



1. Haz los entities que creas convenientes (1 punto).
2. Establece las relaciones entre entities (0.5 puntos).
3. Haz los repositorios JPA pertinentes (0.5 puntos).
4. Muestra en una vista "alumnosjpa.html" los alumnos comprendidos entre las edades 21 y 24 años. (0.5 puntos).
5. Muestra en la misma vista "alumnosjpa.html" el número de alumnos que hay en la base de datos. (0.5 puntos).
6. Haz con SpringMVC una vista llamada "alumnosList.html" que muestre los datos de todos los alumnos (solo de la tabla alumnos). Encuentra la manera de crear un botón "eliminar" asociado a cada alumno, si se pulsa este botón, se eliminará el alumno de la base de datos y se listará el resto de alumnos en la vista (2 puntos).
7. Haz un CRRUD con REST sobre la tabla alumnos, por cada alumno deberemos ver en qué asignaturas está matriculado. (3 puntos).
8. Busca la manera de poner junto a cada alumno, la nota media obtenida en todas las asignaturas en las que está matriculado (no la nota en cada una de ellas, la nota media que conforman todas. (2 puntos)

```

-- Insertar datos en la tabla alumnos
INSERT INTO alumnos (nombre, apellidos, edad) VALUES
('Juan', 'Pérez', 20),
('María', 'González', 22),
('Carlos', 'Sánchez', 19),
('Lucía', 'Ramírez', 21),
('Nini', 'Empanado', 23),
('Ana', 'Martínez', 23);

-- Insertar datos en la tabla asignaturas
INSERT INTO asignaturas (nombre) VALUES
('Matemáticas'),
('Historia'),
('Física'),
('Química'),
('Biología');

-- Insertar datos en la tabla matriculas
  
```

```
INSERT INTO matriculas (idalumno, idasignatura, curso, nota) VALUES
(1, 1, '2022-09-16', 2.50),
(1, 2, '2023-01-10', 7.00),
(1, 1, '2023-01-10', 8.50),
(2, 3, '2023-09-20', 8.75),
(2, 4, '2024-02-15', 9.50),
(4, 1, '2025-03-10', 8.00),
(4, 2, '2025-06-25', 8.00),
(5, 3, '2023-11-01', 9.00),
(5, 4, '2024-08-30', 8.00),
(5, 5, '2024-12-15', 9.00);
```

```
<dependencies>

    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-actuator</artifactId>
    </dependency>

    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-data-jpa</artifactId>
    </dependency>

    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-devtools</artifactId>
        <scope>runtime</scope>
        <optional>true</optional>
    </dependency>
    <dependency>
        <groupId>com.mysql</groupId>
        <artifactId>mysql-connector-j</artifactId>
        <scope>runtime</scope>
    </dependency>

    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-test</artifactId>
        <scope>test</scope>
    </dependency>

    <dependency>
        <groupId>org.springframework.boot</groupId>
```

```
        <artifactId>spring-boot-starter-web</artifactId>
    </dependency>

    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-thymeleaf</artifactId>
    </dependency>
</dependencies>
```

```
spring.application.name=ExamenMayo
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
spring.datasource.url=jdbc:mysql://localhost:3306/institutodbd
spring.datasource.username=root
spring.datasource.password=root
spring.sql.init.encoding=UTF-8
spring.jpa.generate-ddl=false
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQLDialect
spring.jpa.show-sql=true
spring.jpa.hibernate.naming.physical-strategy=org.hibernate.boot.model.naming.PhysicalNamingStrategyStandardImpl
spring.output.ansi.enabled=always
```

OJO CON LOS VALORES, NO TENDRÉIS LOS MISMOS