



Trabajo Gran DAO

Tomando como base la aplicación desarrollada en anteriores entregas realizar una aplicación que permita al usuario obtener información almacenada en múltiples fuentes de datos tales como:

- Bases de datos relacionales SQL (MariaDB)
- Bases de datos no relacionales NoSQL (MongoDB)
- Ficheros XML.
- Ficheros y directorios.

Componentes a implementar de la aplicación:

- **Controladores REST:**
 - En esta capa para cada entidad persistida en fuentes de datos se deben implementar las operaciones CRUD.
- **Lógica de negocio:**
 - En esta capa se deben implementar las validaciones necesarias.
- **DAOs:**
 - Para Mysql y MongoDB con utilizar las interfaces JPARepository y MongoRepository es suficiente.
 - Para los ficheros XML se debe implementar el DAO correspondiente. Se debe utilizar la librería JAXB.
 - Para los ficheros de texto se debe implementar el DAO correspondiente. Se debe utilizar la librería File.
- **DTOS:**
 - Tantos como entidades existan en las diferentes fuentes de datos.

Se debe respetar la nomenclatura de los métodos HTTP implementado una API RESTFUL, así como la **separación de responsabilidades por cada capa**.

El trabajo podrá ser realizado en parejas o individualmente. NO vale cambiarse de pareja a individual y viceversa una vez empezado el proyecto.

Todas las peticiones se harán a través de la [herramienta Postman](#).

Entregar el enlace al repositorio en github en un fichero txt.

Incluir pantallazos de pruebas en un fichero PDF.

David Brasero
Rafael Haro

MongoDB

```
    },
    {
      _id: 1,
      idUsuario: Long('105'),
      idLibro: Long('205'),
      fechaPrestamo: ISODate('2025-02-24T23:00:00.000Z'),
      fechaDevolucion: ISODate('2025-03-09T23:00:00.000Z'),
      penalizacionHasta: ISODate('2025-03-14T23:00:00.000Z'),
      _class: 'Grandao.DTO.Prestamo'
    }
  ]
prestamos> db.prestamos_v2.find().pretty()
[
  {
    _id: 0,
    idUsuario: Long('101'),
    idLibro: Long('202'),
    fechaPrestamo: ISODate('2025-02-24T23:00:00.000Z'),
    fechaDevolucion: ISODate('2025-03-09T23:00:00.000Z'),
    penalizacionHasta: ISODate('2025-03-14T23:00:00.000Z'),
    _class: 'Grandao.DTO.Prestamo'
  },
  {
    _id: 1,
    idUsuario: Long('106'),
    idLibro: Long('205'),
    fechaPrestamo: ISODate('2025-02-24T23:00:00.000Z'),
    fechaDevolucion: ISODate('2025-03-09T23:00:00.000Z'),
    penalizacionHasta: ISODate('2025-03-14T23:00:00.000Z'),
    _class: 'Grandao.DTO.Prestamo'
  }
]
prestamos> 
```

Overview GET http POST http GET http POST http

PUT http://localhost:8090/compras

PUT http://localhost:8081/biblioteca/prestamos/1

Params Authorization Headers (9) Body Scripts

none form-data x-www-form-urlencoded raw

```
1 {
2   "idUsuario": 106,
3   "idLibro": 205,
4   "fechaPrestamo": "2025-02-25",
5   "fechaDevolucion": "2025-03-10",
6   "penalizacionHasta": "2025-03-15"
7 }
8
```

Body Cookies Headers (4) Test Results

Raw Preview Visualize

1

http://localhost:8090/compras

GET http://localhost:8081/biblioteca/prestamos

Params Authorization Headers (9) Body Scripts Settings

Body Cookies Headers (5) Test Results

{ } JSON Preview Visualize

```
1 [
2   {
3     "id": 0,
4     "idUsuario": 101,
5     "idLibro": 202,
6     "fechaPrestamo": "2025-02-25",
7     "fechaDevolucion": "2025-03-10",
8     "penalizacionHasta": "2025-03-15"
9   },
10  {
11    "id": 1,
12    "idUsuario": 106,
13    "idLibro": 205,
14    "fechaPrestamo": "2025-02-25",
15    "fechaDevolucion": "2025-03-10",
16    "penalizacionHasta": "2025-03-15"
17  }
18 ]
```

MongoDB

HTTP http://localhost:8090/compras

POST http://localhost:8081/biblioteca/prestamos

Params Authorization Headers (9) Body Scripts Settings

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL

```
1 { "id": 2,
2   "idUserario": 106,
3   "idLibro": 205,
4   "fechaPrestamo": "2025-02-28",
5   "fechaDevolucion": "2025-03-19",
6   "penalizacionHasta": "2025-03-31"
7 }
8
```

```
]
prestamos> db.prestamos_v2.find().pretty()
[
  {
    _id: 0,
    idUsuario: Long('101'),
    idLibro: Long('202'),
    fechaPrestamo: ISODate('2025-02-24T23:00:00.000Z'),
    fechaDevolucion: ISODate('2025-03-09T23:00:00.000Z'),
    penalizacionHasta: ISODate('2025-03-14T23:00:00.000Z'),
    _class: 'Grandao.DTO.Prestamo'
  },
  {
    _id: 1,
    idUsuario: Long('106'),
    idLibro: Long('205'),
    fechaPrestamo: ISODate('2025-02-24T23:00:00.000Z'),
    fechaDevolucion: ISODate('2025-03-09T23:00:00.000Z'),
    penalizacionHasta: ISODate('2025-03-14T23:00:00.000Z'),
    _class: 'Grandao.DTO.Prestamo'
  },
  {
    _id: 2,
    idUsuario: Long('106'),
    idLibro: Long('205'),
    fechaPrestamo: ISODate('2025-02-27T23:00:00.000Z'),
    fechaDevolucion: ISODate('2025-03-18T23:00:00.000Z'),
    penalizacionHasta: ISODate('2025-03-30T22:00:00.000Z'),
    _class: 'Grandao.DTO.Prestamo'
  }
]
```

HTTP http://localhost:8090/compras

DELETE http://localhost:8081/biblioteca/prestamos/2

Params Authorization Headers (9) Body Scripts Settings

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL **JSON**

```
1 { "id": 2,
2   "idUserario": 106,
3   "idLibro": 205,
4   "fechaPrestamo": "2025-02-28",
5   "fechaDevolucion": "2025-03-19",
6   "penalizacionHasta": "2025-03-31"
7 }
8
```

```
]
prestamos> db.prestamos_v2.find().pretty()
[
  {
    _id: 0,
    idUsuario: Long('101'),
    idLibro: Long('202'),
    fechaPrestamo: ISODate('2025-02-24T23:00:00.000Z'),
    fechaDevolucion: ISODate('2025-03-09T23:00:00.000Z'),
    penalizacionHasta: ISODate('2025-03-14T23:00:00.000Z'),
    _class: 'Grandao.DTO.Prestamo'
  },
  {
    _id: 1,
    idUsuario: Long('106'),
    idLibro: Long('205'),
    fechaPrestamo: ISODate('2025-02-24T23:00:00.000Z'),
    fechaDevolucion: ISODate('2025-03-09T23:00:00.000Z'),
    penalizacionHasta: ISODate('2025-03-14T23:00:00.000Z'),
    _class: 'Grandao.DTO.Prestamo'
  }
]
prestamos>
```

Ficheros TXT

GET http://localhost:8081/biblioteca/libros

Params Authorization Headers (9) Body Scripts Settings

Body Cookies Headers (5) Test Results

{ } JSON Preview Visualize

```
1 [
2   {
3     "isbn": "978-1234567890",
4     "titulo": "El Quijote",
5     "autor": "Miguel de Cervantes"
6   },
7   {
8     "isbn": "978-9876543210",
9     "titulo": "Cien años de soledad",
10    "autor": "Gabriel García Márquez"
11  },
12  {
13    "isbn": "978-0140283334",
14    "titulo": "1984",
15    "autor": "George Orwell"
16  },
17  {
18    "isbn": "978-0345391803",
19    "titulo": "The Hobbit",
20    "autor": "J.R.R. Tolkien"
21  },
22  {
23    "isbn": "978-0061120084",
24    "titulo": "To Kill a Mockingbird",
25    "autor": "Harper Lee"
26  },
27  {
28    "isbn": "978-0451524935",
29    "titulo": "Brave New World",
30    "autor": "Aldous Huxley"
31  },
32  {
33    "isbn": "978-0679404678",
34    "titulo": "The Great Gatsby",
35    "autor": "F. Scott Fitzgerald"
36  },
37 ]
```

FicheroLibros.txt — Kate

Archivo Editar Ver Proyectos Cliente LSP Marcadores Sesiones Herramientas

FicheroLibros.txt

```
1 978-1234567890,El Quijote,Miguel de Cervantes
2 978-9876543210,Cien años de soledad,Gabriel García Márquez
3 978-0140283334,1984,George Orwell
4 978-0345391803,The Hobbit,J.R.R. Tolkien
5 978-0061120084,To Kill a Mockingbird,Harper Lee
6 978-0451524935,Brave New World,Aldous Huxley
7 978-0679404678,The Great Gatsby,F. Scott Fitzgerald
8 978-0062268533,Fahrenheit 451,Ray Bradbury
9 978-0316769488,The Catcher in the Rye,J.D. Salinger
10 978-0385751477,The Hunger Games,Suzanne Collins
11 978-0062268534,Fahrenheit 51,Ray Bradbury
12
```

Salida Buscar y sustituir

POST http://localhost:8081/biblioteca/libros

Params Authorization Headers (9) Body Scripts Settings

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL

```
1 {
2   "isbn": "978-8498726138",
3   "titulo": "Nacidos de la bruma",
4   "autor": "Brandon Sanderson"
5 }
6
```

FicheroLibros.txt — Kate

Archivo Editar Ver Proyectos Cliente LSP Marcadores Sesiones Herramientas

FicheroLibros.txt

```
1 978-1234567890,El Quijote,Miguel de Cervantes
2 978-9876543210,Cien años de soledad,Gabriel García Márquez
3 978-0140283334,1984,George Orwell
4 978-0345391803,The Hobbit,J.R.R. Tolkien
5 978-0061120084,To Kill a Mockingbird,Harper Lee
6 978-0451524935,Brave New World,Aldous Huxley
7 978-0679404678,The Great Gatsby,F. Scott Fitzgerald
8 978-0062268533,Fahrenheit 451,Ray Bradbury
9 978-0316769488,The Catcher in the Rye,J.D. Salinger
10 978-0385751477,The Hunger Games,Suzanne Collins
11 978-0062268534,Fahrenheit 51,Ray Bradbury
12 978-8498726138,Nacidos de la bruma,Brandon Sanderson
13
```

Ficheros XML

POST http://localhost:8081/biblioteca/usuarios

Params Authorization Headers (9) Body Scripts Settings

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL ☐ JSON

```
1 {
2   "id": 1,
3   "dni": "67821394G",
4   "nombre": "Pablo",
5   "email": "pablo@example.com",
6   "password": "password123",
7   "tipo": "administrador",
8   "penalizacionHasta": null
9 }
```

This XML file does not appear to have any style information associated with it. The document below.

```
<?xml version="1.0"?>
<usuarios>
  <usuario>
    <id>0</id>
    <dni>12345678A</dni>
    <nombre>Pedro</nombre>
    <email>pedro@example.com</email>
    <password>password123</password>
    <tipo>usuario</tipo>
  </usuario>
  <usuario>
    <id>1</id>
    <dni>67821394G</dni>
    <nombre>Pablo</nombre>
    <email>pablo@example.com</email>
    <password>password123</password>
    <tipo>administrador</tipo>
  </usuario>
</usuarios>
```

GET http://localhost:8081/biblioteca/usuarios

Params Cookies Headers (5) Body Test Results

Body Cookies Headers (5) Test Results

☒ JSON ☐ Preview ☐ Visualize

```
1 [
2   {
3     "id": 0,
4     "dni": "12345678A",
5     "nombre": "Pedro",
6     "email": "pedro@example.com",
7     "password": "password123",
8     "tipo": "usuario",
9     "penalizacionHasta": null
10  },
11  {
12    "id": 1,
13    "dni": "67821394G",
14    "nombre": "Pablo",
15    "email": "pablo@example.com",
16    "password": "password123",
17    "tipo": "administrador",
18    "penalizacionHasta": null
19  }
20 ]
```

This XML file does not appear to have any style information associated with it. The document below.

```
<?xml version="1.0"?>
<usuarios>
  <usuario>
    <id>0</id>
    <dni>12345678A</dni>
    <nombre>Pedro</nombre>
    <email>pedro@example.com</email>
    <password>password123</password>
    <tipo>usuario</tipo>
  </usuario>
  <usuario>
    <id>1</id>
    <dni>67821394G</dni>
    <nombre>Pablo</nombre>
    <email>pablo@example.com</email>
    <password>password123</password>
    <tipo>administrador</tipo>
  </usuario>
</usuarios>
```


SpingBoot

☐ Perfilando [[Editar en línea](#)] [[Editar](#)] [[Explicar SQL](#)] [[Crear código PHP](#)] [[Actualizar](#)]

☐ Mostrar todo | Número de filas: 25 | Filtrar filas:

Opciones extra

<input type="checkbox"/>	Editar	Copiar	Borrar	1	9781234567890	Prestado
<input type="checkbox"/>	Editar	Copiar	Borrar	2	9781234567890	Disponible
<input type="checkbox"/>	Editar	Copiar	Borrar	3	9781234567890	Prestado
<input type="checkbox"/>	Editar	Copiar	Borrar	4	9789876543210	Disponible
<input type="checkbox"/>	Editar	Copiar	Borrar	5	9789876543210	Dañado
<input type="checkbox"/>	Editar	Copiar	Borrar	6	9781111111111	Disponible
<input type="checkbox"/>	Editar	Copiar	Borrar	7	9781111111111	Prestado

☐ Seleccionar todo | Para los elementos que están marcados: [Editar](#)

Environments

Flows

History

GET

Params Authorization Headers (9) Body Scripts Settings

Body JSON

200 OK

```
{
  "id": 1,
  "isbn": "9781234567890",
  "estado": "Prestado"
},
{
  "id": 2,
  "isbn": "9781234567890",
  "estado": "Disponible"
},
{
  "id": 3,
  "isbn": "9781234567890",
  "estado": "Prestado"
}
```

☐ Perfilando [[Editar en línea](#)] [[Editar](#)] [[Explicar SQL](#)] [[Crear código PHP](#)] [[Actualizar](#)]

☐ Mostrar todo | Número de filas: 25 | Filtrar filas:

Opciones extra

<input type="checkbox"/>	Editar	Copiar	Borrar	4	9789876543210	Disponible
<input type="checkbox"/>	Editar	Copiar	Borrar	5	9789876543210	Dañado
<input type="checkbox"/>	Editar	Copiar	Borrar	6	9781111111111	Disponible
<input type="checkbox"/>	Editar	Copiar	Borrar	7	9781111111111	Prestado
<input type="checkbox"/>	Editar	Copiar	Borrar	8	978-8498726138	Prestado

☐ Seleccionar todo | Para los elementos que están marcados: [Editar](#)

Environments

Flows

History

POST

Params Authorization Headers (9) Body Scripts Settings

Body raw

```
{
  "id": 8,
  "isbn": "978-8498726138",
  "estado": "Prestado"
}
```

SpingBoot

The screenshot shows a REST client interface with a table of book records on the left and a PUT request configuration on the right. A blue arrow points from the 8th row of the table to the request body.

	Editar	Copiar	Borrar	ID	ISBN	Estado
<input type="checkbox"/>				3	9781234567890	Prestado
<input type="checkbox"/>				4	9789876543210	Disponible
<input type="checkbox"/>				5	9789876543210	Dañado
<input type="checkbox"/>				6	9781111111111	Disponible
<input type="checkbox"/>				7	9781111111111	Prestado
<input type="checkbox"/>				8	978-8498726138	Dañado

↑ ☐ Seleccionar todo Para los elementos que están marcados: ☐ Editar

PUT `http://localhost:8081/biblioteca/ejemplares/8`

Params Authorization Headers (9) **Body** Scripts Settings

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL JS

```
1 {
2   "id": 8,
3   "isbn": "978-8498726138",
4   "estado": "Dañado"
5 }
```

The screenshot shows the same REST client interface, but now configured for a DELETE request. The table on the left is partially visible, and the right panel shows the DELETE request configuration.

	Editar	Copiar	Borrar	ID	ISBN	Estado
<input type="checkbox"/>				3	9781234567890	Prestado
<input type="checkbox"/>				4	9789876543210	Disponible
<input type="checkbox"/>				5	9789876543210	Dañado
<input type="checkbox"/>				6	9781111111111	Disponible
<input type="checkbox"/>				7	9781111111111	Prestado

↑ ☐ Seleccionar todo Para los elementos que están marcados: ☐ Editar

DELETE `http://localhost:8081/biblioteca/ejemplares/8`

Params Authorization Headers (9) **Body** Scripts Settings

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL JS

```
1 {
2   "id": 8,
3   "isbn": "978-8498726138",
4   "estado": "Dañado"
5 }
```