Poster Final

Implementación en Go-lang:

- Empleamos Go-lang para la programación del sistema, lo cual consideramos una excelente elección. Go-lang ofrece un modelo de concurrencia sólido y eficiente, con soporte nativo para goroutines y canales, lo que facilita la implementación de sistemas concurrentes como el sistema bancario requerido.

Mecanismos de Sincronización:

Hemos empleado semáforos, mutexes o monitores para la sincronización de hilos. Consideramos que es crucial en entornos concurrentes para garantizar la integridad de los datos y evitar condiciones de carrera. Estos mecanismos son fundamentales en Go-lang y proporcionan las herramientas necesarias para coordinar el acceso a recursos compartidos de manera segura.

Prevención de Deadlock:

Hemos implementado técnicas para prevenir el deadlock, como la asignación ordenada de recursos o el uso de un sistema de detección de deadlocks. Esto es esencial para mantener la estabilidad del sistema en situaciones de concurrencia. Go-lang ofrece un enfoque robusto para la prevención de deadlocks, con herramientas como el uso cuidadoso de canales y select statements para evitar bloqueos indefinidos.

Consideren el Uso de Canales en Go-lang:

Una alternativa que sugerimos considerar es el uso intensivo de canales en Go-lang. Los canales son una poderosa herramienta de sincronización que permite la comunicación entre goroutines de manera segura y eficiente. Al emplear canales para coordinar la ejecución concurrente, podemos simplificar el código y reducir la complejidad asociada con el uso de mutexes y semáforos.

Eugenio Andrés Mejía Fanjón A01412143
Fernando Monroy Sánchez A01750536