

Desafío II: Sistema de comercialización de combustible TerMax

Uno de los objetivos de programar consiste en modelar una situación del mundo real, delimitada por un contexto. Al representar en el programa las entidades y acciones de interés, se pueden estudiar o automatizar las dinámicas propias del contexto. Si el programa es un modelo apropiado de la situación real, entonces los resultados de su ejecución reflejan las interacciones reales; con cierta confianza.

Objetivos

- Desarrollar la capacidad de análisis y solución de problemas en los estudiantes, enfrentándolos a problemáticas de la vida cotidiana.
- Verificar si el estudiante adquirió las destrezas y conocimientos fundamentales de la programación orientada a objetos en C++: abstracción, encapsulación, relaciones, diseño de diagrama de clases, funciones amigas, sobrecarga y uso de plantillas.

Si usted ha llevado un proceso disciplinado de aprendizaje a lo largo del semestre, esta es una oportunidad de demostrarlo y podrá plantear una solución satisfactoria. En caso contrario, podrá identificar sus debilidades y tomar medidas a fin de poder abordar situaciones similares a futuro.

Trate de valorar la verdadera complejidad del problema planteado, no se rinda antes de intentarlo o de plantear los posibles escenarios de solución. Se dará cuenta que si bien, al principio le puede parecer difícil; ya ha tenido la oportunidad de enfrentarse a problemas similares. Si se toma el tiempo adecuado para analizar, el proceso de codificación será eficaz y no le tomará mucho tiempo.

Esperamos que disfrute del desafío propuesto. Lea primero todo el documento antes de comenzar y asegúrese de entender muy bien las instrucciones antes de desarrollar esta actividad evaluativa.

Fue revisado por los profesores Aníbal Guerra y Augusto Salazar.

Introducción

Una estación de servicio (E/S) o bomba gasolinera, es un punto de venta de combustible y otros suministros para vehículos de motor; como gasolina y derivados petrolíferos. El objetivo de este desafío es desarrollar un sistema de gestión para una red de estaciones de servicio de combustible utilizando POO. El sistema debe

permitir la gestión eficiente de las estaciones de servicio, los surtidores y las transacciones de venta de combustible.



Figura 1. Estación de servicio (Tomado de : <https://globalestacionesdeservicio.com/l/>)

La empresa TerMax es una compañía líder del suministro de combustibles en el territorio colombiano. Dicho liderazgo se ha consolidado a través de una amplia cadena de estaciones de servicio repartidas a nivel nacional.

Cada estación de servicio posee un nombre, un código identificador, un gerente, una región y una ubicación geográfica (expresada en coordenadas GPS). Además, cada una tiene un tanque central donde se almacenan separadamente las 3 categorías de combustible disponible para vender (Regular, Premium y EcoExtra). La capacidad de almacenamiento de cada categoría no necesariamente es uniforme en el tanque y puede variar entre los tanques de diferentes estaciones de servicio. A este tanque central se conectan entre 2 a 12 máquinas surtidoras desde las cuales se venderá directamente a los vehículos de los clientes. La estación de servicio puede estar físicamente subdividida en varias naves o islas, en las cuales se agrupan físicamente varios surtidores de forma contigua.

Cada surtidor posee un código identificador y un modelo de dicha máquina, y están habilitados para distribuir las tres categorías de combustible. Cada uno registra individualmente todas las ventas realizadas en el día, registrando: la fecha, hora, cantidad y categoría de combustible, método de pago (Efectivo, TDebito, TCrédito), número de documento del cliente y la cantidad de dinero respectiva. Adicionalmente, cada vez que una surtidora vende combustible, se debe actualizar la disponibilidad del mismo en el tanque de la estación. En caso de que una venta

requiera más combustible del disponible, se procede con la venta pero sólo debe cobrarse la cantidad suministrada.

Funcionalidades esenciales

Sólo se construirá una instancia de red nacional de estaciones de servicio. Adicional a los constructores, destructor, getters, setters y operaciones de despliegue necesarias; **incluya un menú para acceder a funcionalidades que permitan:**

I - Gestión de la red

Red principal Funcionalidades

- a.- Agregar estaciones de servicio.
- b.- Eliminar una E/S de la red nacional (sólo si no posee surtidores activos).
- c.- Calcular el monto total de las ventas en cada E/S del país, **discriminado por categoría de combustible.** (Regular, Premium y EcoExtra)
- d.- Fijar los precios del combustible para toda la red.

Se debe analizar si, si los precios para toda la red serán iguales o dependiendo del lugar o las zonas (NORTE, CENTRO Y SUR)

II - Gestión de estaciones de servicio

- a.- Agregar/eliminar un surtidor a una E/S.
- b.- Activar/desactivar un surtidor de una E/S.
- c.- Consultar el histórico de transacciones de cada surtidor de la E/S.
- d.- Reportar la cantidad de litros vendida según cada categoría de combustible (Regular, Premium y EcoExtra)
- e.- Simular una venta de combustible.
- f.- Asignar la capacidad del tanque de suministro, con un valor aleatorio entre 100 y 200 litros para cada una de las categorías

Por último, considere subprogramas para: la gestión automática de todos los códigos en el sistema, la verificación de fugas de combustible en cualquiera de las estaciones de la red y para la simulación de una venta.

III - Sistema nacional de verificación de fugas

El programa debe permitir detectar la existencia de fugas de combustible en cualquiera de las estaciones del país.

Para ello, según cada categoría de combustible, debe verificarse que lo vendido más lo almacenado en el tanque de la E/S corresponda a más del 95% de la capacidad original del tanque. Esta verificación opera sobre una estación de servicio específica que sea seleccionada entre todas las del país.

IV - Simulación de ventas

Dada una E/S, asigne de forma aleatoria uno de los surtidores activos para que gestione la transacción. Suponga que en una venta se solicita en forma aleatoria entre 3 y 20 litros de gasolina. Una vez que se ha bombeado la gasolina deberán desplegarse los datos de la transacción.

El precio por litro de cada categoría de combustible depende de la región a la que pertenece la estación. Sólo se consideran tres regiones: Norte, Centro y Sur. Dicho valor es temporal, pudiendo variar entre días.

Requisitos del desarrollo

El cliente requiere un programa para la gestión de la red nacional de suministro de combustible. Este debe permitir representar la estructura de una red nacional de suministro cualquiera y ejecutar las operaciones especificadas en este enunciado.

Se recomienda delimitar el desarrollo a lo aquí descrito, ya que la dimensión real del problema puede tener una complejidad superior al tiempo estipulado para la entrega. En caso de duda sobre los requerimientos, consulte con el cliente.

De acuerdo con lo anterior, entre otras cosas, usted deberá:

1. **[10%]** Contextualice el problema, analícelo y diseñe el diagrama de clases correspondiente a su solución. Refleje adecuadamente las relaciones implícitas en la problemática. Recuerde utilizar la notación UML simplificada impartida en las clases teóricas. A pesar de tener una baja ponderación, la entrega de este diagrama es obligatoria y sin ella no se dará lugar a la sustentación.
2. **[10%]** Seleccione debidamente los tipos y estructuras de datos que le permitirán implementar las clases planteadas en su diagrama.
3. **[10%]** Previo a la implementación, verifique el cumplimiento del requisito de eficiencia especificado en la sección 3 del apartado “Requisitos de la entrega” de este documento.
4. **[15%]** Implemente las funcionalidades: “Verificación de fugas de combustible” y “simulación de ventas”.
5. **[55%]** Presente la implementación de su programa, cuya interacción se centra en un menú que permita acceder de manera independiente a las funcionalidades correspondientes a cada clase. Considere las restricciones implícitas en la lógica del problema. Por ejemplo, una máquina surtidora no se puede crear en el aire, sólo puede crearse si está adscrita a una estación de servicio en la que el código de máquina no exista ya.

Formalidades de la entrega

A continuación, se describen los requisitos que se deben cumplir. El incumplimiento de cualquiera de ellos implicaría alguna penalidad o que su nota sea cero.

1. Genere un informe en donde se detalle el desarrollo del proyecto, explique entre otras cosas:
 - a. Análisis del problema y consideraciones para la alternativa de solución propuesta.

- b. Diagrama de clases de la solución planteada. Adicionalmente, describa en alto nivel la lógica de las tareas que usted definió para aquellos subprogramas cuya solución no sea trivial.
 - c. Algoritmos implementados debidamente intra-documentados.
 - d. Problemas de desarrollo que afrontó.
 - e. Evolución de la solución y consideraciones para tener en cuenta en la implementación.
- 2. La solución debe ser implementada en lenguaje C++ y debe basarse en el paradigma de POO.
- 3. La implementación debe considerar el criterio de eficiencia y el buen diseño tanto de las estructuras de datos como de los módulos implementados. Nota: considere además en este apartado la utilización de referencias vs la realización de copias innecesarias sobre datos estructurados.
- 4. La implementación no debe considerar el uso de la STL. Las estructuras de datos usadas deben ser creación propia y deben considerar el uso de memoria dinámica.
- 5. Se debe crear un repositorio público para cargar todos los archivos relacionados a la solución planteada (informe, código fuente y otros anexos).
- 6. Una vez cumplida la fecha de entrega no se debe hacer modificación alguna al repositorio.
- 7. Se deben hacer *commits* de forma regular (al menos uno al día) de tal forma que se evidencie la evolución de la propuesta de solución y su implementación.
- 8. Se debe adjuntar un enlace de *youtube* a un video que debe incluir lo siguiente:
 - a. Presentación de la solución planteada. Análisis realizado y explicación de la arquitectura del sistema (3 minutos máximo).
 - b. Demostración de funcionamiento del sistema. Explicar cómo funciona: ejemplos demostrativos (3 minutos máximo).
 - c. Explicación del código fuente. Justifique la elección de las variables y estructuras de control usados, y destaque que ventaja ofrecen estos en comparación con otras opciones (5 minutos máximo).
 - d. La duración total del video no debe exceder 11 minutos ni ser inferior a 5 minutos.
 - e. Asegúrese que el video tenga buen sonido y que se puedan visualizar bien los componentes presentados.
- 9. El plazo de entrega se divide en dos momentos:
 - a. El día 12 de Octubre para adjuntar la evidencia del proceso de análisis y diseño de la solución.
 - b. El día 15 de Octubre para adjuntar la evidencia del proceso de implementación.
- 10. Se deben adjuntar **dos enlaces**: uno al repositorio y otro al video, nada más.

11. Para la evaluación del desafío se realizará una sustentación oral en un horario concertado con el profesor. La asistencia a la sustentación es obligatoria para optar a calificación.