# Análisis del problema y consideraciones para la alternativa de

# solución propuesta:

# Contextualización:

TerMax es una empresa líder en suministro de combustible en Colombia, con una amplia red de estaciones de servicio en todo el país. El desafío consiste en desarrollar un sistema de gestión para esta red utilizando Programación Orientada a Objetos (POO) en C++

# Funcionalidades Principales:

**Gestión de la Red**

Agregar/Eliminar estaciones de servicio

Calcular ventas totales por estación y categoría de combustible

Fijar precios de combustible para toda la red

**Gestión de Estaciones de Servicio**

Agregar/Eliminar surtidores

Activar/Desactivar surtidores

Consultar historial de transacciones de cada surtidor

Reportar ventas de combustible por categoría

Simular ventas de combustible

Asignar capacidad del tanque (aleatoriamente entre 100-200 litros por categoría)

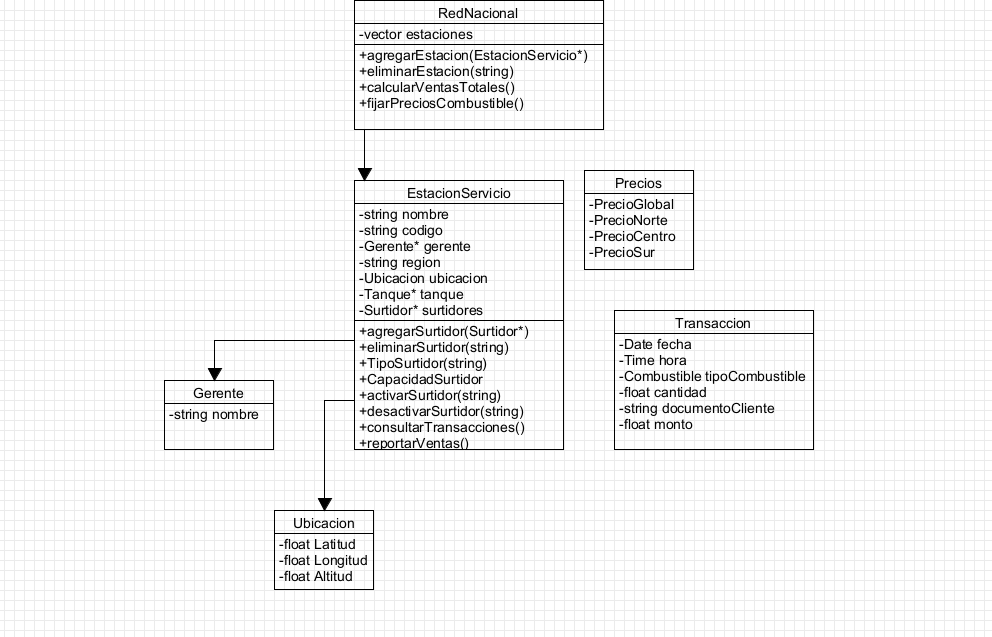
**Sistema de Detección de Fugas**

Verificar la integridad del combustible para cualquier estación de la red

**Simulación de Ventas**

Asignar aleatoriamente un surtidor activo para la transacción

Simular ventas de 3-20 litros aleatoriamente.



# Descripción de la Lógica de Clase:

#### RedNacional

Esta clase representa la red completa de estaciones de servicio TerMax a nivel nacional.

- estaciones: Almacena todas las estaciones de servicio de la red.

- agregarEstacion(EstacionServicio): Añade una nueva estación a la red.

- eliminarEstacion(string): Elimina una estación de la red basándose en su identificador.

- calcularVentas(Fecha): Calcula las ventas totales de todas las estaciones para una fecha específica.

-fijarPreciosCombustible(): Actualiza los precios del combustible para toda la red.

**Lógica principal:**

- Gestiona la colección de todas las estaciones de servicio.

- Proporciona métodos para agregar o eliminar estaciones.

- Realiza cálculos y operaciones a nivel de toda la red.

#### EstacionServicio

Representa una estación de servicio individual dentro de la red.

- Atributos: nombre, código, gerente, región, ubicación, tanque, surtidores.

- agregarSurtidor(Surtidor): Añade un nuevo surtidor a la estación.

- eliminarSurtidor(string): Elimina un surtidor específico.

- tipoSurtidor(string): Devuelve el tipo de un surtidor dado.

- capacidadSurtidor(): Gestiona la capacidad de los surtidores.

- activarSurtidor(string): Activa un surtidor específico.

- desactivarSurtidor(string): Desactiva un surtidor específico.

- consultarTransacciones(): Obtiene el historial de transacciones.

- reportarVentas(): Genera un reporte de ventas de la estación.

**Lógica principal:**

- Gestiona los surtidores y el tanque de la estación.

- Maneja las operaciones diarias como activar/desactivar surtidores.

- Lleva un registro de las transacciones y genera reportes.

#### Gerente

Representa al gerente de una estación de servicio.

- string nombre: Almacena el nombre del gerente.

**Lógica principal:**

- Aunque simple en el diagrama, podría expandirse para incluir responsabilidades de gestión y toma de decisiones para la estación.

#### Ubicacion

ubicación geográfica de una estación de servicio.

- float Latitud: Coordenada de latitud.

- float Longitud: Coordenada de longitud.

- float Altitud: Altitud de la ubicación.

**Lógica principal:**

- Proporciona datos precisos de localización para cada estación.

- Podría utilizarse para cálculos de distancia o agrupación regional.

#### Precios

Clase que maneja los precios del combustible según la región.

- PrecioGlobal: Precio base para todo el país.

- PrecioNorte: Precio específico para la región norte.

- PrecioCentro: Precio específico para la región centro.

- PrecioSur: Precio específico para la región sur.

**Lógica principal:**

- Gestiona la variación de precios por región.

- Podría incluir métodos para actualizar precios y calcular diferencias regionales.

#### Transacción

Representa una transacción individual de venta de combustible.

- Date fecha: Fecha de la transacción.

- Time hora: Hora de la transacción.

- string tipoCombustible: Tipo de combustible vendido.

- float cantidad: Cantidad de combustible vendido.

- string documentoCliente: Identificación del cliente.

- float monto: Monto total de la venta.

**Lógica principal:**

- Registra los detalles de cada venta de combustible.

- Podría incluir métodos para calcular el monto basado en la cantidad y el tipo de combustible.

- Útil para generar reportes y análisis de ventas.