**Formato de Informe - Evaluación Parcial 3: Calidad, pruebas y documentación**

**Asignatura:** DSY1103 - Desarrollo Full Stack I  
**Evaluación:** Parcial 3 (35% del curso)  
**Tipo de trabajo:** Grupal (máx. 3 estudiantes)  
**Modalidad:** Taller de alto cómputo  
**Duración:** 7 horas pedagógicas  
**Evaluación de presentación:** Individual

### **✅ Estructura del Informe**

#### **1. Portada**

* Nombre del equipo
* Nombres y RUT de los integrantes
* Nombre del caso (Perfulandia, EduTech, EcoMarket)
* Fecha de entrega
* Enlace al repositorio GitHub

#### **2. Documentación del Sistema**

* Descripción general del sistema y sus microservicios
* Diagrama de arquitectura (general del sistema)
* Diagrama de flujo de datos (entre microservicios)
* Documentación generada con OpenAPI (Swagger u otro)
  + Captura de Swagger UI con los endpoints documentados
  + Anotaciones utilizadas: @Operation, @ApiResponse, etc.

#### **3. Plan de Pruebas (básico)**

* Descripción del enfoque de pruebas:
  + Pruebas unitarias (JUnit)
  + Pruebas de integración (si aplica)
* Tabla resumen:
  + Nombre del test
  + Clase o método testeado
  + Objetivo del test
  + Resultado (PASSED / FAILED)
* Capturas de ejecución (mvn test) y cobertura si aplica
* Reflexión sobre cobertura y calidad del código

#### 

#### **4. Documentación con HATEOAS**

* Explicación de qué es HATEOAS y por qué se usó
* Descripción del framework usado: Spring HATEOAS
* Capturas de:
  + Ensambladores implementados
  + Respuestas JSON con \_links
  + Controladores V2 documentados
* Relación con la navegabilidad de la API

#### **5. Despliegue en la Nube (hipotético)**

* Descripción del proveedor elegido (Render, Railway, AWS, etc.)
* Pasos básicos para el despliegue (teórico):
  + Configuración de entorno (JDK, puerto, variables)
  + Deploy desde GitHub
  + Supervisión de logs / errores
* Métodos sugeridos de monitoreo/logging (teórico)

#### **6. Conclusiones y reflexión final**

* Lecciones aprendidas sobre pruebas y documentación
* ¿Cómo mejoró la calidad del sistema?
* ¿Cómo se podría escalar o mejorar a futuro?

#### **7. Anexos**

* Enlaces a repositorio
* Swagger UI compartido por red local (si aplica)
* Capturas adicionales de Postman / Swagger / consola

**Presentación individual (evaluación oral)** Cada integrante del equipo deberá presentar:

1. Justificación de herramientas de prueba utilizadas
2. Explicación de pruebas realizadas (JUnit, Mockito)
3. Descripción de Swagger y OpenAPI en su proyecto
4. Implementación y ejemplo de respuesta con HATEOAS
5. (Opcional) Caso hipotético de despliegue en nube
6. Reflexión personal sobre el trabajo colaborativo y los aprendizajes técnicos