



**PONTIFICIA UNIVERSIDAD JAVERIANA**  
**FACULTAD DE INGENIERIA**  
**DEPARTAMENTO INGENIERIA DE SISTEMAS**

**Introducción a Sistemas Distribuidos**  
**Período Académico 2019-10**

Bogotá, abril 10 de 2019

**PROYECTO 2**

Entrega : **martes 22 de mayo de 2019**

Sustentación : **martes 22 de mayo de 2019**

**Objetivos**

El objetivo del proyecto final consiste en crear un conjunto de funciones que permita implementar una aplicación distribuida con manejo transaccional parcial. Las propiedades transaccionales que deben implementar son: atomicidad por medio de un protocolo de consumación atómica de dos fases y control de concurrencia utilizando un protocolo de control de concurrencia optimista con validación hacia adelante.

**Características a desarrollar de la Plataforma.**

**Contexto**

El gobierno nacional ha decidido crear una cuenta de ahorro asociada con una tarjeta de consumo para población en condición de vulnerabilidad, en donde consigna mensualmente una cantidad determinada de dinero. Si el usuario de la tarjeta así lo decide, puede también ingresar dinero a su cuenta. Los beneficiarios de la tarjeta pueden acceder a una tienda virtual en la que podrá hacer compras a precios especiales.

**Control de Concurrencia de Transacciones Distribuidas**

La primera vez que un cliente ingresa al sistema, debe generar una contraseña nueva, se genera el hash correspondiente y se almacena de forma permanente, con el fin de hacer validaciones posteriores.

El cliente, para interactuar con la máquina que tiene la lógica de la tienda virtual contacta “siempre” vía RMI a un gestor de control de concurrencia que maneja la consistencia de los recursos.

El gestor de concurrencia maneja toda la información de los productos que hay en la tienda virtual, así como información transaccional cuando uno o más usuarios ingresan a la plataforma, pero puede presentarse el inconveniente, cuando de forma concurrente varios clientes piden el mismo producto y no hay existencias suficientes.

Al inicio el cliente abre la transacción identificándose con nombre de usuario, número de tarjeta y clave, durante la transacción se seleccionan productos y cantidades, además durante el proceso de compra, el usuario tiene la capacidad de eliminar productos seleccionados y cambiar cantidades y finaliza la transacción solo hasta cuando confirma la compra o termina la sesión sin realizar la transacción.

Se debe validar que la tarjeta tiene saldo suficiente para realizar las compras y que hay existencia suficiente de productos para responder a la solicitud. Se podría dar el caso de que justo en el momento en que se está realizando una transacción, ingresa dinero a la cuenta.

## **2 Control de consumación atómica de dos fases**

Teniendo en cuenta que en una transacción posiblemente se trabajan múltiples productos y que pueden existir múltiples transacciones, es probable que se presenten problemas de concurrencia: por lo tanto, se utilizan los protocolos de consumación atómica para indicar de manera consiste que: o se hacen todos los cambios (las copias se convierten en consumadas) o no se hace ninguno y, además, las operaciones de diferentes transacciones no interfieren entre sí.

Teniendo en cuenta que en una transacción se trabajan múltiples productos, es posible que se presenten problemas de concurrencia en algunos casos; por lo tanto, se debe utilizar un protocolo de **control optimista de concurrencia con validación hacia atrás**.

Recuperación: Cuando el servidor deja de funcionar y posteriormente entra en operación, la reconexión debe hacerse de forma automática, es decir, será transparente para el cliente; las transacciones deben continuar en el punto en que se habían dejado y no iniciar desde el principio.

Modificación de pedido: Un cliente podrá modificar el pedido cuantas veces desee, antes de confirmar las operaciones de la transacción.

Concurrencia (lectura/escritura): Se deben generar todas las copias tentativas de los objetos que están siendo utilizados por los diferentes usuarios y las operaciones correspondientes

Durabilidad: El estado de los productos debe quedar guardado de forma permanente cuando se confirman las transacciones. Igualmente, los usuarios registrados y los *hashes* correspondientes a sus contraseñas deben mantenerse en memoria permanente, de manera que la información debe quedar almacenada y cifrada con un algoritmo simétrico (3DES, AES) para preservar la confidencialidad, aun cuando la aplicación se cierre.

## **Equipos de Trabajo.**

El proyecto se realizará en grupos de trabajo de máximo de tres personas

## **Entrega y condiciones.**

La entrega se realizará el día de la sustentación (martes 22 de mayo). La entrega se debe hacer en medio magnético. Además del código se debe adicionar la documentación correspondiente con el diagrama de clases y la descripción general de cada componente desarrollado. Igualmente se debe presentar el **protocolo de pruebas**. No se considera documentación al código fuente. 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.**

<b>Grupo</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>
RMI (10%)					
Creación de cuenta y de usuario (5%)					
Gestor de concurrencia (15%)					
Consumación atómica (5%)					
Estado de las Tx (lecturas, escrituras) (10%)					
Validación hacia atrás (15%)					
Recuperación de Tx (10%)					
Durabilidad de las transacciones (Usuarios, hashes, productos) 10%					
Cifrado de la información almacenada (10%)					
Documentación (10%)					