

# Examen de Desarrollo con Spring Boot

## Base de Datos: Biblioteca

Trabajaremos con una base de datos para gestionar una biblioteca. Las entidades principales serán:

- **Libros** (id, título, autor, año de publicación, género)
  - **Usuarios** (id, nombre, correo, fecha de registro)
  - **Préstamos** (id, usuario\_id, libro\_id, fecha\_prestamo, fecha\_devolucion, estado)
- 

## Preguntas

### 1. Modelado de Entidades (1 punto)

Crea las entidades **Libro**, **Usuario** y **Préstamo** con sus atributos y relaciones.

### 2. Relaciones entre Entidades (0.5 puntos)

Define las relaciones:

- Un usuario puede tener varios préstamos.
- Un libro puede estar en varios préstamos, pero no puede ser prestado a más de un usuario al mismo tiempo.

### 3. Repositorios JPA (0.5 puntos)

Crea los repositorios JPA para cada entidad.

### 4. Listar libros por género (0.5 puntos)

Crea una vista `libros.html` que permita filtrar los libros por género.

### 5. Contar los libros disponibles (0.5 puntos)

Muestra en la vista `libros.html` cuántos libros no están actualmente en préstamo.

### 6. Listado de préstamos activos con opción de devolución (2 puntos)

Crea una vista `prestamos.html` que muestre todos los préstamos activos con un botón "Devolver" que cambie el estado del préstamo y actualice la fecha de devolución.

### 7. CRUD con REST sobre préstamos (3 puntos)

Crea un CRUD con API REST para la entidad **Préstamo**, permitiendo crear, leer, actualizar y eliminar registros. Cada préstamo debe mostrar el usuario y el libro correspondiente.

### 8. Mostrar la cantidad de libros prestados por usuario (2 puntos)

Modifica la API REST para que cada usuario muestre cuántos libros tiene actualmente en préstamo.

### 9. Listar libros prestados con el nombre del usuario (1 punto)

Crea una consulta que muestre una lista de los libros que están actualmente en préstamo junto con el nombre del usuario que los tiene.

**10. Contar cuántos libros ha tomado prestados cada usuario (1 punto)**

Crea una consulta que muestre el nombre de cada usuario y la cantidad total de libros que ha tomado prestados (incluyendo los devueltos).

**11. Obtener el historial de préstamos de un usuario específico (1 punto)**

Crea un endpoint REST que, dado un `idUsuario`, devuelva el historial de libros que ha prestado, con las fechas de préstamo y devolución.

**12. Implementar búsqueda de libros por título y autor (1 punto)**

Modifica el sistema para permitir buscar libros por su título o por el nombre del autor, utilizando una vista en **Thymeleaf** o una consulta REST.

**14. Configuración básica de autenticación en Spring Security (1 punto)**

Explica cómo configurar la autenticación en una aplicación Spring Boot usando **Spring Security** con usuarios en memoria. Proporciona un ejemplo de código básico.

**15. Restricción de acceso a endpoints (1 punto)**

¿Cómo puedes restringir el acceso a ciertos endpoints en una aplicación con Spring Security? Escribe un fragmento de código donde solo los usuarios con el rol **"ADMIN"** puedan acceder a `/admin`.

```
CREATE TABLE Libro (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    titulo VARCHAR(255) NOT NULL,  
    autor VARCHAR(255) NOT NULL,  
    año_publicacion INT NOT NULL,  
    genero ENUM('FICCION', 'NO_FICCION', 'MISTERIO', 'FANTASIA', 'CIENCIA_FICCION',  
'HISTORIA') NOT NULL  
);  
  
CREATE TABLE Usuario (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    nombre VARCHAR(255) NOT NULL,  
    correo VARCHAR(255) UNIQUE NOT NULL,  
    fecha_registro DATE NOT NULL  
);
```

```

CREATE TABLE Prestamo (
    id INT AUTO_INCREMENT PRIMARY KEY,
    usuario_id INT NOT NULL,
    libro_id INT NOT NULL,
    fecha_prestamo DATE NOT NULL,
    fecha_devolucion DATE,
    estado ENUM('ACTIVO', 'DEVUELTO') NOT NULL DEFAULT 'ACTIVO',
    FOREIGN KEY (usuario_id) REFERENCES Usuario(id) ON DELETE CASCADE,
    FOREIGN KEY (libro_id) REFERENCES Libro(id) ON DELETE CASCADE
);

```

```

INSERT INTO Libro (titulo, autor, fecha_publicacion, genero) VALUES
('1984', 'George Orwell', '1949-06-08', 'FICCION'),
('Cien años de soledad', 'Gabriel García Márquez', '1967-05-30', 'FICCION'),
('El código Da Vinci', 'Dan Brown', '2003-03-18', 'MISTERIO'),
('Dune', 'Frank Herbert', '1965-08-01', 'CIENCIA_FICCION'),
('El señor de los anillos', 'J.R.R. Tolkien', '1954-07-29', 'FANTASIA'),
('Sapiens', 'Yuval Noah Harari', '2011-09-04', 'HISTORIA'),
('Los pilares de la Tierra', 'Ken Follett', '1989-10-12', 'FICCION'),
('Crimen y castigo', 'Fiódor Dostoievski', '1866-01-12', 'MISTERIO'),
('La Odisea', 'Homero', '-0800-05-01', 'HISTORIA'),
('Fundación', 'Isaac Asimov', '1951-06-01', 'CIENCIA_FICCION');

```

```

INSERT INTO Usuario (nombre, correo, fecha_registro) VALUES
('Juan Pérez', 'juanperez@email.com', '2023-05-10'),
('María López', 'marialopez@email.com', '2023-06-15'),
('Carlos García', 'carlosgarcia@email.com', '2023-07-20'),
('Ana Fernández', 'anafernandez@email.com', '2023-08-25'),
('Pedro Martínez', 'pedromartinez@email.com', '2023-09-12'),
('Laura Sánchez', 'laurasanchez@email.com', '2023-10-05'),
('Miguel Torres', 'migueltorres@email.com', '2023-11-18'),

```

```
('Elena Ramírez', 'elenaramirez@email.com', '2023-12-01');
```

```
INSERT INTO Prestamo (usuario_id, libro_id, fecha_prestamo, fecha_devolucion, estado)  
VALUES
```

```
(1, 2, '2024-03-01', NULL, 'ACTIVO'),
```

```
(2, 3, '2024-03-05', NULL, 'ACTIVO'),
```

```
(3, 1, '2024-02-28', '2024-03-10', 'DEVUELTO'),
```

```
(4, 4, '2024-03-10', NULL, 'ACTIVO'),
```

```
(1, 5, '2024-02-20', '2024-02-28', 'DEVUELTO'),
```

```
(5, 6, '2024-03-02', NULL, 'ACTIVO'),
```

```
(6, 7, '2024-03-03', NULL, 'ACTIVO'),
```

```
(7, 8, '2024-02-15', '2024-03-05', 'DEVUELTO'),
```

```
(8, 9, '2024-02-22', '2024-02-28', 'DEVUELTO'),
```

```
(2, 10, '2024-03-07', NULL, 'ACTIVO');
```