**Diseño de la Solución para el Sistema de Gestión de Estaciones de Gasolina**

**1. Objetivo del Sistema**

El objetivo principal de este sistema es gestionar una red de estaciones de gasolina distribuidas en diferentes zonas geográficas (norte, centro y sur). Cada estación tiene características específicas como su número, nombre, red a la que pertenece, y los precios de tres tipos de gasolina (Regular, Premium, EcoExtra). El sistema debe permitir agregar nuevas estaciones, registrar ventas y consultar información de cada estación de forma interactiva, con los datos almacenados en archivos de texto para persistencia.

**2. Componentes Principales del Sistema**

El sistema está dividido en varias clases y archivos, que se detallan a continuación:

* **Clase Estacion**:
  + Representa cada estación de gasolina.
  + Atributos:
    - numero: Identificador único de la estación.
    - nombre: Nombre de la estación.
    - red: Zona geográfica a la que pertenece (norte, centro o sur).
    - precioRegular, precioPremium, precioDiesel: Precios de los diferentes tipos de gasolina.
  + Métodos:
    - registrarVenta(): Registra una venta de un tipo específico de gasolina.
    - setPrecios(): Permite actualizar los precios de las gasolinas.
    - toString(): Convierte la información de la estación en una cadena de texto para facilitar la escritura en archivos.
  + Persistencia: Los datos de cada estación se guardan y leen desde archivos de texto.
* **Clase Red**:
  + Agrupa un conjunto de estaciones pertenecientes a una zona (norte, centro o sur).
  + Atributos:
    - estaciones: Un arreglo de estaciones con un límite predefinido.
    - nombreArchivo: Nombre del archivo donde se almacenan las estaciones de la red.
    - numEstaciones: Cantidad de estaciones actualmente en la red.
  + Métodos:
    - cargarDesdeArchivo(): Lee las estaciones desde un archivo de texto y las carga en memoria.
    - guardarEnArchivo(): Guarda las estaciones de la red en el archivo de texto.
    - agregarEstacion(): Añade una nueva estación a la red y actualiza el archivo.
    - mostrarEstaciones(): Muestra la lista de estaciones de la red en la consola.
* **Archivo main.cpp**:
  + Maneja la interacción con el usuario mediante un menú.
  + Permite seleccionar la red a gestionar y realizar acciones como agregar estaciones, registrar ventas y consultar información.
  + Los datos se obtienen de la entrada del usuario y se guardan en los archivos de texto correspondientes a cada red.

**3. Flujo de Trabajo del Sistema**

1. **Inicialización**:
   * Al iniciar el programa, cada objeto Red (para norte, centro y sur) carga sus estaciones desde su archivo de texto correspondiente utilizando el método cargarDesdeArchivo().
   * Esto permite que la información de las estaciones esté disponible en memoria para las acciones posteriores.
2. **Agregar una Estación**:
   * El usuario selecciona la red (norte, centro o sur) donde desea agregar una nueva estación.
   * Se le solicita al usuario el número de la estación, su nombre, y los precios de los tres tipos de gasolina.
   * Se crea un objeto Estacion con esta información y se añade a la red seleccionada mediante el método agregarEstacion().
   * Inmediatamente, se actualiza el archivo de la red para asegurar que la información persista.
3. **Registrar Ventas de Gasolina**:
   * El usuario selecciona una red y una estación específica dentro de esa red.
   * Se le pide el tipo de gasolina y la cantidad a vender.
   * El sistema actualiza la información de ventas utilizando el método registrarVenta() de la clase Estacion.
4. **Consulta de Estaciones**:
   * El usuario puede solicitar ver las estaciones de una red específica.
   * El sistema muestra la información de todas las estaciones de la red seleccionada utilizando el método mostrarEstaciones().
   * Esto incluye detalles como el número de estación, nombre, red, y los precios de cada tipo de gasolina.
5. **Guardar y Cargar desde Archivos**:
   * La información de cada red se guarda en archivos de texto específicos, permitiendo la persistencia de datos.
   * Al cerrar y abrir el programa, los datos se recargan para mantener la continuidad.

**4. Formato de Almacenamiento en Archivos de Texto**

* Cada red tiene un archivo de texto independiente para almacenar las estaciones.
* Cada línea en el archivo representa una estación, con los atributos separados por el delimitador |:

numero|nombre|red|precioRegular|precioPremium|precioEcoExtra

* Ejemplo de una línea:

1|Estacion Centro 1|Centro|5000|5000|7000

**5. Pasos a Completar la Implementación**

1. Completar la implementación de los métodos de la clase Estacion y Red para la gestión de precios y ventas.
2. Implementar la carga y guardado de datos en el formato adecuado desde y hacia archivos de texto.
3. Integrar el flujo de interacción en main.cpp para que el usuario pueda realizar todas las operaciones de forma interactiva.
4. Realizar pruebas para verificar que los datos se gestionen y se persistan correctamente.
5. Hacer el detector de fugas.
6. Realizar la administración de la gasolina en las estaciones