

Trabajo Práctico Final Integrador DIED 2019

Publicación	Entrega
21/06/2018	02/08/2019

Objetivo

Este trabajo práctico tiene por objetivo aplicar los conceptos estudiados dentro del área de estructuras de datos, modelado UML y programación en Java. También deberán aplicarse conocimientos en el uso del API de estructuras de datos.

Consignas

Se desea **implementar una aplicación de escritorio escrita en Java** que permita administrar un **Sistema de Gestión Logística** para una industria que tiene plantas de producción distribuidas a lo largo del país.

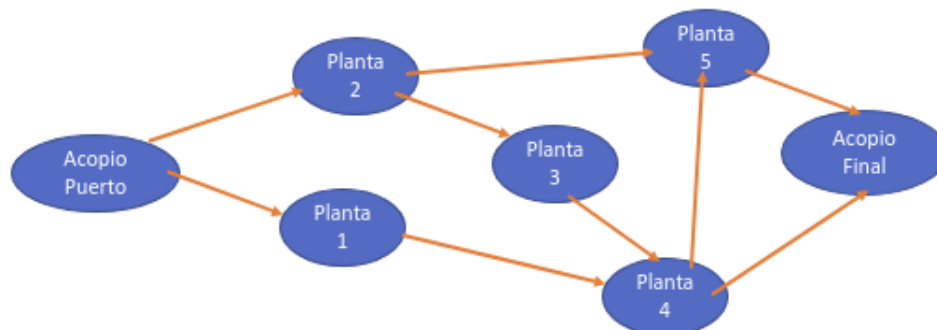
Esta industria posee una planta de acopio de materiales que son insumos para su producción ubicado en la ciudad donde se encuentra el puerto en el que recibe todos los materiales.

Luego posee numerosas plantas de producción que se encuentran distribuidas a lo largo de todo el país y los materiales una vez que son recibidos en la planta de acopio, se distribuyen a estas plantas de producción según lo vayan solicitando para poder realizar sus tareas.

Finalmente, todos los materiales que no se entregan se dejan en una planta acopio final, que está ubicado en el centro del país.

El siguiente es un ejemplo posible de las rutas entre las plantas. Cada ruta posee los siguientes datos:

- Distancia (en kilómetros)
- Duración del viaje (en minutos)
- Peso máximo aceptado por el camino (en toneladas)



Cada planta guarda información de su nombre y un id. Además, posee información del stock de cada producto.

Cuando se registra el stock de cada producto en cada planta, se almacena el ID del registro, el insumo que se representa, la cantidad del insumo que hay en stock según las unidades determinadas, y cuál es el punto de reposición o punto de pedido (es decir un valor que determina que si la cantidad de unidades es menor al punto de reposición esta planta necesita reponer el producto, antes de que su stock llegue a cero)

De los insumos que recibe esta industria se guardan los siguientes datos:

- id;
- descripción;
- unidadDeMedida (es la unidad en que se calcula el costo y se solicita el insumo, puede ser Kilo, Pieza, Gramo, Metro, Litro, M3, M2, etc)
- costo: es el costo del insumo
- stock: cantidad de unidades que hay en el centro de acopio
- peso: peso en KG del insumo
- esRefrigerado: si su transporte debe realizarse en un medio refrigerado para mantener sus propiedades.

Hay un subconjunto de insumos, que son líquidos, por lo que su pedido se realiza siempre por litros, y además de todos los datos que se guardan para los insumos en general, se guarda también la densidad de dicho producto, de forma tal de calcular el peso.

De esta manera el peso de un producto se calcula como:

$$P \text{ [kg]} = \text{Volumen [m}^3\text{]} * \text{Densidad [Kg / M}^3\text{]}$$

Entonces, por ejemplo, si una planta solicita 1000 litros de Oxígeno licuado, siendo la densidad del oxígeno de 1141 kg/m³ el peso de esos 1000 litros será $1141 * 1\text{m}^3 = 1141 \text{ KG}$.

El sistema también debe permitir registrar los camiones que posee disponible esta industria para asignar al transporte de un producto. Los camiones guardan información del ID del registro, la marca y modelo, el dominio, el año, el costo por kilómetro transportado y si es apto para transportar líquidos, y cuál es su capacidad.

Requerimientos

Parte 01 – Definir el modelo del dominio del problema

Defina un diagrama de clases UML que muestre las clases principales del dominio del problema y la relación entre ellas.

Parte 02 – Gestión de entidades

El sistema debe permitir crear, buscar, editar y borrar los registros creados á través de una interface gráfica de usuario SWING.

Debe poder crear

- Insumos
- Plantas
 - Cuando se muestren las plantas en una tabla, se debe mostrar además por cada una de ellas la información solicitada en el taller 04.
- Agregar stock de insumos a cada planta
- Caminos entre plantas

- Camiones.
- Cualquier otra información que considere necesaria para registrar en el sistema.

(Opcional) Los datos podrá guardarlos en una base de datos relacional

Parte 03 – Gestión de insumos

El sistema debe permitir buscar insumos, por nombre, por costo mínimo y máximo o por stock mínimo o máximo mostrar la información del insumo y ordenarlos por de manejar ascendente o descendente según elija el usuario por:

- Stock total en el sistema (es decir la suma de stock que hay de un producto en todas las plantas y los centros de acopio)
- Costo
- Nombre

Para implementar esta funcionalidad debe utilizar un árbol binario de búsqueda (no es necesario que esté balanceado) y la interface comparable o comparator para cada uno de los criterios. Puede utilizar el código del taller 03.

Parte 04 – Gestión de Logística

1. El sistema debe tener una opción que presente en pantalla un grafo con todas las plantas y la información de las rutas entre ellos **de manera visual (es decir el usuario debe poder ver cada planta como una arista y cada camino como una línea que conecta dichas aristas)**.
 - a. Por simplicidad puede mostrar un grafo inicial, dejar que el usuario acomode los nodos y los caminos con **"drag & drop"** o cualquier otra
2. El usuario debe poder en la misma pantalla que visualiza el grafo, elegir un insumo de una lista desplegable y la interface debe mostrar la siguiente información
 - a. con un color destacado cuales son las plantas que necesitan dicho insumo porque se encuentra debajo del punto de reposición.
 - b.Cuál es el mejor camino para ir desde el acopio del puerto al acopio final, pasando por todas las plantas que necesitan el insumo (el camino puede incluir plantas que no necesiten el insumo, teniendo en cuenta como mejor camino: **la distancia o la duración** (sin tener en cuenta la restricción de peso de las rutas).
3. Dadas dos plantas cualesquiera, una inicial y una final, el usuario debe poder consultar todos los caminos entre ellas y mostrarlos por pantalla. Además, debe mostrar visualmente información de cada uno de los caminos (duración total, distancia total, cantidad máxima de peso).

Parte 05 – Gestión de General

1. El sistema debe poder calcular cual es el flujo máximo que se puede enviar a través de la red, desde el acopio inicial, al acopio final, sin dejar nada en ninguna planta intermedia, teniendo en cuenta la capacidad máxima de transporte en kilos que permite cada camino.
2. El sistema debe mostrar también una lista ordenada de las plantas, según el "Page Rank" de las mismas. Esto implica que una planta es más importante que otra, según la cantidad de caminos que lleguen a ella.
3. Calcular mejor selección de envío

- a. El sistema debe mostrar una lista de todos los insumos faltantes en cada una de las plantas, y la cantidad del mismo.
- b. El sistema debe mostrar todos los camiones disponibles e información del camión
- c. El usuario puede seleccionar uno de estos camiones y presionar un botón generar solución. La solución le va a indicar cual o cuales son los insumos que conviene que el camión transporte, sujeto a el peso máximo que puede transportar el camión y buscando como objetivo maximizar el valor de los insumos transportados (el valor de un insumo transportado se mide a través de su costo). **Debe usar programación dinámica o back tracking para resolver este problema.**

Entrega

MODALIDAD DE ENTREGA

- Subir el archivo comprimido al Campus Virtual en la tarea “Entrega Trabajo Práctico”.
- Subir el archivo al repositorio GIT donde realizó los trabajos prácticos y enviarlo por e-mail a la dirección **died.frsf@gmail.com**, indicando en el campo asunto "TP – Entrega grupo XX" (donde XX indica el número de grupo) y la URL del repositorio GIT donde está el trabajo.

CONSULTAS

Las consultas deben realizarse en el “Foro de consultas sobre el Trabajo Práctico”.

Para aquellos problemas específicos que se presenta en la resolución particular de un grupo y que no estén asociados al resto de los grupos o a aclaraciones del enunciado, pueden realizar la consulta vía e-mail a la dirección de correo electrónico de la cátedra indicando en el asunto: "TP - Consulta grupo XX" (donde XX indica el número de grupo).

IMPORTANTE

- TODOS los ítems solicitados serán tenidos en cuenta para la corrección del trabajo práctico, por lo que es obligatorio resolverlos.
- Cualquier programa entregado que no se ejecute correctamente por utilización de librerías propietarias automáticamente entra en la categoría de no aprobado.
- La entrega y recepción satisfactoria de la resolución es responsabilidad de cada grupo.
- Se debe poder verificar la implementación de todas las funcionalidades solicitadas.
- Se debe verificar la legibilidad de los documentos y diagramas entregados.
- El proyecto entregado debe haber pasado satisfactoriamente las pruebas que se incluirán en un Anexo PRUEBAS del presente trabajo práctico.
- El trabajo debe ser realizado en forma conjunta por TODOS los integrantes del equipo, y debe ser un trabajo personal.
- En el turno de agosto se fijarán fechas para realizar el coloquio de defensa. En este coloquio tienen que estar en forma concurrente todos los integrantes del equipo. Se le podrán solicitar nuevas pruebas, explicaciones o modificaciones sencillas.