)



# Guía para el nombrado de recursos

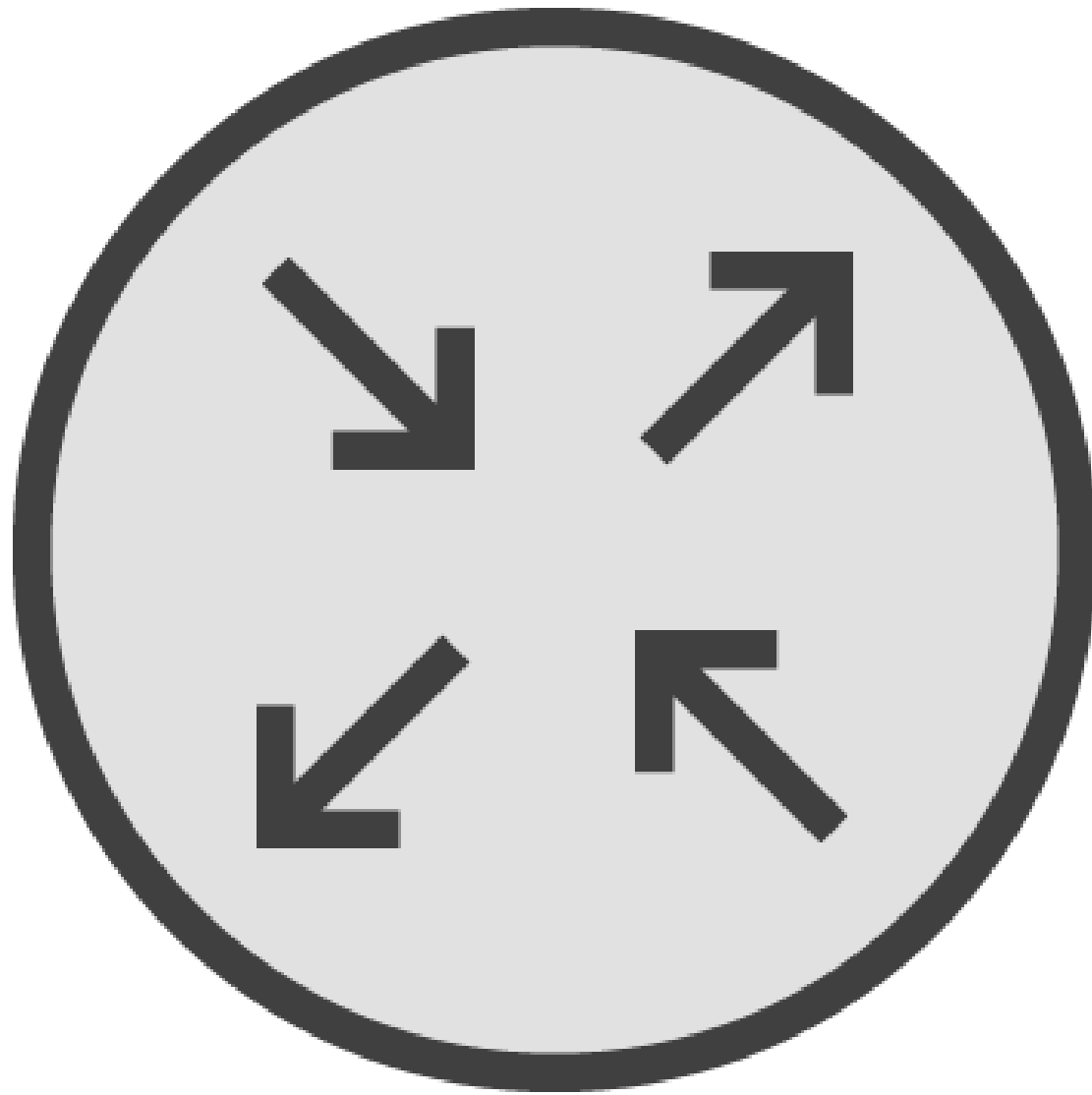


A veces, las llamadas de tipo RPC no se asignan fácilmente a nombres de recursos en plural.

~~api/authors/{authorId}/pagetotals~~

- ~~api/authorpagetotals/{id}~~
- api/authors/{authorId}/totalamountofpages

# Trabajando con el enrutamiento



El enrutamiento hace coincidir una URI de petición con una acción en un controlador

```
// set up the request pipeline
```

```
...
```

```
app.MapControllers();
```

```
...
```

## Enrutamiento a puntos de conexión

**Añadir puntos de conexión para acciones del controlador al `EndpointRouteBuilder` sin especificar ninguna ruta**

**Especificar rutas mediante atributos de ruta**

# Contrato de cara al exterior



Identificador de  
recursos

`http://host/api/authors`



Método HTTP

<https://datatracker.ietf.org/doc/html/rfc9110>



Carga

(representación: tipos  
de medios)

# Interactuando con Recursos mediante Métodos HTTP

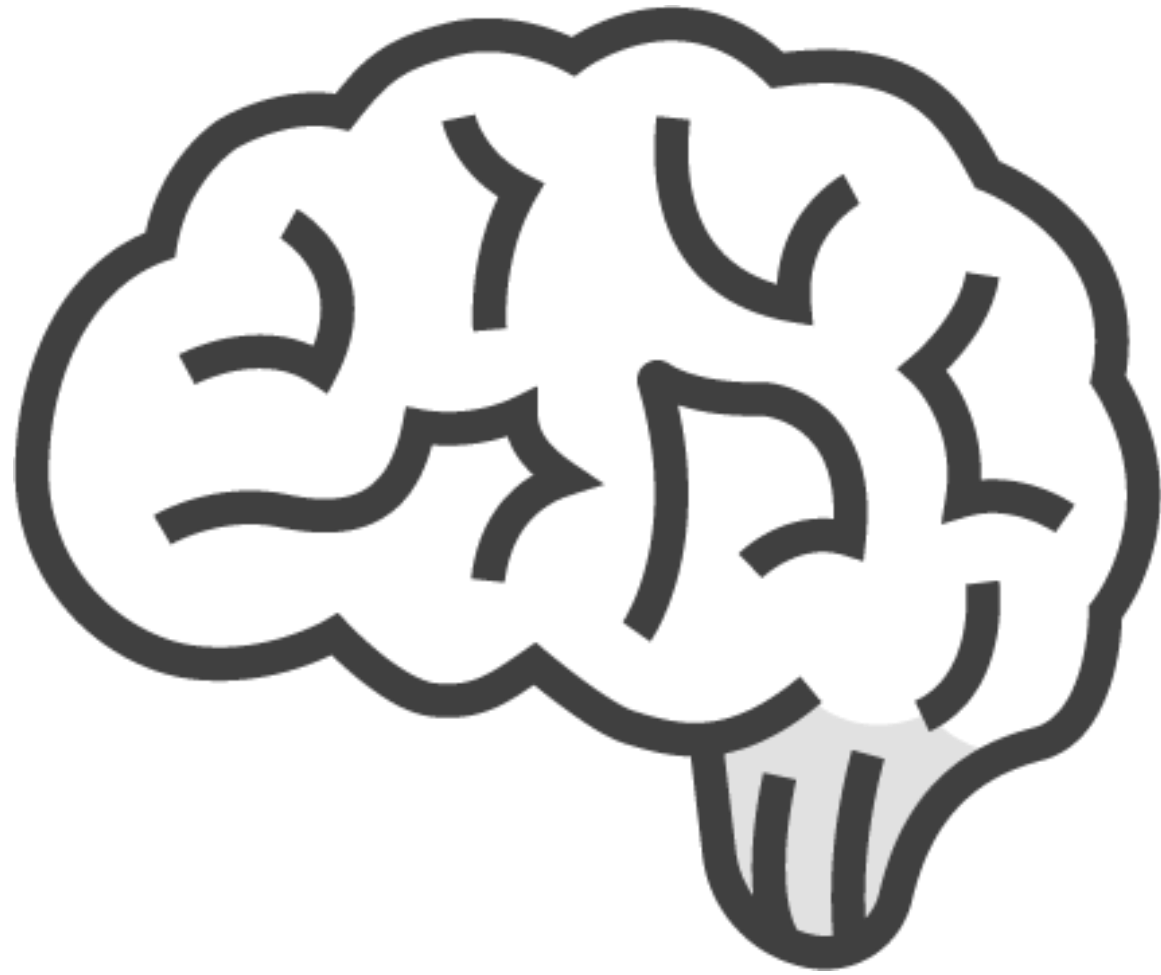
Método HTTP	Carga de la petición	URI Ejemplo	Carga de la respuesta
GET	-	/api/authors /api/authors/{authorId}	colección de autores un autor
POST	un autor	/api/authors	un autor
PUT	un autor	/api/authors/{authorId}	un autor o vacío
PATCH	JsonPatchDocument sobre autor	/api/authors/{authorId}	un autor o vacío
DELETE	-	/api/authors/{authorId}	-
HEAD	-	/api/authors /api/authors/{authorId}	-
OPTIONS	-	/api/...	-

# Demo



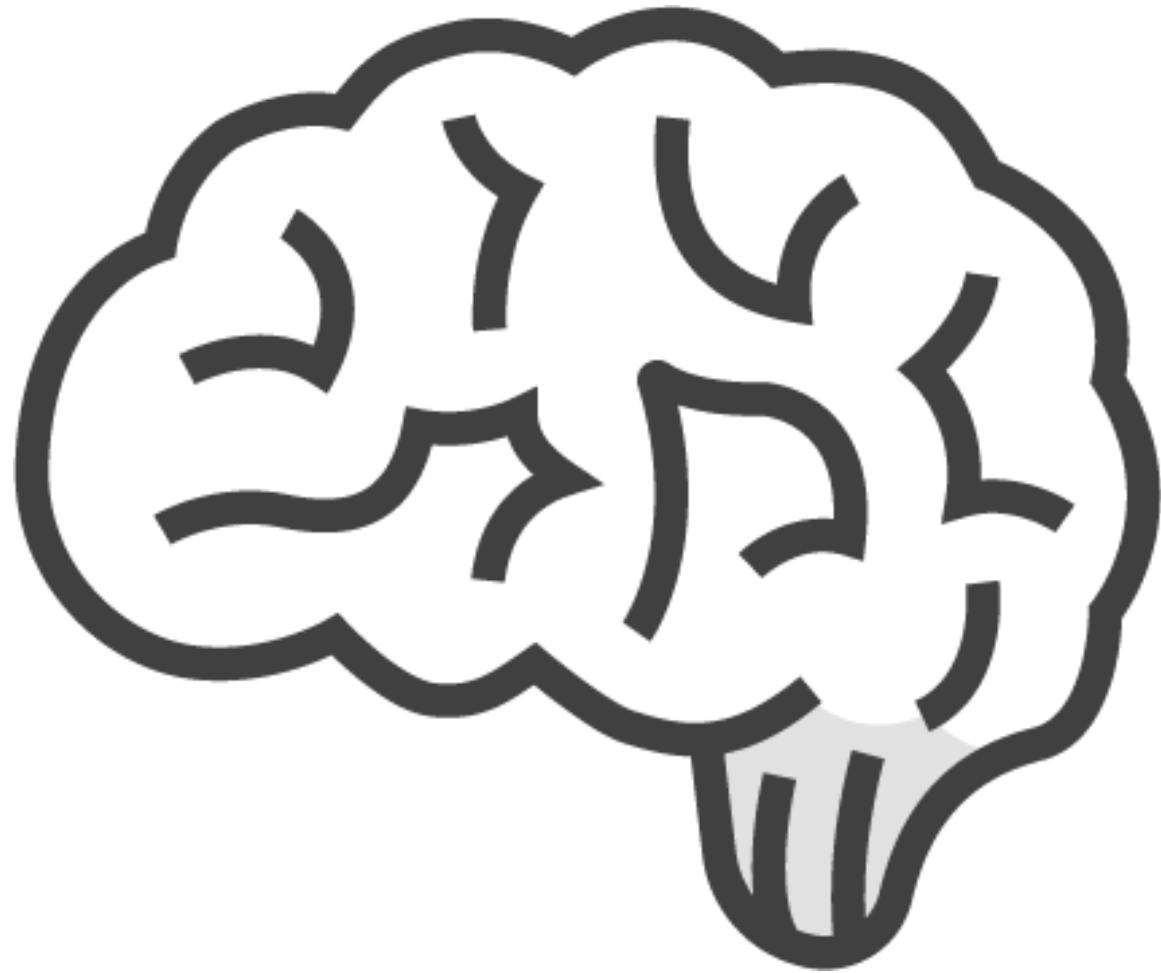
**Cumplimiento de las directrices relativas a la URI**

# La importancia de los códigos de estado



- Los códigos de estado le informan al cliente de la API...
  - Si la solicitud funcionó o no tal como se esperaba
  - Cual ha sido el motivo por el que se hay un fallo en la solicitud

# La importancia de los códigos de estado



- Sé lo más específico posible
  - Los consumidores de APIs no suelen ser humanos.
- Sé especialmente específico a la hora de informar sobre quién o qué es responsable de un error



# La importancia de los códigos de estado

No se suele utilizar

**Nivel 100**  
**Informativo**

200 – Ok  
201 – Created  
204 – No content

**Nivel 200**  
**Correcto**

No se suele utilizar

**Nivel 300**  
**Redirección**

# La importancia de los códigos de estado

**400 – Bad request**

**401 – Unauthorized**

**403 – Forbidden**

**404 – Not found**

**405 – Method not allowed**

**406 – Not acceptable**

**409 - Conflict**

**415 – Unsupported media  
type**

**422 – Unprocessable  
entity**

**500 – Internal server error**

**Nivel 400**  
**Errores de cliente**

**Nivel 500**  
**Errores de servidor**

# Demo



**Devolución de códigos de estado correctos**

# Errores frente a fallos

## Errores

El consumidor pasa datos no válidos a la API, y la API los rechaza correctamente.

Códigos de estado de nivel 400

No cambian la disponibilidad de la API

## Fallos

La API no devuelve una respuesta a una solicitud válida

Códigos de estado de nivel 500

Cambian la disponibilidad de la API

# Demo



**Gestión de fallos y evitar la exposición de detalles de implementación**

# Diseño del contrato externo



Identificador de  
recursos

`http://host/api/authors`



Método HTTP

`https://datatracker.ietf  
.org/doc/html/rfc9110`

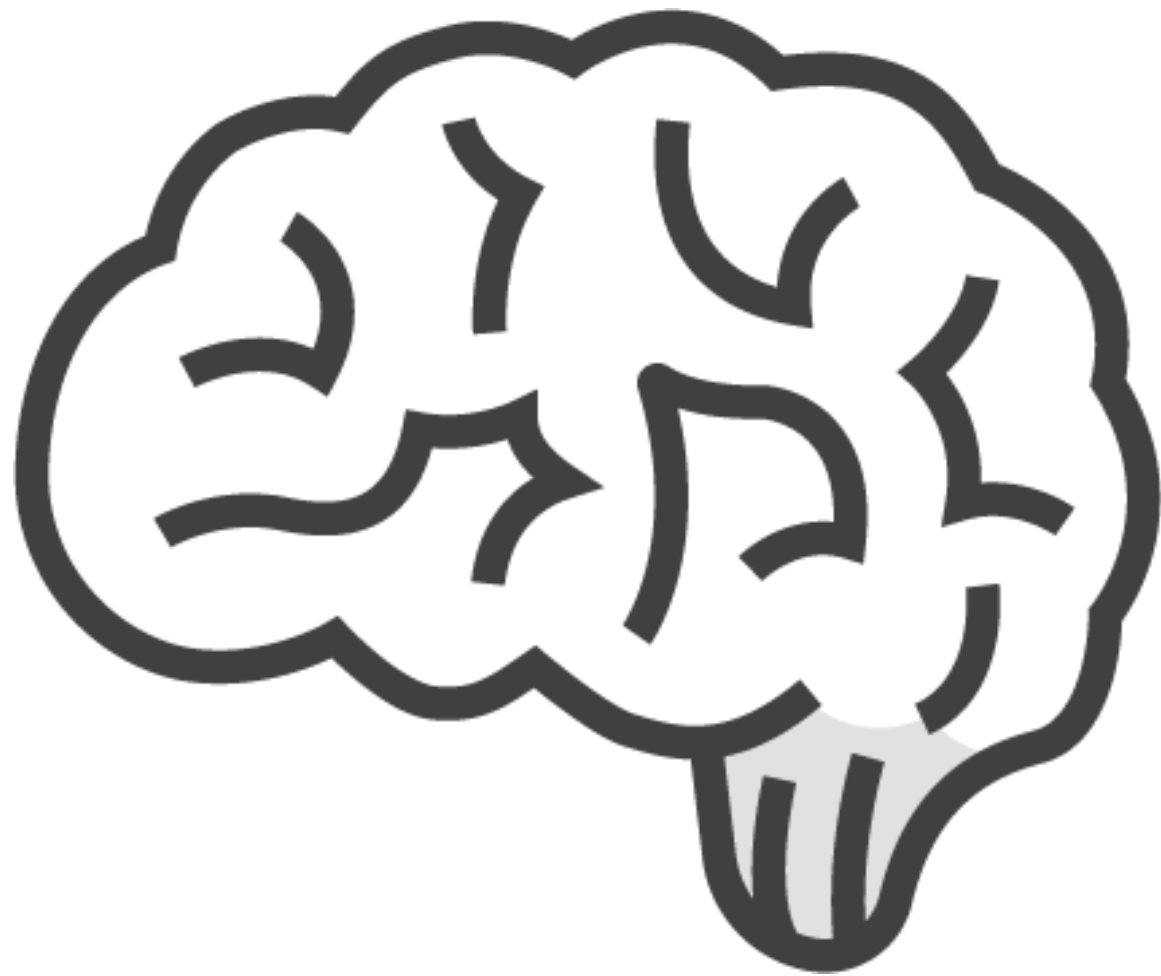


**Carga**  
**(representación:**  
**tipos de medios)**

# Negociación de contenido

**El proceso de selección de la mejor representación para una respuesta dada cuando hay múltiples representaciones disponibles**

# Formatos y negociación de contenidos



- El tipo de medio se pasa a través del encabezado Accept (Aceptar) de la solicitud
  - aplicación/json
  - aplicación/xml
  - ...



# Trabajar con negociación de contenidos y formateadores



Es aceptable devolver una representación en un formato predeterminado cuando no se incluye el encabezado Accept.



No es aceptable devolver una representación en un formato predeterminado cuando el tipo de medio solicitado no está disponible.  
Devolver: **406 - No aceptable**

# Formatos y negociación de contenidos



Formateador de  
salida

Se ocupa de la salida

Tipo de medio: accept header



Formateador de  
entrada

Se ocupa de la entrada

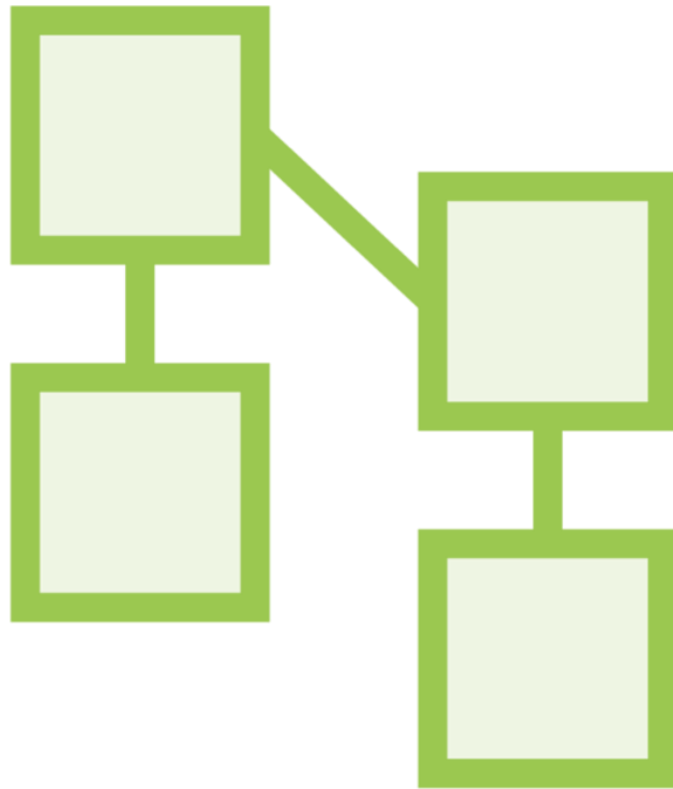
Tipo de medio: content-type  
header

# Demo

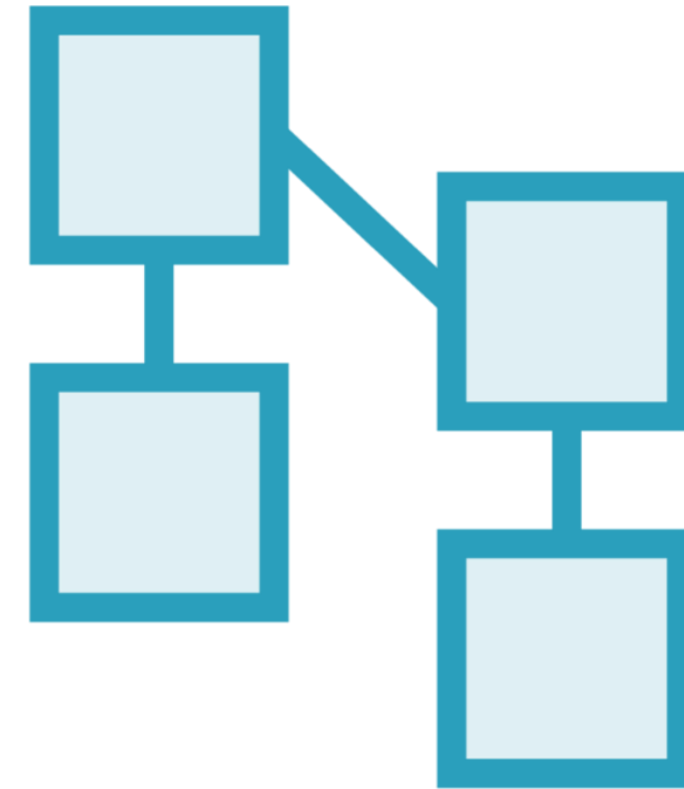


## Soporte de XML

# Modelo exterior vs Modelo de Entidad



**El modelo de entidad  
representa las filas de la  
base de datos como  
objetos**



**El modelo exterior representa lo  
que se envía por la red**

# Modelo exterior vs Modelo de Entidad

## Modelo exterior (AuthorDto)

```
Guid Id
string FirstName
string LastName
int Age
```

## Modelo de Entidad (Author)

```
Guid Id
string FirstName
string LastName
DateTimeOffset DateOfBirth
```



# Modelo exterior vs Modelo de Entidad

## Modelo exterior (AuthorDto)

`Guid Id`

`string Name`

`int Age`

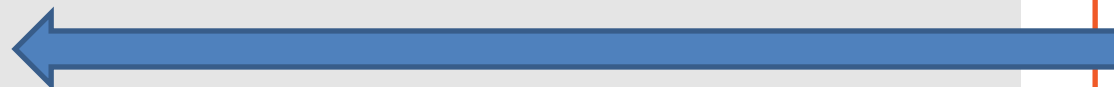
## Modelo de Entidad (Author)

`Guid Id`

`string FirstName`

`string LastName`

`DateTimeOffset DateOfBirth`



# Modelo exterior vs Modelo de Entidad

## Modelo exterior (AuthorDto)

```
Guid Id  
string Name  
int Age  
float Royalties
```

## Modelo de Entidad (Author)

```
Guid Id  
string FirstName  
string LastName  
DateTimeOffset DateOfBirth
```

La separación de los modelos externos  
y de entidad permite obtener un código  
más robusto, fiable y evolutivo

**Declaración importante**



# Soporte de HEAD

Método HTTP	Carga de la petición	URI Ejemplo	Carga de la respuesta
GET	-	/api/authors /api/authors/{authorId}	colección de autores un autor
POST	un autor	/api/authors	un autor
PUT	un autor	/api/authors/{authorId}	un autor o vacío
PATCH	JsonPatchDocument sobre autor	/api/authors/{authorId}	un autor o vacío
DELETE	-	/api/authors/{authorId}	-
HEAD	-	/api/authors /api/authors/{authorId}	-
OPTIONS	-	/api/...	-

# Soporte de HEAD

- **HEAD es idéntico a GET, con la notable diferencia de que la API no debe devolver un cuerpo de respuesta**
- **Puede utilizarse para obtener información sobre el recurso**

# Demo



Soporte de HEAD

# Resumen



- **Contrato exterior**
  - **Identificadores de recursos**
  - **Métodos HTTP**
  - **Carga opcional (Tipo de medios)**

# Resumen



- **Identificadores de recursos**
  - Utilizar sustantivos pluralizados que transmitan significado
  - Representar la jerarquía del modelo
  - Ser coherente

# Resumen



- **Métodos HTTP**
  - GET
  - POST
  - PUT/PATCH
  - DELETE
  - HEAD
  - OPTIONS

# Resumen



**El enrutamiento hace coincidir un URI de solicitud con una acción en un controlador**

# Resumen



- **La negociación del contenido es el proceso de selección de la mejor representación para una respuesta dada cuando hay múltiples representaciones disponibles**



# Resumen



- **Códigos de estado**
  - Nivel 200: correcto
  - Nivel 400: errores (cliente)
  - Nivel 500: fallos (servidor)

# Resumen



**No devuelvas trazas de la pila de ejecución a los clientes de la API, ya que no les sirven de nada.**

# Resumen



- **El modelo de entidad representa las filas de la base de datos como objetos, el modelo exterior representa lo que se envía por la red.**
- **Separar estos elementos produce un código más robusto, fiable y evolutivo**

# Resumen



**Utiliza HEAD para obtener información sobre un recurso**

A continuación:  
Gestión de recursos

---