

**PRESENTACIÓN**

**Nombre del curso**

Programación Paralela

**Título del proyecto**  
Sistema Bancario de transacciones

**Integrantes**

David Alejandro Liranzo Zabala - 2023-1127

Eduardo Andres Vallejo Zabala - 2023-1096

José Manuel Santillan - 2023-1156

Naroly Anyelin Tolentino Peña - 2023-1785

**Líder del proyecto**

Edison Manuel Aguero Diaz - 2023-1130

**Fecha de entrega:**

Lunes 21 de Abril, 2025

ÍNDICE (PENDIENTE)

1. **Introducción**

**Presentación general del proyecto**

El proyecto “Simulación de Banco con Transacciones Paralelas” tendrá como finalidad reproducir el comportamiento de un sistema bancario en línea bajo condiciones de carga y concurrencia. A través de la ejecución simultánea de múltiples transacciones bancarias, se buscará demostrar cómo manejar adecuadamente la sincronización de datos compartidos, garantizar la integridad de la información y analizar el rendimiento de las operaciones paralelas.

**Justificación del tema elegido**

Hoy en día, con tanta gente usando servicios bancarios en línea, es súper importante que las plataformas financieras puedan manejar muchas operaciones al mismo tiempo sin comprometer la seguridad ni la integridad de los datos.

Este proyecto nos ayudará a entender, de manera práctica, los retos de la programación concurrente y la sincronización de recursos compartidos, que son clave en aplicaciones críticas como los sistemas bancarios.

Simular transacciones bancarias concurrentes ofrece un escenario realista y educativo para explorar conceptos como el paralelismo, la sincronización de hilos, la consistencia de datos y la medición del rendimiento.

**Objetivos**

**General**

Desarrollar una simulación de un sistema de transacciones bancarias en línea que ejecute transacciones paralelas, con el fin de analizar el comportamiento del sistema ante múltiples operaciones concurrentes, garantizando la integridad de los datos y evaluando su rendimiento y escalabilidad.

**Específicos**

1. Implementar transacciones bancarias concurrentes utilizando la Task Parallel Library (TLP).
2. Aplicar estructuras de datos seguras como diccionarios concurrentes para el manejo compartido de cuentas bancarias.
3. Evaluar diferentes técnicas de sincronización, como el lock e interlocked, para mantener la integridad de los datos.
4. Simular escenarios realistas con múltiples clientes y transacciones simultáneas.
5. Medir métricas de rendimiento como el tiempo de respuesta y la precisión de los saldos finales.
6. Analizar la escalabilidad del sistema al incrementar la carga de usuarios y operaciones.

**2. Descripción del Problema**

**Contexto del problema**

En los sistemas bancarios modernos, especialmente en los que funcionan en línea, miles de usuarios realizan transacciones simultáneamente: transferencias, pagos, depósitos, retiros, entre otras. Estas operaciones deben ser procesadas de manera rápida, precisa y segura, sin que se presenten inconsistencias como pérdida de datos, duplicación de operaciones o saldos incorrectos.

El manejo incorrecto de múltiples transacciones concurrentes puede generar serios problemas como transacciones fallidas, bloqueos de cuentas o incluso fraudes. Por tanto, es esencial que los sistemas bancarios sean capaces de gestionar la concurrencia de manera eficaz, garantizando la integridad de los datos aún bajo condiciones de alta demanda.

**Aplicación del problema en un escenario real**

Imaginemos un banco en línea en el que 1,000 usuarios están intentando transferir dinero al mismo tiempo. Cada uno accede a su cuenta, realiza una transferencia y espera una confirmación inmediata. Si el sistema no es capaz de procesar estas solicitudes en paralelo correctamente, podrían presentarse errores como:

* Saldos mal calculados.
* Transacciones duplicadas o perdidas.
* Condiciones de carrera.
* Colisiones en la base de datos.

Un sistema mal diseñado podría colapsar o causar pérdidas financieras, lo cual afectaría directamente la confianza del cliente y la reputación de la entidad bancaria.

**Importancia del paralelismo en la solución**

El paralelismo es clave para resolver este problema. A través de la ejecución simultánea de múltiples hilos de trabajo, se puede lograr que el sistema procese un alto volumen de transacciones de manera eficiente, reduciendo el tiempo de respuesta y mejorando la experiencia del usuario.

Sin embargo, el uso del paralelismo también introduce nuevos retos: el acceso concurrente a datos compartidos puede causar errores si no se implementan mecanismos adecuados de sincronización, como lock, interlocked o estructuras thread-safe como los diccionarios concurrentes.

Este proyecto permite demostrar cómo aplicar correctamente estas técnicas para garantizar que las transacciones sean atómicas, los datos estén consistentes, y el sistema sea escalable y robusto ante el aumento de usuarios.