

Desarrollo de Sistemas con Tecnología Java

Desarrollo de aplicaciones empresariales con Jakarta EE

Práctica No. 3

Nombre: _____

Fecha: _____

Apellido Paterno

Apellido Materno

Nombre(s)

Calificación: _____

Objetivo:

El alumno realizará la implementación de un servicio REST por medio de la especificación Jakarta JAX-RS para exponer un servicio de negocio.

Instrucciones:

Como parte del desarrollo del Sistema Pixup se requiere exponer la funcionalidad para dar de alta un disco a través de un servicio REST, el cual será consumido por una aplicación implementada en angular. Dado lo anterior el servicio debe ser interoperable.

Con base en la implementación realizada de la clase DiscoServiceImpl en la Práctica No. 2 se requiere crear el recurso REST para exponer dicha funcionalidad.

La clase debe crearse en el siguiente paquete: unam.diplomado.pixup.disco.api y se debe implementar el método altaDisco.

Contrato API:

```
@Consumes(MediaType.APPLICATION_JSON)
@Produces(MediaType.APPLICATION_JSON)
@Path("discos")
public interface DiscoApi {
    @POST
    @Path("alta")
    Response altaDisco(@NotNull @Valid Disco disco);
}
```

Clase Implementación:

```
public class DiscoResource implements DiscoApi {

    @Inject
    private DiscoService discoService;

    @Override
    public Response altaDisco(Disco disco) {
        return null;
    }

}
```

Framework de validaciones: Agregar un par de anotaciones para validar 2 propiedades de la entidad Disco.

Para la implementación del método altaDisco tomar en cuenta las siguientes consideraciones:

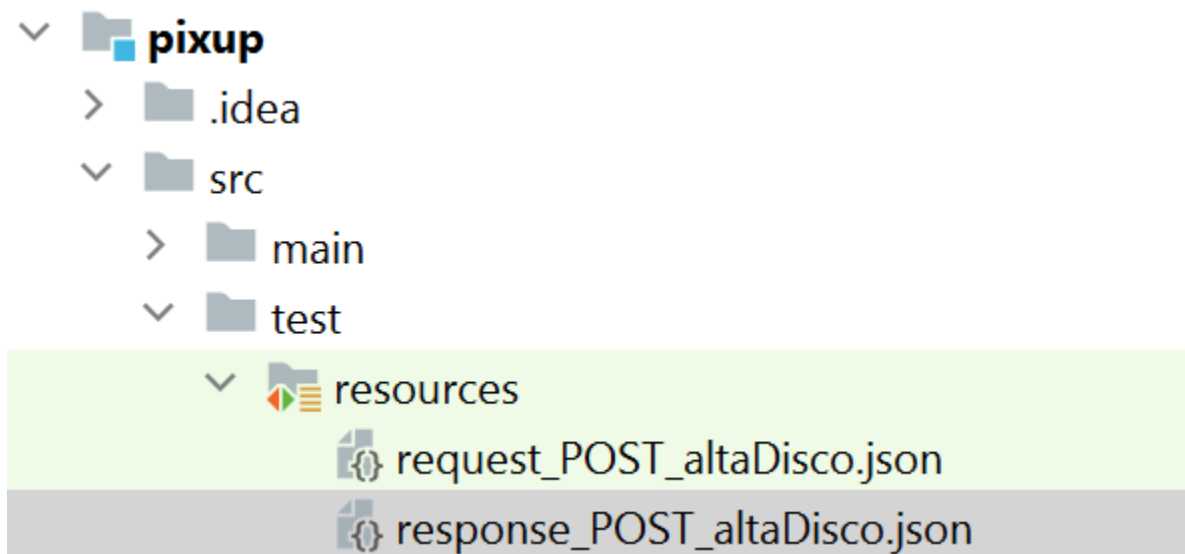
- 1) Regresar un código HTTP 201 (Created) para el escenario exitoso.
- 2) Regresar un código HTTP 409 (Conflict) para el escenario de error cuando el disco ya existe en el catálogo.
- 3) Regresar un código HTTP 422 (Unprocessable Content) o 428 (Precondition Required) para los escenarios donde no exista el artista, disquera o género musical.
- 4) Regresar un código HTTP 400 (Bad Request) para el escenario cuando los campos no cumplan con validaciones de formato.

NOTA IMPORTANTE: No hacer uso de DTOs ni ExceptionMappers, las excepciones deberán ser gestionadas por medio de bloques try/catch en DiscoResource.

En el directorio src/test/resources del proyecto agregar los siguientes archivos json con el request y el response respectivamente:

request_POST_altaDisco.json

response_POST_altaDisco.json



request_POST_altaDisco.json

```
{
  "titulo": "U2 Exitos",
  "precio": 100.30,
  "existencia": 10,
  "descuento": 10.00,
  "fechaLanzamiento": "2020-11-11T11:12:12",
  "imagen": "u2_exitos.png",
  "artista": {
    "id": 1
  },
  "disquera": {
    "id": 1
  },
  "generoMusical": {
    "id": 1
  }
}
```

response_POST_altaDisco.json

```
{
  "artista": {
    "id": 1
  },
  "descuento": 10.0,
  "disquera": {
    "id": 1
  },
  "existencia": 10,
  "fechaLanzamiento": "2020-11-11T11:12:12Z[UTC]",
  "generoMusical": {
    "id": 1
  },
  "id": 2,
  "imagen": "u2_exitos.png",
  "precio": 100.3,
  "titulo": "U2 Exitos"
}
```

Generar el siguiente entregable:

1) Archivo zip del directorio src del proyecto y pom.xml, el archivo deberá nombrarse

<nombre_completo_alumno_empezando_por_apellido_practica_3_jakarta>.zip

Valor 10 puntos
