

*Informe del Desafío II de Informática II* *Fecha: 12/10/24*

**Sistema de comercialización de combustible TerMax**

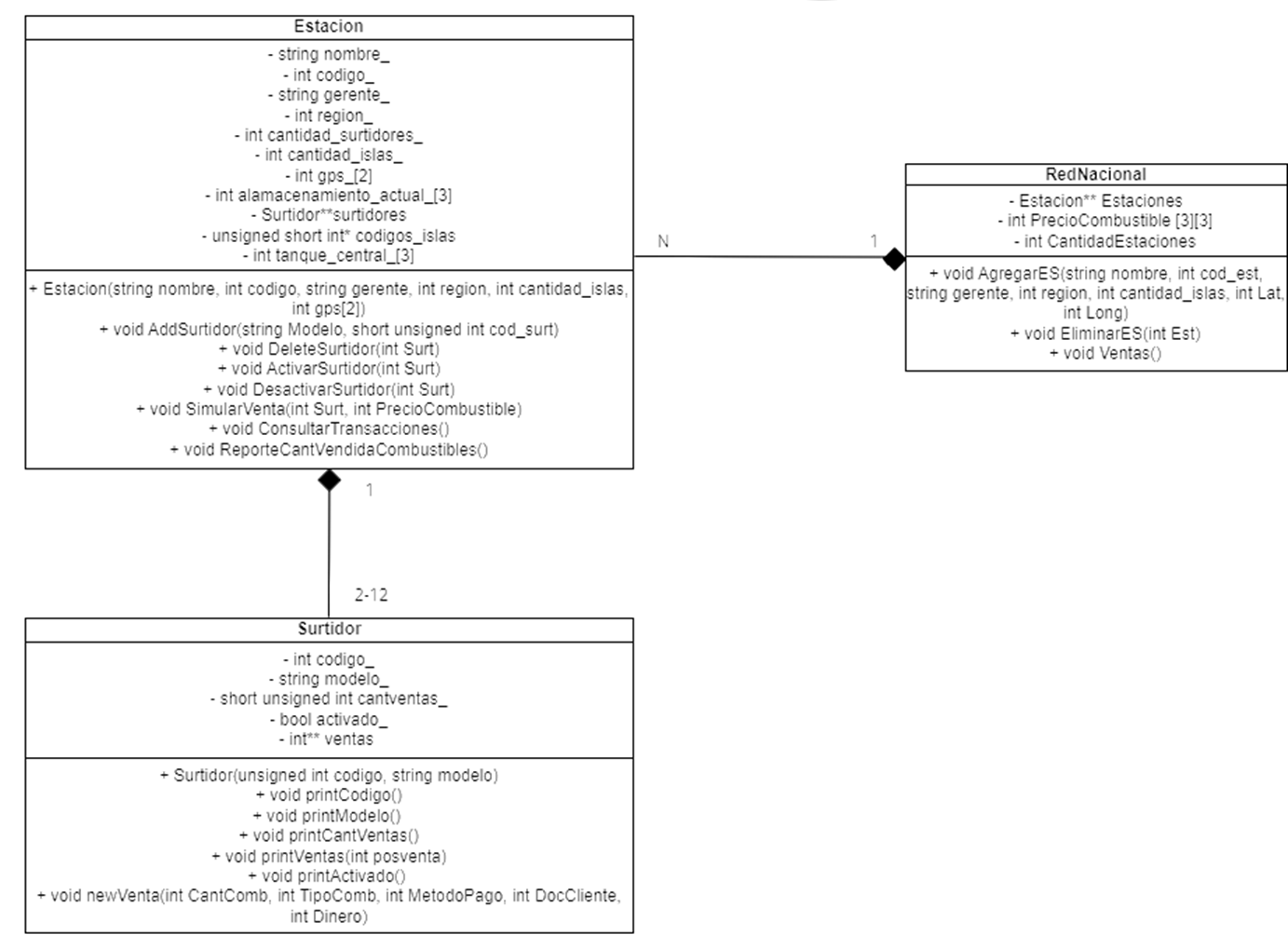
**Santiago Cortés Osorio s.cortes@udea.edu.co Sara María Hincapié Hincapié**  **sara.hincapie1@udea.edu.co**

**Resumen:** Este proyecto se enfoca en la optimización de la gestión de estaciones de servicio de Terpel, con un enfoque particular en la administración de los surtidores de combustible. El sistema diseñado permite una eficiente asignación de recursos, simulación de ventas y control de las transacciones. A través del uso de un enfoque basado en programación orientada a objetos en C++, se busca mejorar la operación y la toma de decisiones dentro de la red nacional de estaciones.

**1) Diseño**

En primer lugar, se abordó el problema por medio de la creación de tres clases principales: RedNacional, Estacion y Surtidor, por medio de estas se da el funcionamiento de las estaciones de servicio, los surtidores y se busca cumplir cada requerimiento del programa.

**1.1 Diagrama de clases:**

****

**2) Análisis del Problema:**

**Clase RedNacional:** Se pensó en ella como la clase encargada de gestionar toda la red de estaciones de servicio en el país, lo que abarca la administración de los surtidores, los precios del combustible y las ventas realizadas en cada estación., de esta clase se resalta el uso de la memoria dinámica en estaciones y precios de combustible.

Se establecen como atributos de la clase un puntero a puntero de Estaciones, un arreglo de 3x3, que busca ajustar los precios del combustible dependiendo la región y un entero que dirá la cantidad de estaciones

Con respecto a los métodos de la clase, dejando de lado los setter y getter incluidos, se busca hacer uso de AgregarES y EliminarES donde se permite la creación y eliminación de estaciones haciendo uso de la memoria dinámica.

Por otro lado, el método Ventas genera un reporte detallado de las ventas que se realizan en cada surtidor y discrimina el tipo de combustible, no obstante, este método interactúa con las clases Estación y Surtidor.

**Clase Estacion:** Se diseña como la clase con la labor de gestionar diversas operaciones en una estación de servicio, por medio de ella se administran los surtidores, el inventario de combustible y registros de las transacciones de venta. Para la creación de surtidores en una estación se busca el uso de memoria dinámica.

En cuanto a los atributos de la clase, se incluyen diversos como los datos sobre la estación de servicios, sin embargo también se incluyen elementos clave, como la creación de un arreglo que representa la capacidad de los tanques dependiendo el tipo de combustible y otro arreglo para gestionar la capacidad en los tanques de almacenamiento, existen otros como un puntero dinámico hacia surtidores que conforman la estación.

Asimismo, en esta clase se permite la gestión dinámica de surtidores mediante los métodos AddSurtidor y DeleteSurtidor, los cuales agregan o eliminan surtidores según las decisiones del usuario.

El método SimularVenta registra ventas simuladas, asignando surtidores activos y actualizando el inventario de combustible tras cada venta. Esto permite mantener un control preciso de los niveles de combustible, además de ajustar la operatividad de los surtidores según sea el caso.

Por otro lado, existen los métodos de activar y desactivar surtidor, este que será relevante para cumplir el requisito de solo poder eliminar una estación en caso de tener todos sus surtidores desactivados

Por último, el método ConsultarTransacciones genera reportes detallados de las ventas realizadas, clasificándolas por surtidor y tipo de combustible.

**Clase Surtidor:** Esta clase es la responsable de gestionar las ventas de cada surtidor dentro de una estación, por consiguiente los únicos atributos que se tienen son los datos del surtidor, un booleano que indica si el surtidor está activado y un puntero a puntero de ventas.

Los métodos incluyen newVenta, que registra una nueva transacción, y printVentas, que permite visualizar los datos de cada venta realizada. Cada transacción almacenada contiene información como la cantidad de combustible vendida, el tipo de combustible, el método de pago y la identificación del cliente. Conforme se incrementa el número de ventas, el sistema adapta el tamaño de un arreglo que almacena dichas transacciones.

Además, la clase Surtidor cuenta con los métodos printCodigo y printModelo para desplegar el código y modelo del surtidor, junto con el método printCantVentas, que muestra la cantidad total de ventas realizadas por ese surtidor. Estos métodos son fundamentales para el control y la supervisión de las actividades de cada surtidor dentro de la estación.