

## **Taller: Sistema de Gestión de un Gimnasio**

Descripción del Monolito: Un sistema monolítico para la gestión de un gimnasio que incluye manejo de miembros, clases, entrenadores y equipos. El sistema permite registrar nuevos miembros, programar clases, asignar entrenadores y gestionar el inventario de equipos.

### **Componentes principales del monolito:**

1. Gestión de Miembros
2. Gestión de Clases
3. Gestión de Entrenadores
4. Gestión de Equipos

Tarea para los estudiantes: Transformar este monolito en una arquitectura de microservicios utilizando los principios de Domain-Driven Design (DDD).

<https://github.com/jrquinte/monilito-gimnasio>

Este código proporciona una base sólida para el Sistema de Gestión de un Gimnasio como un monolito en Spring Boot. Incluye entidades para Miembros, Clases, Entrenadores y Equipos, con sus respectivos repositorios, un servicio que maneja la lógica de negocio, un controlador para exponer endpoints REST, y un DataLoader para cargar datos de ejemplo.

Los estudiantes pueden usar este código como punto de partida para su taller, donde deberán aplicar los principios de DDD para transformarlo en una arquitectura de microservicios.

### **Pasos a seguir:**

1. Analizar el dominio
2. Definir los contextos acotados
3. Definir entidades, agregados y servicios
4. Identificar microservicios (máximo 4)

### **Entregables:**

1. Diagrama de componentes o despliegue UML de la arquitectura de microservicios propuesta.
2. Código fuente refactorizado en proyectos independientes para cada microservicio.
3. (Opcional) Implementación de comunicación REST entre microservicios.
4. Presentación de 15 minutos, haciendo un demo de los microservicios en Java.

**Criterios de Evaluación:**

Criterios de Evaluación (Total: 100 puntos):

1. Diseño de la Arquitectura (30 puntos)
  - Correcta identificación de los contextos acotados (10 puntos)
  - Adecuada segregación de responsabilidades entre microservicios (10 puntos)
  - Claridad y precisión del diagrama UML (10 puntos)
2. Implementación del Código (40 puntos)
  - Correcta refactorización del monolito en microservicios (15 puntos)
  - Definición adecuada de entidades, agregados y servicios según DDD (15 puntos)
  - Calidad del código y adherencia a buenas prácticas (10 puntos)
3. Funcionalidad (20 puntos)
  - Cada microservicio funciona independientemente (10 puntos)
  - Integridad de la funcionalidad original del monolito mantenida en los microservicios (10 puntos)
4. Presentación y Demostración (10 puntos)
  - Claridad en la explicación de la arquitectura propuesta (5 puntos)
  - Demostración efectiva de los microservicios en ejecución (5 puntos)

**Rúbrica de Calificación (escala de 1 a 5):**

5.0 (Excelente): 90-100 puntos

- Demuestra una comprensión profunda de DDD y microservicios.
- La implementación es robusta, bien pensada y sigue las mejores prácticas.
- Excelente presentación y demostración del proyecto.

4.5 (Muy Bueno): 85-89 puntos

- Muy buena aplicación de los principios de DDD.
- La implementación es sólida con mínimos puntos de mejora.
- Muy buena presentación y demostración.

4.0 (Bueno): 80-84 puntos

- Buena comprensión y aplicación de DDD y microservicios.

- Implementación correcta con algunos aspectos menores por mejorar.
- Buena presentación y demostración.

3.5 (Por encima del promedio): 75-79 puntos

- Comprensión adecuada de los conceptos principales.
- Implementación satisfactoria con áreas identificables de mejora.
- Presentación y demostración claras pero con espacio para mejorar.

3.0 (Satisfactorio): 70-74 puntos

- Comprensión básica de DDD y microservicios.
- La implementación cumple con los requisitos mínimos.
- Presentación y demostración aceptables.

2.5 (Por debajo del promedio): 65-69 puntos

- Comprensión limitada de algunos conceptos clave.
- Implementación con deficiencias notables pero aún funcional.
- Presentación y demostración con claras áreas de mejora.

2.0 (Necesita mejorar): 60-64 puntos

- Falta de comprensión de varios conceptos importantes.
- Implementación con deficiencias significativas.
- Presentación y demostración débiles.

1.0-1.5 (Insuficiente): < 60 puntos

- No demuestra comprensión de los conceptos básicos de DDD y microservicios.
- Implementación incompleta o no funcional.
- No logra presentar o demostrar el proyecto de manera efectiva.