

UNIVERSIDAD DE ANTIOQUIA

FACULTAD DE INGENIERIA

INFORMATICA II  
  
  
PRESENTADO POR:

Cristopher corrales

Tomas Restrepo

Medellín, Colombia

**Análisis del problema:**

Para el ejercicio planteado propone hacer una interfaz en la que un usuario pueda crear una red de estaciones de gasolina con distintas opciones como en el siguiente menú:

1. **Gestión de Red:**

**A.**  Agregar Estación.

**B.** Eliminar Estación.

**C.**  Calcular el Monto Total de Ventas.

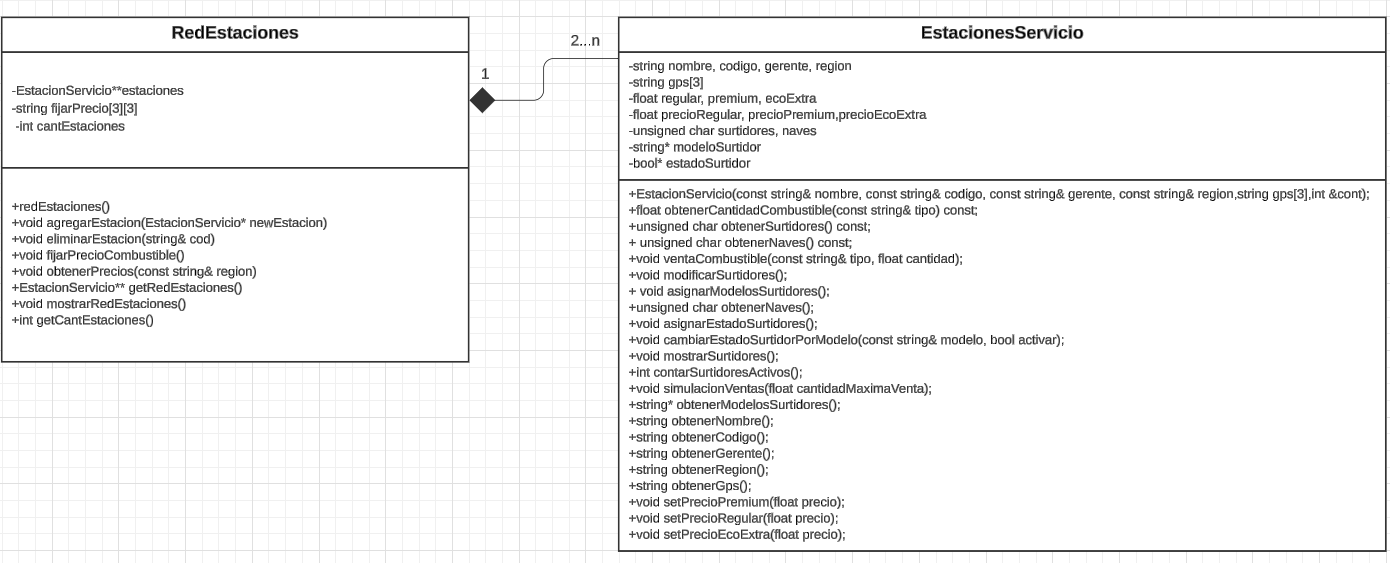
**D.** Mostrar Estaciones.

1. **Gestión de Estaciones de Servicio:**
2. **Sistema Nacional de Verificación de Fugas.**
3. **Simulación de Ventas.**

Las Estaciones de Servicio se manipularán a través de un arreglo dinámico de objetos. Una vez se inicia el programa creamos una estación aleatoria y pediremos que se fijen los precios para las regiones y el tipo de combustible, seguido a esto entraremos en un ciclo mostrando las opciones que tiene el usuario para manipular la Red de Estaciones, este ciclo terminará hasta que el usuario ya no desee seguir modificando la Red Estaciones.

Diagrama

Descripción generada automáticamente



**ALGORITMOS IMPLEMENTADOS:**

En la clase EstacionServicio se usaron los siguientes métodos:

|  |
| --- |
|  |

En la clase RedEstaciones se usaron los siguientes métodos:

|  |
| --- |
| 1. void agregarEstacion (EstacionServicio\* nuevaEstacion): Método que tiene como parámetro nueva estación que es un puntero de tipo EstacionServicio, el cual espera un objeto de EstacionServicio, siguiente a esto se guarda en la redEstaciones. 2. void eliminarEstacion (string& cod): Método que tiene como parámetro la dirección de memoria de un codigo el cual se busca eliminar, este cod se busca en redEstaciones para eliminarlo. 3. void fijarPrecioCombustible (): Método que sirve para fijar el precio del combustible depende la region y tipo de combustible. 4. void obtenerPrecios (const string& region): Método que recibe una region y buscaremos el precio de los tipos de combustible de la region. 5. EstacionServicio\*\* getRedEstaciones (): Método que sirve para retornar las estaciones. 6. void mostrarRedEstaciones (): Método que sirve para mostrar los valores de las estaciones de la red. 7. int getCantEstaciones (): Método que retorna el total de estaciones que hay en la red. |

**PROBLEMAS DURANTE EL DESARROLLO DEL PROGRAMA:**