

**CICLO FORMATIVO DE GRADO SUPERIOR-TÉCNICO EN  
ADMINISTRACIÓN DE SISTEMAS INFORMÁTICOS EN REDES**

**ADMINISTRACIÓN DE SISTEMAS OPERATIVOS**

**Nombre y apellidos:**

- Borja Torres Hernández y Diego Pastrana Monzón

**Tarea 2.1. Estudio de Caso: Problemas de Concurrencia y Sincronización de Procesos**

**Una empresa de servicios financieros ha implementado un sistema de procesamiento de transacciones en tiempo real que maneja datos financieros críticos, como cuentas bancarias, pagos y transferencias entre clientes. En este sistema, existen tres tipos de procesos:**

**1. Procesos de Validación de Transacciones**

- Estos procesos verifican la validez de cada transacción financiera antes de su aprobación.
- Requieren acceso de lectura/escritura a un registro compartido donde se almacenan las transacciones pendientes y sus detalles.
- Si dos procesos de validación acceden simultáneamente a este registro, existe un riesgo de que se produzcan condiciones de carrera, lo que podría provocar que las transacciones se aprueben o rechacen incorrectamente.

**2. Procesos de Cálculo de Saldos**

- Estos procesos calculan y actualizan el saldo de las cuentas de los clientes en función de las transacciones aprobadas.
- Solo pueden acceder al registro de transacciones aprobadas después de que los procesos de validación terminen su ejecución.
- Debido a la naturaleza secuencial de este acceso, es crítico implementar un mecanismo de sincronización para que los procesos de cálculo de saldos no intenten acceder a los datos hasta que los procesos de validación hayan terminado.

**3. Procesos de Generación de Reportes**

- Estos procesos generan reportes periódicos y estadísticas financieras a partir de las transacciones registradas y los saldos de los clientes.
- Realizan operaciones intensivas de lectura, por lo que es fundamental que no bloqueen los procesos de validación o cálculo de saldos.
- Requieren acceso en paralelo a los datos sin interferir con otros procesos que necesiten modificar estos datos.

**Debido al crecimiento del volumen de transacciones, el sistema ha comenzado a experimentar problemas de rendimiento y fallas esporádicas, en las cuales el acceso simultáneo de varios procesos está provocando inconsistencias y errores en los datos financieros. Como administrador de sistemas, se te ha asignado la tarea de proponer un mecanismo de sincronización adecuado para resolver estos problemas y asegurar un flujo ordenado y seguro en el procesamiento de transacciones.**

## **Actividad:**

### **1. Identificar los Problemas de Concurrency**

- **Describe al menos dos problemas de concurrencia específicos que podrían surgir en este sistema debido al acceso simultáneo de los procesos a los recursos compartidos. En particular, menciona las condiciones de exclusión mutua y los problemas de inanición e interbloqueo.**
- **Explica cómo estos problemas podrían afectar la estabilidad y el rendimiento del sistema.**

## **Condiciones de Carrera:**

Se dan cuando dos o más procesos que trabajan juntos comparten datos que pueden ser modificados. El problema se debe a que el resultado final depende del orden en que se ejecuten esos procesos, lo que puede llevar a conflictos inesperados.

## **Interbloqueo:**

Se produce donde dos o más procesos quedan bloqueados porque están esperando que el otro libere los recursos que necesitan para continuar.

Características del interbloqueo:

**-Bloqueo Mutuo:** Cada proceso tiene un recurso que necesita el otro para seguir, y ambos se quedan bloqueados porque no pueden ceder lo que el otro necesita.

**-Espera Circular:** Se forma una cadena donde cada proceso espera que el siguiente libere un recurso, creando un bucle sin salida.

**-Recursos No Liberados:** Los recursos quedan bloqueados porque no se pueden liberar debido a errores de programación, mala sincronización o problemas en la planificación.

## **Efectos:**

**-Estabilidad:** Las condiciones de carrera pueden causar errores en datos importantes, como los financieros, comprometiendo la integridad en el sistema.

**-Rendimiento:** Los interbloqueos pueden detener procesos críticos, reduciendo la capacidad del sistema para responder en tiempo real.

## 2. Investigar Mecanismos de Sincronización

- Investiga y elige al menos dos mecanismos de sincronización, como semáforos, mutexes, monitores, o barreras, y explica cómo cada uno ayudaría a prevenir problemas de contención en este contexto.
- Justifica por qué cada mecanismo es adecuado para resolver los problemas identificados en cada tipo de proceso (validación, cálculo de saldos y generación de reportes) en el sistema financiero.
- Infórmate de cómo se redacta un pseudocódigo para la representación de soluciones por software.

### Métodos de Sincronización:

**Semáforos:** Son herramientas que controlan el acceso a recursos compartidos usando operaciones simples como wait y signal para evitar que varios procesos accedan al mismo recurso al mismo tiempo.

#### Aplicación:

**-Procesos de Validación:** Garantizan que solo un proceso pueda modificar el registro de transacciones que están pendientes a la vez.

**-Procesos de Cálculo de Salos:** Aseguran que estos procesos empiecen únicamente cuando las validaciones hayan finalizado, usando semáforos para coordinarse.

**Monitores:** Son estructuras de datos avanzadas que combinan datos compartidos con métodos sincronizados, permitiendo un acceso seguro y ordenado, incluso en operaciones paralelas. Sus métodos se ejecutan con exclusión mutua, esto significa que un hilo puede cualquiera de sus métodos

**Aplicación:** Son capaces de realizar procesos de generación de reportes. Permiten que varios procesos lean los datos al mismo tiempo sin afectar las validaciones o los cálculos de saldos, manteniendo toda la información organizada. Una vez leído todos los datos, genera los reportes correspondientes.

#### Justificación:

**-Semáforos:** Es un método de sincronización efectiva para evitar conflictos entre procesos al compartir recursos.

**-Monitores:** Ideales para escenarios con muchas lecturas paralelas, asegurando la seguridad y consistencia de los datos.

### 3. Implementación Práctica

- o **Desarrolla los algoritmos de pseudocódigo de cómo se ejecutarían los diferentes procesos, los mecanismos de sincronización que propones para gestionar el acceso a un recurso compartido.**
- o **Explica por qué elegiste este mecanismo para el caso específico y cómo tu solución permitirá que cada tipo de proceso acceda a los recursos sin causar inconsistencias o bloqueos. Describe los beneficios que aporta la solución en términos de seguridad y rendimiento del sistema.**

#### Procesos de Validación

semaphore mutex = 1

```
procedure validarTransaccion(transaccion):  
    wait(mutex) # Exclusión mutua  
    leerRegistroCompartido(transaccion)  
    verificarValidez(transaccion)  
    escribirRegistroCompartido(transaccion)  
    signal(mutex)
```

**Mecanismo:** Semáforos binarios garantizan que solo un proceso pueda acceder al registro compartido a la vez, eliminando condiciones de carrera.

**Beneficio:** Consistencia en la aprobación o rechazo de transacciones.

#### Procesos de Cálculo de Saldos

semaphore validacionCompleta = 0

```
procedure calcularSaldos():  
    wait(validacionCompleta)  
    leerRegistroTransaccionesAprobadas()  
    actualizarSaldos()  
    signal(validacionCompleta)
```

**Mecanismo:** Semáforos contadores aseguran que los procesos de cálculo de saldos no accedan al registro hasta que los procesos de validación hayan completado su trabajo.

**Beneficio:** Sincronización secuencial que evita errores en los saldos.

## **Procesos de Generación de Reportes**

```
monitor GeneracionReportes
    shared data registros

    procedure generarReporte():
        accesoLectura(registros)
        generarEstadisticas()
        liberarAcceso()
    end monitor
```

**Mecanismo:** Monitores permiten el acceso concurrente para lecturas sin interferir con procesos de escritura.

**Beneficio:** Operaciones de lectura eficientes y consistentes, mejorando el rendimiento general del sistema.

### **Bibliografía:**

Tanenbaum, A. S. (2014). Modern Operating Systems. Pearson.

Silberschatz, A., Galvin, P. B., & Gagne, G. (2020). Operating System Concepts. Wiley.

Comunidad ESCOM. (n.d.). Principios de la sincronización. Recuperado de [https://www.comunidad.escom.ipn.mx/israelr/RDD\\_SO\\_U2/pages/2-4-1-principios-de-la-sincronizacion.html](https://www.comunidad.escom.ipn.mx/israelr/RDD_SO_U2/pages/2-4-1-principios-de-la-sincronizacion.html)

Departamento de Ingeniería de Sistemas y Automática. (s.f.). *Sistemas operativos de tiempo real: Mecanismos de comunicación y sincronización entre procesos*. Universidad de Valladolid.

Entregar un documento en formato pdf con la realización de los tres apartados, incluyendo los nombres de todos los miembros del grupo y la bibliografía consultada.