DESAFÍO II

#### **Análisis del problema**

El problema consiste en gestionar una red nacional de estaciones de servicio de combustibles que incluye múltiples estaciones, cada una con surtidores que distribuyen tres tipos de combustible (Regular, Premium y EcoExtra). El sistema debe permitir la simulación de ventas, la gestión de inventario, la verificación de fugas, y ofrecer un menú para la interacción del usuario con estas funcionalidades.

Las funcionalidades a implementar se dividen en cuatro áreas principales:

1. **Gestión de la red nacional**:Para la administración de las estaciones de servicio, incluyendo agregar, eliminar y calcular ventas.
2. **Gestión de estaciones de servicio**: Se manejan los surtidores, consulta de transacciones, capacidad de tanques y ventas de combustible.
3. **Verificación de fugas**:Se hará la comprobación para detectar posibles pérdidas de combustible.
4. **Simulación de ventas**: Se hará la realización de ventas simuladas, actualizando el combustible disponible y registrando la venta.

#### **Consideraciones para la solución propuesta**

1. **Datos centralizados:** Habrá una única instancia de la red nacional para gestionar todas las estaciones de servicio, lo que permite administrar el sistema de manera eficiente.
2. **Modularidad:** El diseño debe ser modular para separar claramente la lógica de las estaciones de servicio, surtidores y ventas.
3. **Actualización del inventario:** Cada vez que se realice una venta, se debe actualizar el inventario del combustible disponible.
4. **Control de errores:** Se debe manejar las ventas parciales si no hay suficiente combustible disponible.
5. **Gestión automática de códigos:** Se debe implementar un sistema para asignar códigos únicos a cada estación y surtidor.
6. **Verificación de fugas:** Se realizará una comprobación del inventario de combustible para detectar fugas basadas en los datos históricos y actuales.
7. **Variabilidad de precios:** Los precios del combustible deben ser ajustables según la región y deben poder modificarse fácilmente.

### **subprogramas**

1. **Simulación de ventas**
   * Se selecciona aleatoriamente un surtidor activo.
   * Se genera una cantidad aleatoria de litros a vender.
   * Se verifica si hay suficiente combustible disponible, ajustando la venta si es necesario.
   * Se registra la venta y se actualiza la disponibilidad del tanque.
2. **Verificación de fugas de combustible**
   * Se suma la cantidad de combustible vendido y el almacenado en el tanque.
   * Se compara con la capacidad original del tanque para verificar que se mantenga dentro del 95%.
   * Si se detecta una discrepancia, se reporta una posible fuga.

Cordial saludo Profesor.

La presente era para que por favor nos brindara una pequeña asistencia con el trabajo de la red nacional de servicio de gasolina respecto al diagrama de clase que ya organizamos con sus métodos y atributos, la cuestión está en este primer análisis que hicimos pensamos utilizar algunas clases que no se han utilizado o visto clase como lo son map, list y pair por un lado queríamos saber si podríamos utilizarlas, la profesora de laboratorio el día de ayer nos hizo una pequeña introducción sobre cómo funcionan estas clases y de igual forma nos compartió la documentación, pero en lo que vamos de teoría no hemos visto sobre ellas al momento de este desafío quisiéramos saber si podríamos trabajar con estas antes de empezar a implementar y codificar.

Respecto al diagrama de clases analizamos que vamos a utilizar 4 (Red nacional, Estación de servicio, Surtidor y Transacción) De las cuales entre las cosas relevantes tenemos

**La red nacional** :

Pensamos tener una lista en la cual tendremos nuestras estaciones y un mapa donde tendremos una clave string que será el nombre del combustible y un valor tipo float que será el precio del combustible.

Entre los métodos tendremos unos que nos permitirán agregar, eliminar, fijar precios y calcular ventas

La relación que tiene la red nacional es con estación de servicio la cual es que puede haber muchas estaciones de servicio pero solo habrá una red nacional.

**Estación de servicio:**

Respecto a la estación de servicio esta tendrá un nombre, código, gerente y entre los atributos principales pusimos un pair el cual tendrá 2 flotante para la ubicación la cual será por medio de unas coordenadas de GPS, otro tipo mapa el cual tendrá una clave string que será el nombre del tanque y un float que será el valor y llevara la capacidad que tiene el tanque. Por ultimo tendremos una lista con los surtidores.

Bueno en esta clase los métodos serán los siguientes uno para crear, eliminar, activar y desactivar surtidores, También habrá un método para consultar el histórico de las transacciones de cada uno de los surtidores que tendrá dicha estación de servicio; como se nos solicito también tendrá una para simular una venta y por ultimo una para asignarle capacidad a un tanque

En este la relación que tiene la estación de servicio es como la anterior que puede tener solo una red nacional, pero con respecto a la clase transacción que puede ser 0 o muchas transacciones de los diferentes surtidores y surtidores puede tener igualmente 0 o muchos

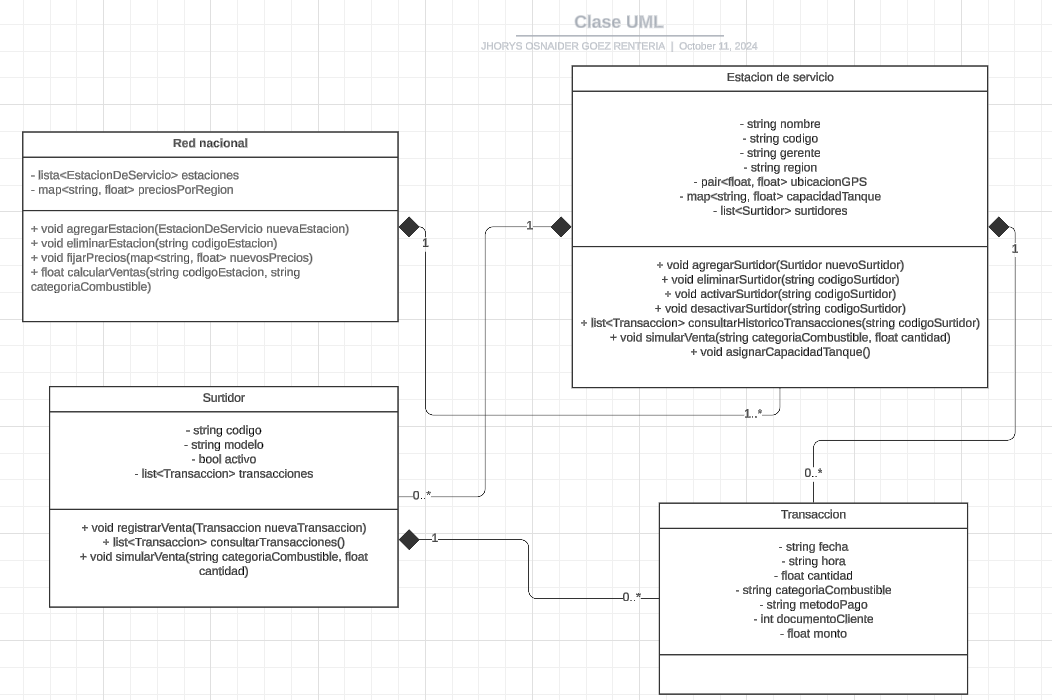
**Surtidor**

Para la clase surtidores tenemos los atributos tipo string que llevara el código de dicho surtido otro para saber su modelo y un boleano que nos indicara si esta activo o inactivo por ultimo una lista que tendrá las transacciones o ventas hechas dicho surtidor.

Respecto a sus métodos tendremos uno de registrar venta el cual será un void que tendrá como parámetros de entrada Transacción y la nueva transacción, Además una lista transacciones en las cuales podemos verificar las transacciones hechas por dicho surtidor por ultimo un método el cual nos permitirá simular la venta y será uno tidpo void que tendrá como parámetros de entrada un string que será la categoría de dicho combustible y un flotante con la cantidad de combustible que va a commprar.

Bueno respecto a como se relaciona esta clase con las demás tenemos pensado que este si no hay estación de combustible pues no habrá surtidores y que cada estación tendrá 0 o muchos surtidos y que cada surtidor tendrá 0 o muchas transacciones.

**Clase Transacción**

En esta clase pensamos que transacción tendría los siguientes atributos tipo string privados (Fecha, Hora) “Nos falta analizar que librería o de que vamos a utilizar para que esto identifique la hora exacta en la que se esta haciendo la transacción”, Además de los dos string anteriores también habrá unos para la categoría del combustible, método de pago, y un int para el documento de identidad de el cliente. Por ultimo dos float uno para la cantidad y el monto de pago.