



Unidad Didáctica 3: Ejercicio de Caso Final Integrador

Programación Concurrente

1. Enunciado

Contexto General: Eres parte de un equipo de desarrollo encargado de implementar un sistema integrado para la gestión y monitoreo de datos en diversos sectores críticos. Este sistema abarca desde la gestión de pedidos y transacciones, hasta el monitoreo de sistemas de seguridad, y el procesamiento de datos de sensores en entornos extremadamente exigentes como Marte y Jurassic Park. El objetivo es crear una solución escalable y resiliente que combine las mejores prácticas de programación concurrente, microservicios, programación reactiva, y procesamiento por lotes.

Descripción del Proyecto:

El proyecto se divide en seis módulos, cada uno correspondiente a un sector específico. El desafío es integrar estos módulos en un sistema cohesivo que asegure la eficiencia, consistencia, y resiliencia a través de los diferentes contextos operativos.

1. Gestión de Pedidos en Tiempo Real:

- **Problema:** Manejo de grandes volúmenes de transacciones concurrentes, asegurando la consistencia de los datos y optimizando el rendimiento del sistema.
- **Solución:** Utilización de Spring con HikariCP para la gestión de transacciones, mejorando la concurrencia y evitando bloqueos de base de datos.

2. Monitoreo de Seguridad en Gotham City:

- **Problema:** Gestión de aspectos transversales como la seguridad y sincronización de tareas concurrentes utilizando AOP (Programación Orientada a Aspectos).
- **Solución:** Implementación de aspectos de seguridad con Spring AOP para mejorar la modularidad y seguridad del sistema.

3. Sistema de Análisis de Datos en Stark Industries:

- **Problema:** Procesamiento de grandes volúmenes de datos en tiempo real con programación multihilo, enfrentando problemas de bloqueo y condiciones de carrera.
- **Solución:** Uso de ExecutorService y técnicas de sincronización para manejar tareas concurrentes de manera eficiente.

4. Monitoreo de Dinosaurios en Jurassic Park:

- **Problema:** Gestión de grandes volúmenes de datos de sensores en tiempo real utilizando Spring WebFlux, enfrentando problemas de latencia y sincronización.
- **Solución:** Uso de flujos de datos reactivos con WebFlux para mejorar la capacidad de respuesta y manejo de datos en tiempo real.

5. Gestión de Hechizos en el Mundo Mágico:

- **Problema:** Resiliencia y gestión de microservicios en un entorno de alta carga, asegurando la eficiencia y capacidad de respuesta.
- **Solución:** Implementación de patrones como Circuit Breaker y API Gateway con Spring Cloud para manejar la resiliencia y monitoreo centralizado.

6. Procesamiento de Datos de Sensores en Marte:

- **Problema:** Procesamiento por lotes de datos de sensores en tiempo real, optimizando la eficiencia y el rendimiento utilizando Spring Batch.
- **Solución:** Configuración de Spring Batch para manejar grandes volúmenes de datos de manera eficiente, asegurando la sincronización y consistencia de los lotes procesados.

Tareas del Proyecto Integrador:

1. Diseño de la Arquitectura Integrada:

- Diseña una arquitectura que integre los seis módulos de manera eficiente, asegurando la interoperabilidad y resiliencia entre los diferentes componentes.
- Proporciona un diagrama de arquitectura que visualice la integración de los módulos.

2. Desarrollo del Sistema Integrado:

- Implementa cada módulo de acuerdo a las soluciones propuestas, asegurando que funcionen de manera cohesiva dentro del sistema global.
- Implementa mecanismos de logging y monitoreo centralizados para detectar y manejar fallos en tiempo real.

3. Pruebas y Validación:

- Realiza pruebas unitarias e integradas para asegurar que cada módulo funcione correctamente dentro del sistema integrado.
- Valida la consistencia de los datos, la capacidad de respuesta, y la resiliencia del sistema bajo carga.

4. Optimización y Escalabilidad:

- Identifica posibles cuellos de botella en el sistema y propón mejoras que aseguren la escalabilidad y rendimiento óptimo en todos los sectores.
- Considera el uso de técnicas de optimización como caching, parallel processing, y balanceo de carga.

5. Análisis Post-Implementación:

- Realiza un análisis exhaustivo del rendimiento del sistema después de la integración.
- Compara el rendimiento y la eficiencia antes y después de la implementación, utilizando métricas específicas para cada sector.
- Redacta un informe detallado que incluya gráficos, estadísticas, y recomendaciones para futuras mejoras.

2. Instrucciones de entrega

- Formato: PDF con repositorio a Github
- Extensión máxima: 15 folios
- Incluir nombre del fichero: "EjercicioFeedback_NombreApellido.pdf"

WELCOME
TO
UAX

UAX

Universidad
Alfonso X el Sabio

GRACIAS

UAX.COM