Informe técnico sobre el proyecto

“Sistema de gestión de suministro de combustible”

Presentado por:

Sofia Casas Centeno

Asignatura: Informática II

Profesor: Augusto Salazar

Universidad de Antioquia

**Introducción**

Hoy en día, es crucial una administración eficaz de las estaciones de gestión de combustible para asegurar un abastecimiento apropiado y sin contratiempos para los vehículos de motor. Una estación de servicio, también llamada bomba de combustible es un lugar de venta donde se comercializan combustibles derivados del petróleo, como gasolina, diésel, entre otros, junto con otros recursos para vehículos. En este escenario, la creación de un sistema de administración basado en Programación Orientada a Objetos (POO) facilita la organización y el control de los distintos componentes que participan en el funcionamiento de una red de estaciones de servicio. El sistema debe tener la capacidad de administrar eficazmente las estaciones, sus surtidores y las operaciones de venta de combustible, proporcionando una solución completa y automatizada para mejorar las actividades cotidianas. El presente proyecto ofrecerá un análisis detallado del problema planteado, describiendo las tareas definidas para el desarrollo e implementación de los algoritmos propuestos. Se abordarán los desafíos enfrentados durante la creación del código fuente y se presentará la solución final implementada, destacando su eficiencia y validación a través de pruebas exhaustivas.

**Desarrollo técnico**

La idea central o el objetivo es desarrollar un sistema de gestión para la empresa TerMax, líder en el suministro de combustibles en Colombia, que permita administrar de manera eficiente las estaciones de servicio. Cada estación tiene un tanque central con tres tipos de combustibles (Regular, Premium y EcoExtra) y varias máquinas surtidoras que distribuyen el combustible a los clientes. El sistema debe controlar aspectos como la gestión del inventario de combustible, el registro de ventas diarias con detalles de transacciones, y la actualización del inventario del tanque central en función de las ventas realizadas, garantizando que las ventas solo cobren lo que efectivamente se suministre.

El sistema por desarrollar para la red de estaciones de servicio TerMax debe permitir gestionar la red nacional de estaciones y sus operaciones. Las funcionalidades clave incluyen:

Gestión de la red:

- Agregar/eliminar estaciones de servicio (solo si no tienen surtidores activos).

- Calcular ventas por categoría de combustible.

- Fijar los precios del combustible para toda la red.

Gestión de estaciones:

- Agregar/eliminar surtidores.

- Activar/desactivar surtidores.

- Consultar el histórico de transacciones de surtidores.

- Reportar ventas por categoría de combustible.

- Simular ventas de combustible.

- Asignar capacidades de tanque (aleatoriamente entre 100 y 200 litros por categoría).

Sistema de verificación de fugas:

- Detectar fugas de combustible verificando que lo vendido más lo almacenado sea más del 95% de la capacidad original del tanque.

Simulación de ventas:

- Simular ventas en surtidores activos de una estación, seleccionando un surtidor y cantidad de combustible aleatoria.

- Mostrar los detalles de la transacción, con precios dependientes de la región (Norte, Centro, Sur).

El sistema debe gestionar de forma automática los códigos de estaciones, surtidores y transacciones, además de supervisar posibles fugas de combustible en las estaciones.

A continuación, se presentará la correspondiente solución al problema planteado teniendo en cuenta que se trabajará bajo una perspectiva superficial por lo que todo lo propuesto es con base a lo definido en el problema:

Se presenta un sistema de gestión de una red de estaciones de servicio que permite simular y registrar la venta de combustibles, administrar surtidores, verificar posibles fugas y manejar diversas estaciones a nivel nacional. A continuación, se desglosan los componentes principales del código:

Clases Principales:

**RegistroVenta**: Esta clase almacena la información de cada venta realizada, como la fecha, hora, cantidad de litros vendidos, tipo de combustible, método de pago, documento del cliente, y el monto total.

**Maquina**: Es una clase base que contiene el código y modelo de los equipos.

**Surtidor**: Gestiona los surtidores en las estaciones. Cada surtidor puede registrar ventas, calcular el dinero total a pagar, y registrar dicha información en un historial. Además, cada surtidor puede activarse o desactivarse.

**TanqueCentral**: Clase que gestiona el almacenamiento de los combustibles y permite verificar si hay fugas al comparar la capacidad del tanque con las ventas realizadas.

**EstacionServicio**: Esta clase representa una estación de servicio, que contiene un conjunto de surtidores y permite realizar operaciones como agregar surtidores, fijar precios, verificar fugas, y mostrar el historial de ventas.

**RedNacional**: Es el administrador de varias estaciones de servicio en diferentes regiones. Permite agregar, eliminar y gestionar estaciones de servicio a nivel nacional.

Funciones Principales:

**realizarVenta**: Registra una venta de combustible, sumando los litros vendidos de cada tipo (Regular, Premium, EcoExtra), calcula el total a pagar en función de los precios definidos y guarda el registro en un historial.

**activar**/**desactivar**: Permiten cambiar el estado de un surtidor entre activo e inactivo.

**verificarFuga**: Revisa si hay inconsistencias entre las ventas y la capacidad del tanque, lo que podría indicar una fuga de combustible.

**mostrarHistorialVentas**: Muestra un resumen de todas las ventas realizadas por un surtidor.

Estructura General: El programa cuenta con un menú principal que permite gestionar una red de estaciones de servicio distribuidas en diferentes regiones. El usuario puede agregar nuevas estaciones, administrar los surtidores dentro de cada estación, simular ventas, fijar precios y verificar el estado del tanque.

**Problemas afrontados a partir del desarrollo de los algoritmos:**

Durante la elaboración de la solución se tuvieron presentes algunas dificultades sobre el contexto y la manera de interpretar lo pedido por el cliente, puesto que al no tener conocimiento suficiente del tema se tuvo se prestó para ambigüedades y errores durante el proceso de codeo. No obstante, a medida de la elaboración se dio paso a una idea de una estructura más sólida.

**Conclusión**

A partir de los conocimientos adquiridos en clase y del esfuerzo realizado para la entrega de este proyecto se puede llegar a la conclusión de que la programación orientada a objetos es una manera eficaz y a la vez un poco compleja a la hora de crear una estructura para crear programas dinámicos y que se puede llevar a niveles bastantes altos y formar grandes proyectos.