

**INFORME DESAFÍO II INFORMATICA II**

Jaider Bedoya Carmona

Wilbergt Alexander Osorio Trespalacios

Informe sobre proceso de desarrollo del desafío dos

PROFESORES

Augusto Salazar

 Aníbal Guerra

Universidad de Antioquia

Facultad de Ingeniería

Ingeniería Electrónica

Medellín

2024

**Análisis del problema**

La red Nacional se divide en estaciones y estas mismas en surtidores.

La propuesta a desarrollar, es implementar 3 clases que tienen una relación estrecha en sus características, las cuales son:

* Clase Red Nacional 🡪 Busca expresar simplemente una red de varias estaciones(por el momento solo en Colombia).

* Clase Estación 🡪 Estaciones que están repartidas a nivel nacional y cada una cuenta con atributos de diferenciación e implícitamente contiene un numero de surtidores.

* Clase Surtidor 🡪 Permite administrar las características y tareas hechas por un surtidor, puesto que el surtidor es el contacto directo entre el cliente y la empresa.

Para este desarrollo estamos planteando usar archivos como una “base de datos” para guardar la información perteneciente a cada una de las entidades implicadas en el sistema y así al momento de usar las funciones de despliegue sea más fácil acceder a la información.

**Diagrama de clases**

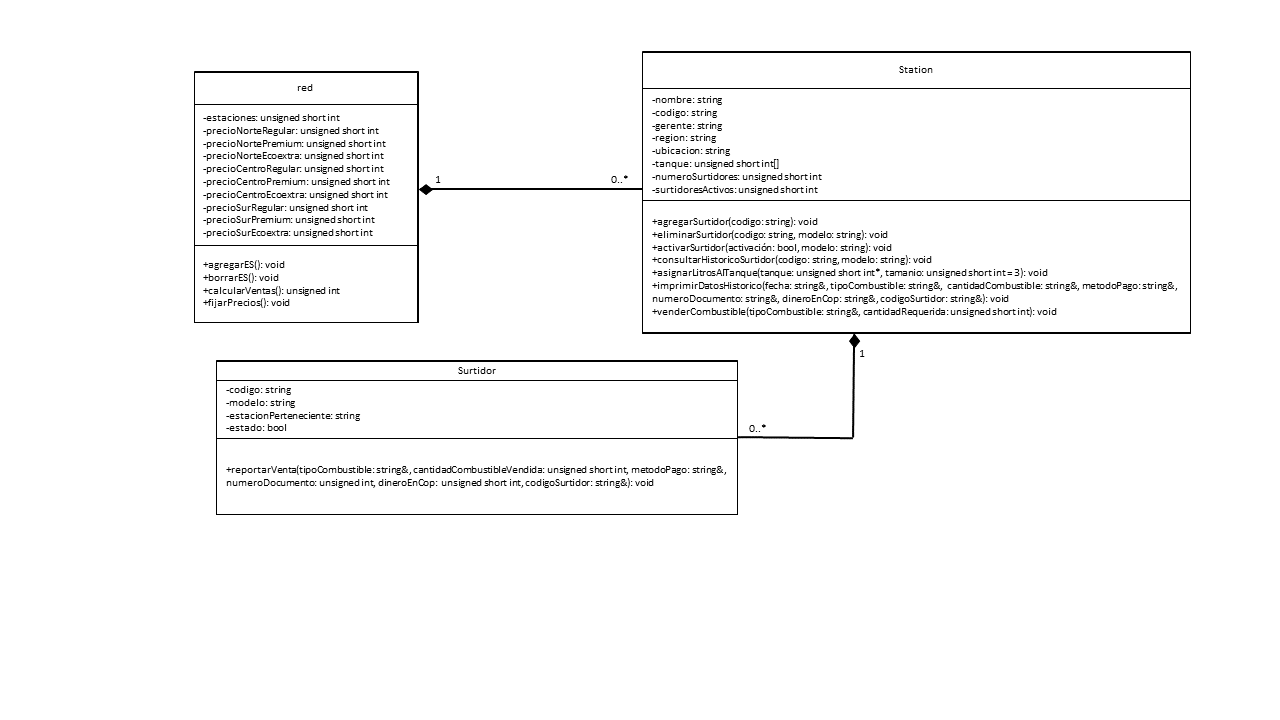


Ilustración 1 Diagrama de clases

**Algoritmos implementados**

Algunos algoritmos implementados fuera de cada clase que son importantes en el programa son las siguientes funciones:

**Station leerDatosEstacion(const string datosEstacion){}**

Devuelve una instancia de la clase Station, recibiendo como parámetro un string que contiene toda la información esencial de la estación, para separar y asignar cada atributo a la instancia.

**void llenarArregloEstaciones(Station\* arregloEstaciones, const string& nomArchivo){}**

Llena un arreglo que contiene todas las instancias de las estaciones existentes en el archivo de texto que usamos como “base de datos”, esto con el fin de que los tratamientos de las estaciones sean más rápido y eficiente.

**void Menu(){}**

Tiene como objetivo crear una interfaz semigrafica por consola, en la cual el usuario accede a las opciones que están establecidas en los requisitos del desafío.

**Otras funciones auxiliares:**

**unsigned short int contarLineasEnArchivo(const string& nombreArchivo){}**

**unsigned short int contarCaracteresEnString(const string cadenaCaracteres, const char caracterParaContar){}**

Estas funciones nos ayudan en la implementación ya que hacen tareas muy sencillas, las cuales deben ser realizadas en numerosas ocasiones.

**Problemas afrontados**

El primer problema en el desarrollo fue la idea de correlación entre las clases y pensar en la definición de cada atributo y método, puesto que también hay que tener en cuenta la abstracción para no sobre objetivizar las entidades.

El tener criterio para decidir cómo distribuir los reportes de venta, información de estaciones y surtidores, y demás cosas que se necesitan guardar, resulta un problema que le dimos solución con criterios de eficiencia, facilidad y organización, siempre con argumentos para cada decisión.

Al usar metodos y funciones que dependian de otros, ambas ramas de desarrollo tenian partes que dependian de otras, sobre todo el menu principal que es el que interactua con el usuario y despliega sus respectivas funcionalidades, por lo que en algunos casos no coincidian ambas partes y se debian hacer ajustes que en algunos casos fueron cambios grandes en alguna de las partes.

**Evolución de la solución**

En la primera sesión simplemente nos limitamos a analizar muy bien el problema y pensar en la solución, teniendo en cuenta todos los requerimientos para poder ir pensando en la implementación que llevaremos luego.

Empezando con la implementación del programa, se dividió equitativamente el desarrollo del mismo, cada uno tomando 2 tareas grandes, ya sea una o dos clases, o funciones anexas.

Para el desarrollo declaramos 3 clases como lo planeamos, divididos respectivamente en archivos .h y .cpp, para declaración e implementación de cada componente, usando la programación modular y buenas prácticas para el orden del proyecto.

Se llevó a cabo el desarrollo satisfactoriamente de la clase surtidor, estación y red, además de funciones anexas que nos ayudan al programa, así como el menú principal, hubo bastante trabajo colaborativo usando la herramienta de GitHub como principal fuente de transferencia y compartición de los archivos desarrollados.