



Universidad Libre  
Departamento de Ingeniería de Sistemas  
Programación WEB – 2023-2  
Taller. Aplicación MVC con servicios Rest – Backend (trabajo en grupo)  
Profesor Jorge Valenzuela

## 1. Objetivo del Taller

La motivación de este taller es desarrollar la parte del Backend de una aplicación Web que sigue el patrón de arquitectura MVC y utiliza microservicios tipo Rest en un framework Spring con las siguientes dependencias tecnológicas:

### Spring Web

WEB

Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.

### Spring Data JPA

SQL

Persist data in SQL stores with Java Persistence API using Spring Data and Hibernate.

### PostgreSQL Driver

SQL

A JDBC and R2DBC driver that allows Java programs to connect to a PostgreSQL database using standard, database independent Java code.

Adicionalmente se sugiere utilizar la librería **Lombok**, para ahorrar la escritura de código utilizando anotaciones. Los componentes a desarrollar que conforman la arquitectura de la aplicación solicitada en el taller son:

#### 1.1 Repositorio

Componente abstracto (interfaz) que separa la lógica de manejo de datos de la lógica de negocio, mediando entre los objetos del dominio y la Base de datos que se encarga de su persistencia, ofreciendo una colección de objetos en el modelo a través de un rico conjunto de métodos predefinidos automáticamente por una implementación específica, en nuestro caso será **JpaRepository**.

#### 1.2 Controlador

Componente para definir y manejar el API de los servicios web Rest que utilizan peticiones HTTP para retornar respuestas de datos en formato Json.

#### 1.3 Excepcion

Componente encargado de hacer validaciones de lógica de negocio específica de la aplicación para mostrar mensajes explicativos al usuario.

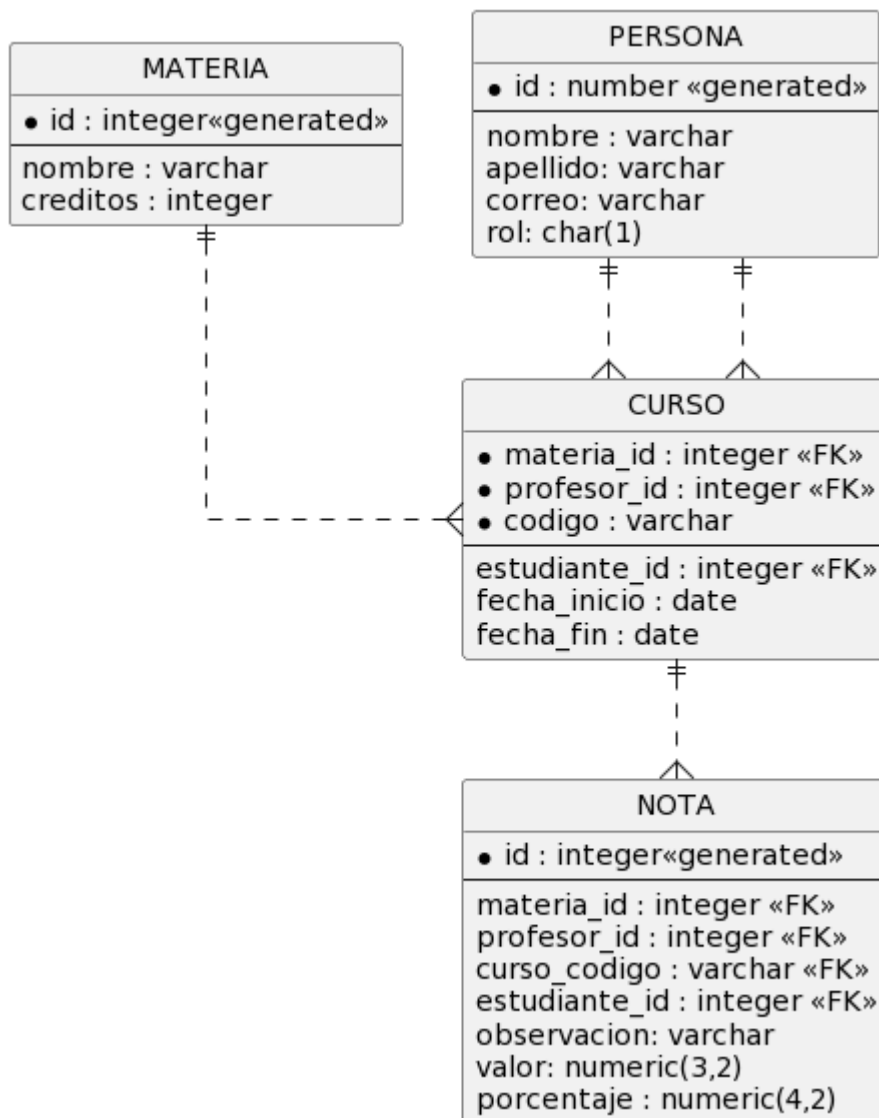


#### 1.4 Modelo

Clases simples de Java (POJO) que abstraen las tablas de la BD y sirven como argumentos de entrada / salida para los servicios web.

#### 2. Modelos de datos

Las tablas que conforman el modelo de datos se puede ver en el siguiente diagrama entidad relación:





Universidad Libre  
Departamento de Ingeniería de Sistemas  
Programación WEB – 2023-2  
Taller. Aplicación MVC con servicios Rest – Backend (trabajo en grupo)  
Profesor Jorge Valenzuela

Modelo que corresponde a la siguiente definición en lenguaje SQL para PostgreSQL:

```
CREATE TABLE materia (  
    id integer GENERATED ALWAYS AS IDENTITY PRIMARY KEY,  
    nombre varchar,  
    credits integer  
);  
CREATE TABLE persona (  
    id integer GENERATED ALWAYS AS IDENTITY PRIMARY KEY,  
    nombre varchar,  
    apellido varchar,  
    correo varchar,  
    rol char(1) DEFAULT 'E'  
);  
CREATE TABLE curso (  
    materia_id integer NOT NULL,  
    profesor_id integer NOT NULL,  
    codigo varchar,  
    estudiante_id integer,  
    fecha_inicio date,  
    fecha_fin date,  
    PRIMARY KEY (materia_id, profesor_id, numero)  
);  
CREATE TABLE nota (  
    id integer GENERATED ALWAYS AS IDENTITY PRIMARY KEY,  
    materia_id integer NOT NULL,  
    profesor_id integer NOT NULL,  
    curso_codigo varchar NOT NULL,  
    estudiante_id integer NOT NULL,  
    observacion varchar,  
    valor numeric(3, 2),  
    porcentaje numeric(4,2)  
);  
ALTER TABLE persona ADD CHECK (rol in ( 'E', 'P'));  
ALTER TABLE curso ADD FOREIGN KEY (estudiante_id) REFERENCES persona (id);  
ALTER TABLE curso ADD FOREIGN KEY (profesor_id) REFERENCES persona (id);  
ALTER TABLE curso ADD FOREIGN KEY (materia_id) REFERENCES materia (id);  
ALTER TABLE nota ADD FOREIGN KEY (materia_id, profesor_id, curso_codigo) REFERENCES curso  
((materia_id, profesor_id, numero);  
ALTER TABLE nota ADD FOREIGN KEY (estudiante_id) REFERENCES persona (id);  
  
CREATE VIEW estudiante AS  
SELECT  
    id, nombre, apellido, correo
```



```
FROM persona
WHERE rol = 'E';
CREATE VIEW profesor AS
SELECT
    id, nombre, apellido, correo
FROM persona
WHERE rol = 'P';
```

Nótese que se construyen dos vistas a partir de la tabla “persona”, para distinguir entre “estudiante” y “profesor” a través de una columna discriminadora “rol”, en donde “rol” solo puede tener uno de estos dos valores, condición que está restringida con un “constraint”.

### 3. Detalles de la aplicación

El objetivo del taller es construir una aplicación que ofrezca las cuatro funcionales básicas (CRUD) de crear, traer, actualizar y borrar registros de las tablas “*estudiante*” y “*nota*” persistidas en una base de datos PostgreSQL. Las funcionalidades CRUD deben implementarse bajo el patrón MVC como servicios Web tipo Restful con formato de mensajería Json.

Se puede crear uno o muchas notas para un estudiante en un curso, pero la suma de los porcentajes no puede superar el 100%, por tanto, se debe validar al crear una nota que no se supere este porcentaje. Así mismo, la nota definitiva será el promedio de las notas ponderado por su porcentaje respectivo, también se debe validar que solo se puede obtener una nota definitiva cuando se tengan las notas (una o más) cuyo suma de porcentajes sea el 100%

Además de las funcionalidades CRUD se deben entregar dos vistas de consulta: la primera que recupere los registros de la tabla “estudiante” mostrando en una tabla con todos sus campos más un botón que al dar click permita traer las notas del estudiante en todas las materias que esté cursando en el semestre actual. Se provee un ejemplo:

#### Estudiantes

Id	Nombre	Apellido	Correo	Notas
1	Alfredo	Diaz	a.diaz@universidad.edu.co	<button>Mostrar</button>
2	Francisco	Merchan	f.mercha@universidad.edu.co	<button>Mostrar</button>
3	Rolando	Mendez	r.mendez@universidad.edu.co	<button>Mostrar</button>
4	Helena	Bennedetti	h.bennedetti@universidad.edu.co	<button>Mostrar</button>
5	Jose	Cancino	j.cancino@universidad.edu.co	<button>Mostrar</button>

La segunda una tabla que muestre para el estudiante de la fila seleccionada en el botón sus respectivas notas de los cursos que esté tomando: Se provee un ejemplo



Universidad Libre  
Departamento de Ingeniería de Sistemas  
Programación WEB – 2023-2  
Taller. Aplicación MVC con servicios Rest – Backend (trabajo en grupo)  
Profesor Jorge Valenzuela

#### Estudiante: Alfredo Diaz

Codigo Curso	Nombre Materia	Nota Definitiva
TC-001	Teoría de la computación	3.5
ED-001	Estructura de datos	3.8
PW-001	Programación WEB	4.2
ED-002	Estructura de datos	2.7
BD-001	Bases de Datos	3.7

Las tablas mostradas son solo un ejemplo, no se exigen especificaciones particulares de fuentes o colores, se deja al arbitrio de los estudiantes.

#### 4. Evaluación

La entrega se hará enviando un archivo .zip con el proyecto hasta antes de la media noche del 31 de octubre del 2023, con la sustentación del grupo completo en la sesión del día siguiente.

La evaluación del taller tendrá la siguiente escala:

- Excelente (5.0/5.0): Se desarrollan todos los servicios y consultas solicitados y se ejecutan a satisfacción, se hacen las pruebas exitosamente y se cargan y actualizan registros de las tablas en la BD a través de los servicios.
- Bueno (3.5/5.0): Se desarrollan todos los servicios y consultas solicitados, pero no se ejecutan todos a satisfacción, algunas pruebas no ejecutan exitosamente o no se cargan o actualizan los registros de las tablas en la BD.
- Aceptable (3.0/5.0): El código tiene problemas de compilación o no genera los resultados esperados y descritos en el numeral 3.
- Necesita mejoras sustanciales (2.5/5.0): El código tiene problemas de compilación o no genera resultados correctos para los servicios solicitados (las cuatro funcionalidades CRUD y traer todos los registros).
- No entregó (1.0/5.0): No entregó lo requerido o la entrega no sigue las instrucciones de este documento.

EL resultado anterior es la mitad de la nota del segundo corte, la otra mitad corresponde a la sustentación del taller que se hará individualmente y está dirigida a verificar lo aprendido por el estudiante en la elaboración del taller en grupo. Quien no sustente tendrá 1 en la sustentación, solo se admiten excusas médicas o de fuerza mayor verificable.