



PONTIFICIA UNIVERSIDAD JAVERIANA
FACULTAD DE INGENIERIA
DEPARTAMENTO INGENIERIA DE SISTEMAS

Introducción a Sistemas Distribuidos – Proyecto 2
Período Académico 2018-1

Entrega y Sustentación 16/Mayo de 2018

Banca Virtual

Objetivo

El objetivo general del proyecto final del curso ISD, además de contribuir a la apropiación de los conceptos de gestión de transacciones distribuidas, se centra en crear un sistema con un conjunto de funciones que permitan implementar aplicaciones distribuidas teniendo manejo transaccional parcial en el contexto de una problemática real. Los objetivos específicos son:

- Poner en práctica conceptos de control de concurrencia en un problema práctico del mundo real.
- Implementar, mediante el uso de RMI, un protocolo de consumación atómica de dos fases en un entorno distribuido.

Para el semestre 2018-1, el proyecto se focalizará la gestión de transacciones aplicándolas al manejo de operaciones bancarias.

Descripción del Sistema a Desarrollar

El proyecto debe producir una implementación de una aplicación distribuida para la gestión de operaciones de un banco virtual. Los clientes acceden desde su computador/dispositivo para realizar consultas de un producto o de todos al mismo tiempo, transferencias entre cuentas (propias y/o de un tercero) y pago de tarjetas de crédito (una a la vez o todas al mismo tiempo). Un cliente puede realizar estas seis operaciones bancarias sobre sus productos financieros que incluyen: tarjetas de crédito (visa, master) y cuentas (ahorro, crédito); todo cliente cuenta con un portafolio que incluye estos cuatro productos.

Se debe tener en cuenta que los objetos que representan los productos del banco deben estar repartidos en múltiples máquinas (servidores); de igual forma los procesos/hilos que intervienen en el procesamiento de las transacciones deben operar en forma distribuida. Por ejemplo, tener un servidor para las cuentas bancarias y otros para cada tipo de tarjeta de crédito.

Los objetos que representan estos productos deben ser accedidos remotamente utilizando RMI y solo disponen de métodos *set()* y *get()*. No existen métodos más complejos que realicen en forma integrada lecturas y escrituras; por ejemplo para incrementar el saldo de una cuenta, primero debe hacerse un *get()* para conocer el saldo, y luego el *set()* con el nuevo valor después calcular el valor ya incrementado. Por tanto, todas las operaciones bancarias disponibles se implementan como secuencias que utilizan solo *get()* y *set()*.

Para provocar mayor concurrencia en la ejecución de las transacciones, los métodos *get()* y *set()* incluyen un retardo interno. El tiempo de este retardo es configurable a través de los parámetros que se pasan al lanzar cada uno de los servidores.

Para asegurar el cumplimiento de las propiedades ACID, cada operación bancaria se debe implementar como una transacción distribuida. La gestión de la transacción debe incluir la puesta en operación de un mecanismo de *control de concurrencia optimista con validación hacia adelante* y debe seguir el protocolo de *commit* de 2 fases 2PC.

Para la validación y evaluación del sistema se deben construir clientes que operen en un ambiente distribuido. Los clientes pueden solicitar en forma manual o automática la realización de las seis operaciones bancarias. En un cliente manual, un menú sencillo permite seleccionar la operación y luego especificar los parámetros requeridos según el tipo de operación; en todos los casos se deben suministrar el/los identificadores de las cuentas sobre la que se realiza la operación bancaria. Para facilitar el uso del cliente, en el menú también se debe incluir una opción para especificar el número de cuenta del cliente que está realizando las operaciones (simulando un inicio de sesión, pero sin autenticación real). En los clientes automáticos, se genera una secuencia de al menos 10 operaciones generadas a partir de la lectura de un archivo; cada operación se solicita con una diferencia de tiempo configurable a través del menú del cliente.

Equipos de Trabajo

El proyecto se realizará en grupos de trabajo mínimo 2 personas y máximo 3 personas.

Entrega y Condiciones

La entrega se realizará el día de la sustentación, el miércoles 16/Mayo a las 9am. La entrega se debe hacer a través de UVirtual, siguiendo las directrices de entrega dadas en el foro creado para tal fin. Además del código, se debe adicionar la documentación correspondiente con el esquema de procesos y sus interacciones, el diagrama UML para identificar las clases utilizadas y la descripción general de los componentes principales. Igualmente, se debe presentar el protocolo de pruebas y el análisis de los resultados obtenidos.

No se considera documentación al código fuente, pero si se espera que esté bien estructurado y sea de fácil comprensión. No se requiere hacer interfaces gráficas sofisticadas, pero sí de fácil uso y que permitan verificar el correcto funcionamiento del sistema distribuido corriendo en al menos 3 máquinas de forma ágil y flexible. Se debe hacer la sustentación con el código que se entrega y deben estar presentes todos los integrantes del grupo.

No puede existir replicación de código entre grupos, lo cual se consideraría plagio.