
Nombre del docente: _____

Apellidos del alumno: _____

Nombres del alumno: _____

Sección: _____ Fecha: _____

Nota

Indicaciones:

La duración es de 100 minutos.

La evaluación consta de 2 preguntas escritas y enunciado propuesto a implementar

Para la parte de implementación en código solo se permite el uso de la documentación oficial

Pregunta Abierta (3 pts)

- Respecto a los principios REST vistos en clase, describa en sus propios términos en qué consiste el principio Client-Server y además describa un principio adicional, el que usted prefiera.

Análisis de código (2 pts)

Dada la siguiente implementación de un método de testing sobre el DTO AuthorDTO indique cuál será el output a consola de la última línea de dicho método

```
public class AuthorDTO {  
    private Long id;  
  
    @NotNull  
    @Size(min = 2, max = 50)  
    private String firstname;  
  
    @Size(min = 2, max = 50)  
    private String lastname;  
  
    @Positive  
    private Integer age;  
  
    @NotNull  
    @Email  
    private String email;  
}  
  
@Test  
void shouldFail() {  
    AuthorDTO authorDTO = new AuthorDTO();  
    authorDTO.setFirstname("Jesús");  
    authorDTO.setLastname(null);  
    authorDTO.setAge(-10);  
    authorDTO.setEmail("@gmail.com");  
  
    Set<ConstraintViolation<AuthorDTO>> violations;  
    violations = validator.validate(authorDTO);  
    System.out.println(violations.size());  
}
```

Implementación en código (15 pts)

La empresa de tecnología "InnovateTech" ha experimentado problemas importantes con su actual sistema de gestión de proyectos. Los datos se pierden, los permisos son confusos y la interfaz es propensa a errores. Por eso, han contratado tus servicios para desarrollar un nuevo sistema de gestión de proyectos temporal, que permita gestionar tareas y equipos de manera segura y eficiente mientras desarrollan un sistema permanente.

El sistema debe permitir a los empleados registrarse y loguearse de forma segura. Cada empleado se identifica por un ID único, nombre, apellido, correo electrónico, departamento y fecha de ingreso. Cada empleado puede pertenecer a uno o varios proyectos, y cada proyecto tiene un líder. Los proyectos se identifican por un código único y contienen detalles como el nombre del proyecto, descripción, fecha de inicio y fecha de finalización.

Los proyectos están compuestos por varias tareas. Cada tarea tiene un código único y contiene detalles como nombre, descripción, fecha límite, y responsable (empleado asignado). Un proyecto puede tener múltiples tareas, pero cada tarea tiene solo un responsable.

El sistema debe permitir realizar acciones clave, como crear un proyecto, asignar empleados a un proyecto, crear y asignar tareas, y cambiar el estado de una tarea (por ejemplo, de "pendiente" a "en progreso" o "completada"). El sistema también debe incluir opciones para filtrar tareas por estado y obtener una lista de proyectos activos.

Además, se requiere un sistema de permisos para garantizar que solo los empleados autorizados puedan realizar ciertas acciones, como eliminar un proyecto o reasignar tareas a otros empleados. La seguridad debe ser una prioridad, con manejo adecuado de autenticación y autorización.

Tu objetivo es desarrollar este sistema de gestión de proyectos temporal, cumpliendo con todos los requerimientos mencionados, incluyendo modelado de datos, operaciones CRUD, testing, autenticación, manejo de excepciones y desarrollo de endpoints funcionales.

Rúbrica:

	2pts	1pts	0pts
Entidades y relaciones	El estudiante identifica e implementa de manera correcta todas las entidades necesarias para resolver el problema. Además, establece todas las relaciones adecuadas entre estas entidades sin cometer errores. Todos los atributos requeridos para la resolución del problema están presentes y se vinculan de manera coherente.	El estudiante identifica e implementa algunas entidades, pero puede faltar una o más. Las relaciones entre las entidades pueden tener errores en solo una entidad o estar incompletas. Los atributos del problema no están completamente relacionados o tienen errores en su conexión.	El estudiante no logra identificar ni implementar la mayoría de las entidades necesarias para el problema, o las relaciones entre ellas están incorrectas en 2 entidades o más o no existen. Los atributos y sus conexiones no se aplican correctamente, lo que indica falta de comprensión del problema.
Capa de persistencia	El estudiante define correctamente la interfaz de repositorio para cada entidad. Define correctamente las configuraciones de conexión a base de datos en el archivo <code>application.properties</code> . Define y usa por lo menos 4 query methods distribuidos entre las 4 interfaces de repositorios.	El estudiante no define la interfaz de repositorio para todas las entidades y/o presenta errores en su implementación. Define y usa menos de 4 query methods entre las 4 entidades.	El estudiante no define ninguna interfaz de repositorio. No define ningún query method.
Implementación de endpoints	El estudiante logra implementar correctamente entre 10 y 8 endpoints solicitados en el enunciado, sin ninguna falla de compilación.	El estudiante logra implementar correctamente entre 7 a 4 endpoints solicitados en el enunciado, con algunas fallas en el código o en la sintaxis en los demás endpoints.	El estudiante logra implementar correctamente menos de 4 endpoints solicitados en el enunciado, con fallas evidentes en los demás endpoints.
Manejo de excepciones	El estudiante implementa por lo menos 3 clases propias de excepción y las añade correctamente al controlador global de excepciones de modo que devuelven los código de estados adecuados.	El estudiante implementa 2 o 1 clases propias de excepción y las añade correctamente al controlador global de excepciones de modo que devuelven los código de estados adecuados.	El estudiante no implementa ninguna clase propias de excepción y/o las excepciones no devuelven los códigos de estado correctamente

	2pts	1pts	0pts
Spring Security	El estudiante implementa de manera correcta el flujo de inicio de sesión (login), registro (register) y redirección para usuarios no autorizados. Los usuarios autorizados tienen acceso a las rutas correctas, mientras que los no autorizados son redirigidos correctamente, con respuestas de error apropiadas como HTTP 401 o 403. El flujo cumple todos los requisitos y no presenta errores.	El estudiante implementa parcialmente el flujo de login, registro y redirección para usuarios no autorizados. Puede haber errores en el uso de códigos HTTP, como códigos incorrectos o respuestas ambiguas. El flujo de redirección puede tener fallas, permitiendo el acceso a usuarios no autorizados o bloqueando a usuarios legítimos.	El estudiante no logra implementar el flujo de login, registro y redirección para usuarios no autorizados de manera correcta. El uso de códigos HTTP es incorrecto o confuso, sin cumplir las normas de seguridad. Los usuarios no autorizados pueden tener acceso indebido a recursos, o el flujo se interrumpe debido a errores críticos.
Testing	El estudiante emplea Testcontainers para probar un repositorio que contiene funciones especiales. Además, realiza pruebas para al menos dos endpoints de la aplicación	El estudiante no utiliza Testcontainers para probar el repositorio o lo hace de manera erróneo/incompleta. Además, se realizan pruebas para menos de dos endpoints.	El estudiante no emplea Testcontainers para probar el repositorio, y realiza testing en menos de dos endpoints.

3 pts SonarQube: Calificador automático de calidad de código.