

**PONTIFICIA UNIVERSIDAD JAVERIANA**

**FACULTAD DE INGENIERIA**

**DEPARTAMENTO INGENIERIA DE SISTEMAS**

**Proyecto Introducción a Sistemas Distribuidos**

**Período Académico 2020-3**

**Entrega y Sustentación - 29 y 30 de septiembre de 2020**

**Sistema de Citas médicas**

**Objetivos Generales**

• Poner en práctica conceptos de sistemas distribuidos en un problema práctico del mundo real.

• Implementar, mediante el uso de RMI, una aplicación funcional.

• Afianzar los conceptos de control de concurrencia utilizando uno de los métodos principales.

**Objetivo Específico**

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 atrás.

**Descripción del Sistema a Desarrollar**

La distribución de la vacuna contra el COVID 19 para Colombia estará a cargo de 3 diferentes laboratorios, cada uno como generador de su propia vacuna (Vac1, vac2, y vac3). Las entidades encargadas de centralizar la recepción de las vacunas para el país son las Instituciones Prestadoras de Salud (IPS), se considerarán únicamente dos IPS.

Las Entidades Prestadoras de Salud (EPS) deben hacer las solicitudes de vacunas correspondientes para aplicárselas a sus usuarios de acuerdo con las solicitudes diarias. Teniendo en cuenta que la demanda es variable y que cada EPS tiene diferentes afiliados, se harán pedidos de acuerdo con la demanda diaria generada por los usuarios, quienes están en libertad de elegir la vacuna que desean que se les aplique (Vac1, Vac2 o Vac3). Una transacción estará conformada por la demanda continua de los usuarios a la EPS, de manera que las solicitudes por cada tipo de vacuna pueden variar. Se puede considerar un mínimo de 10 solicitudes de vacunas en total para generar una transacción (Petición de la EPS a la IPS). Las transacciones se generan de forma continua a lo largo del día por parte de las 3 EPS. A continuación, se presenta un ejemplo:

Transacción 1 Transacción 2 Transacción 3

**EPS 1**

* 3 unidades de la Vac1
* 12 unidades de la *Vac2*
* 4 unidades de la *Vac3*

**EPS 2**

* 8 unidades de la Vac1
* 2 unidades de la *Vac2*
* 7 unidades de la *Vac3*

**EPS 3**

* 15 unidades de la Vac1
* 4 unidades de la *Vac2*
* 6 unidades de la *Vac3*

Los usuarios de cada EPS harán la solicitud de la vacuna de su preferencia, este procedimiento se debe realizar de forma **automática**, por ejemplo, leyendo de archivos de texto con diferentes usuarios, cada usuario solicitando el tipo de vacuna de su preferencia. Posteriormente, las EPS generan las transacciones (mínimo 10 solicitudes de vacunas) y realizarán operaciones de lectura para confirmar que hay existencias suficientes de cada vacuna y posteriormente se generarán las escrituras correspondientes. En caso de que no haya existencias de alguna de las 3 vacunas, se debe hacer la solicitud de las dos restantes garantizando la **atomicidad** de la transacción (se ejecutan todas las operaciones o no se ejecuta ninguna).

El control de concurrencia se hará utilizando un método optimista con validación hacia atrás teniendo en cuenta las reglas de conflicto.

**Control de Concurrencia de Transacciones Distribuidas**

La primera vez que una EPS ingresa al sistema, debe crear su perfil con un nombre de usuario y una contraseña, se genera el hash correspondiente (SHA) y se almacena de forma permanente, con el fin de hacer validaciones posteriores.

La EPS, para interactuar con la máquina que tiene la lógica de las IPS, 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 las vacunas disponibles que hay en las IPS, así como información transaccional cuando una o más EPS ingresan a la plataforma, pero puede presentarse el inconveniente, cuando de forma concurrente varias EPS piden el mismo producto y no hay existencias suficientes para todos.

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

**Recuperación**: Cuando un servidor de la IPS deja de funcionar, debe asumir la carga el servidor homólogo. Cuando el servidor caído y entra en operación, la reconexión y distribución de cargas 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.

**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, es decir, si se cierra la aplicación se debe guardar el registro de las operaciones realizadas. Igualmente, las EPS registradas 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.

**PRIMERA ENTREGA**

* Se debe utilizar el modelo de vistas de arquitectura 4+1 ([*https://es.wikipedia.org/wiki/Modelo\_de\_Vistas\_de\_Arquitectura\_4%2B1*](https://es.wikipedia.org/wiki/Modelo_de_Vistas_de_Arquitectura_4%2B1)) para describir la arquitectura del sistema y se deben generar los entregables correspondientes de cada vista (Lógica, de desarrollo, física, de procesos y escenarios);
* Especificación y análisis de requisitos
* Documento de diseño
* Protocolo de pruebas
* **Análisis** de la metodología de desarrollo de software (Espiral, cascada, etc.) y metodología elegida.
* Modelos fundamentales (Interacción, fallos, seguridad)

La interacción entre las entidades se podría ver de la siguiente manera:

Paciente 1-1

Paciente 1-2

Paciente 1-n

EPS 1

EPS 2

EPS N

Datos

IPS 1

IPS 2

Paciente 1-1

Paciente 1-2

Paciente 1-n

Paciente 1-1

Paciente 1-2

Paciente 1-n

GCC

Se debe contar con un mínimo de tres máquinas comunicándose entre sí, los dos servidores IPS1 e IPS2 deben estar implementados en máquinas físicamente separadas, así como los clientes y el servidor.

#### 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 29 y miércoles 30 de septiembre) a través de la plataforma.

Para la segunda entrega se debe hacer la sustentación mostrando la funcionalidad del proyecto con la documentación correspondiente, se debe complementar la documentación de la primera entrega y deben estar presentes todos los integrantes del grupo.

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