**Informe: Sistema de Gestión de Estaciones de Servicio TerMax**

**Brayan Camilo Silva Porras -** 1100972730

**José Vicente Zabaleta Montiel -** 1063309664

**Análisis del Problema y Consideraciones para la Solución**

**Contexto del Problema**

TerMax es una empresa líder en el suministro de combustibles en Colombia, con una amplia red de estaciones de servicio distribuidas por todo el país. La compañía necesita un sistema de gestión eficiente para administrar su red de estaciones, controlar las ventas de combustible y monitorear el inventario en tiempo real.

**Requerimientos Principales**

1. Gestión de la red nacional de estaciones de servicio
2. Administración individual de cada estación de servicio
3. Control de surtidores y ventas de combustible
4. Monitoreo de inventario y detección de fugas
5. Simulación de ventas para pruebas y proyecciones

**Consideraciones para la Solución**

* Escalabilidad: El sistema debe ser capaz de manejar un gran número de estaciones de servicio y transacciones.
* Modularidad: Diseño que permita fácil mantenimiento y expansión futura.
* Eficiencia: Operaciones optimizadas para manejar grandes volúmenes de datos.
* Seguridad: Protección de datos sensibles como información de ventas e inventario.

**2. Diagrama de Clases**

Aquí incluiríamos el diagrama de clases UML. Para este informe, describiremos las clases principales en texto.

**Clases Principales:**

1. RedNacional: Representa la red completa de estaciones de servicio.
2. EstacionServicio: Modela una estación individual con sus propiedades y operaciones.
3. Surtidor: Representa un surtidor de combustible en una estación.
4. Tanque: Modela el tanque de almacenamiento de combustible de una estación.

**4. Problemas de Desarrollo Afrontados**

1. Manejo de concurrencia: Asegurar que múltiples ventas simultáneas no causen inconsistencias en el inventario.
2. Optimización de consultas: Mejorar el rendimiento al calcular estadísticas de ventas para toda la red.
3. Diseño de la interfaz de usuario: Crear un menú intuitivo que permita acceder a todas las funcionalidades requeridas.

**5. Evolución de la Solución y Consideraciones para la Implementación**

1. Inicialmente, consideramos usar una base de datos relacional, pero optamos por una solución en memoria para mayor velocidad en las operaciones frecuentes.
2. Implementamos un patrón Observer para notificar cambios en el nivel de combustible de los tanques.
3. Utilizamos el patrón Singleton para la clase RedNacional para asegurar una única instancia en todo el sistema.

Diagrama

Descripción generada automáticamente