

Desarrollo de Sistemas con Tecnología Java

Desarrollo de aplicaciones empresariales con Jakarta EE

Práctica Final

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. Aplicará principios y patrones de diseño para la generación de APIs.

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 DiscoRequestDTO disco);
}
```

Clase Implementación:

```
public class DiscoResource implements DiscoApi {

    @Inject
    private DiscoService discoService;
    @Inject
    private DiscoMapper discoMapper;

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

}
```

Framework de validaciones: Agregar un par de anotaciones para validar 2 propiedades del DTO DiscoRequestDTO.

Se deben generar DTOs tanto para el request (DiscoRequestDTO) como para el response (DiscoResponseDTO).

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: Se debe hacer uso de DTOs y de ExceptionMappers, las excepciones deberán ser gestionadas de manera centralizada por medio de ExceptionMappers por lo que no deben aparecer bloques try/catch en la clase 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



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_final_jakarta>.zip

Valor 10 puntos
