2do Programa Evaluado

Se necesita desarrollar una aplicación para la supervisión y control de los parámetros Operativos de un tanque de almacenamiento de petróleo en una planta industrial. La Aplicación debe ser capaz de:

* monitorear constantemente las condiciones del tanque.
* Detectar desviaciones críticas en los parámetros y generar alertas.

Para el desarrollo de la aplicación deberá crear un programa modular (proyecto en devC++), utilizando programación orientada a objetos para simular el proceso de producción de dispositivos electrónicos.

**Requisitos del programa:**

**Clases principales:**

* **TanquePetroleo**: Representa un tanque de almacenamiento.
* **Sensor**: Representa un sensor que mide un parámetro específico.
* **Alerta**: Representa una alerta generada en caso de que un parámetro Supere los límites establecidos.
* **SistemaSupervisorio**: que represente el sistema de supervisión de los Tanques
* **Supervisor**: Representa al encargado de monitorear los tanques.

Crear un menú interactivo para que el usuario pueda realizar las Siguientes operaciones:

1. Registre y guarde en archivos de datos la información de tanques de Almacenamiento, sensores, alertas, y supervisores.
2. Genere un informe del:

* Estado actual de todos los tanques monitoreados, incluyendo alertas Generadas.
* Listado de supervisores indicando número de tanque supervisado y fecha de supervisión.

1. Asigne y cambie supervisores en el sistema.
2. Agregue tanques al sistema de supervisión.
3. Simule la actualización de los parámetros de los tanques.
4. Detecte condiciones críticas, como nivel bajo o excesivo

**SOLUCION:**

Para este código primeramente se mostrara un menú con todas las opciones pedidas por el cliente.

Para la parte a) se guardara únicamente la información actual del código en su ejecución dentro de un archivo de datos.

Para la parte b) se necesitara primero haber iniciado los tanques y pasados por el sistema supervisorio, esta opción debe indicar un error en caso de no haber iniciado los tanques, o ingresado los supervisores correspondientes a los tanques, junto con su fecha de supervisión.

Para la parte c) se adaptara el sistema a los cambios ingresados por el usuario con respecto a los supervisores.

Para la parte d) se adaptara el sistema para que el usuario pueda añadir tanques al sistema de supervisión. Dichos tanques tienen que ser supervisados por los supervisores ya ingresados al sistema.

Para la parte e) se mostrara en pantalla el estado actual de los parámetros de cada tanque.

Para la parte f) se hará una serie de límites a los parámetros de los tanques, de tal forma que se pueda dividir en tres rangos:

Rango bajo; Rango normal; Rango excesivo.

Para las clases requeridas por el código se tuvieron en cuenta las siguientes consideraciones:

Para la clase tanque: se investigo acerca de los parámetros a usar para la representación de los mismos; esto dio como resultado la siguiente estructura de la clase:

**Clase:** TanquePetroleo (Representa un tanque de almacenamiento)

Variables:

* CodigoTanque
* Nivel de líquido
* Temperatura
* Presión
* Densidad
* Nivel de agua

Método:

* Generar tanque (acá se iniciaran los valores del tanque de forma aleatoria para generar el efecto de simulación).

Para la clase sensor: esta se encargara de detectar y analizar los valores dentro de los tanques, para ello los atributos del objeto creado en la clase tanquepetroleo se pueden componer con la clase sensor. Además esta clase contara con los límites físicos que determinen si los parámetros de los tanques se encuentran en:

* Nivel bajo
* Nivel normal
* Nivel excesivo

De esta forma la estructura de la clase sensor es:

**Clase:** sensor (Composición con TanquePetroleo)

**Métodos:**

* MedirNivelLiquido
* MedirNivelAgua
* MedirTemperatura
* MedirPresión
* MedirDensidad

Para la clase alerta: esta se encargara de mostrar una alerta en caso de que los parámetros detectados en los sensores estén fuera de lo normal, por lo que dentro de un método de la clase alerta se usaran todos los métodos de la clase sensor para determinar el estado activo/inactivo de la alerta en cada parámetro; esto nos deja con la siguiente estructura:

**Clase**: Alerta (heredado de sensor)

**Métodos:**

* DetectarAlerta
* ImprimirAlerta

Para la clase supervisor: esta clase se encargara de representar a los supervisores dentro del sistema, dichos supervisores tienen la opción de generar un reporte

**Clase:** Supervisor

**Parámetros:**

* Nombresupervisor
* Fecha

**Métodos:**

* GenerarReporte

Para la clase sistema supervisorio:

**Clase:** SistemaSupervisorio

**Métodos:**

* LeerParametros