

Informe de progreso desafío N.º 2

Juan David Castaño Hernández

Diego Andrés Pedrozo Figueroa

Facultad de Ingeniería, Universidad de Antioquia

Informática II

Aníbal José Guerra Soler

12 de octubre de 2024

INTRODUCCIÓN

El siguiente informe tiene como objetivo documentar los avances iniciales del desafío II. Hasta el momento, hemos enfocado nuestros esfuerzos en el análisis preliminar del problema y los primeros pasos del desarrollo, por lo que dedicamos buen tiempo al análisis del problema para realizar una codificación exitosa y no divagar en la solución. Si bien aún estamos en las fases tempranas del proyecto, hemos comenzado a definir la estructura básica del programa y hemos realizado avances significativos en la comprensión de los requisitos y propuestas para desarrollar la solución. Presentamos los análisis y las propuestas de solución desarrolladas hasta el momento.

ANÁLISIS DEL PROBLEMA

Se pide simular y modelar el funcionamiento y gestión de una red nacional de estaciones de servicio, que son centros de suministro de combustible. Es necesario hacer uso del paradigma de programación orientada a objetos, pues resulta ser la opción más eficiente para dar solución a este problema. Al emplear este paradigma debemos esforzarnos lo máximo posible para que el software desarrollado sea fiel a lo que sucede en la realidad, es por esto que las funcionalidades del software deben ser similares a las funcionalidades que tiene una estación de servicio de la vida real. Identificamos las siguientes funcionalidades:

1. Creación / eliminación de estaciones de servicio: el programa debe ser capaz de crear y de eliminar estaciones de servicio. Para esto es necesario hacer uso de la ubicación en coordenadas GPS para no permitir la creación de una estación de servicio sobre otra existente, además de verificar que no se vaya a eliminar una estación donde no está ubicada realmente.
2. Asignación de precios de combustible según la ubicación y la categoría de combustible: el problema describe la ubicación de las estaciones de servicio y las divide en tres regiones: norte, sur y centro, donde cada estación debe tener tres categorías de combustible: regular, premium y EcoExtra. El programa de software debe permitir ajustar el precio del combustible según la región y la categoría de combustible.
3. Simulación de ventas: se debe simular la venta de combustible a un cliente, por lo que se le debe pedir la categoría y cantidad de combustible que desea, el método de pago y la identificación.
4. Actualización de la cantidad de combustible: como estamos teniendo en cuenta que el software es una representación del mundo real, entonces al vender una cantidad de combustible se reduce la cantidad del mismo en el tanque central de la estación de servicio.
5. Base de datos: teniendo en cuenta que al finalizar la ejecución del programa se pierde la información de las ventas esto no sería para nada bueno en una empresa real, por lo que es necesario hacer uso de archivos .txt para almacenar este tipo de información.

6. Sistema de verificación de fugas: el combustible al ser un fluido contenido en un recipiente, en este caso un tanque central de almacenamiento en una estación de servicio, entonces puede presentar fugas, por lo tanto es necesario establecer un algoritmo que permita identificar fugas en una estación de servicio.

Sabemos también que se dispondrá de un menú para acceder a las diferentes opciones que pueda tener el usuario del programa, por lo que debemos establecer procedimientos que aseguren que un cliente no pueda crear o eliminar una estación de servicio, o desactivar un surtidor por ejemplo, por lo que es necesario incorporar una clave de gerente para realizar este tipo de acciones.

DISEÑO DE LA SOLUCIÓN

En la arquitectura del sistema se implementarán las siguientes clases:

1. Clase Estaciones: Esta clase tiene como atributos la capacidad del tanque principal de combustible que en este caso serán tres uno para cada tipo de combustible y estos tendrán una capacidad entre 100 y 200 litros.
2. Clase Surtidores: Esta clase tiene como atributos y esta misma tiene relación con la clase Estaciones ya que utiliza el atributo de la clase estaciones para así al momento de hacer una venta tener en cuenta la capacidad del tanque principal de la estación para si no poder vender más de lo que tiene disponible.

Las funciones planteadas hasta el momento son:

1. **Función buscarFuga:** verifica la capacidad del tanque y la cantidad de combustible vendido y que este valor no sea inferior al 95% de la cantidad inicial del tanque.
2. **Función mostrarHistorial:** Imprime el historial de ventas de cada surtidor cuando el usuario lo pide.

CONCLUSIONES

Las clases definidas para este problema están relacionadas de la misma forma como sucede en la realidad: los surtidores son parte de las estaciones de servicio, y a su vez las estaciones de servicio son parte de una red nacional. Mirar el comportamiento real de una red de estaciones de servicio nos da una guía a la hora de establecer los métodos de una clase y cuál es el comportamiento de la misma.

Según las funcionalidades especificadas en el planteamiento del problema hemos definido los métodos de las clases, además de otros métodos propios para el correcto funcionamiento del programa y el cumplimiento satisfactorio de los requisitos establecidos. Continuamos con el proceso de relacionamiento entre clases a través de los métodos de cada clase, para así aproximarnos más al funcionamiento real de una red de distribución de combustible.