



VirtualMall [Fase 3]

Primer Semestre 2021

Ing. Jesus Guzman, Ing. Alvaro Hernandez, Ing. Luis Espino

Marvin Martinez, Andree Avalos, Randolph Muy, Jorge Salazar, Josué Pérez, José Véliz
Universidad de San Carlos de Guatemala

Facultad de Ingeniería

Estructuras de Datos



Visión general

A raíz de la crisis sucedida por la pandemia y la casi imposible movilización para poder realizar cualquier tipo de compras, la empresa EDDsystems le ha encargado a usted como desarrollador crear un proyecto que sea capaz de gestionar tiendas simulando el acceso a centro comercial con los comercios que este posee de manera segura y confiable.

VirtualMall será un aplicación que le permita a los clientes interactuar con las tiendas que ellos deseen de manera eficaz y segura, para esto, el sistema debe ser capaz de manejar los distintos servicios que las tiendas brindarán a sus clientes, como lo son las ventas bajo pedido y los posibles envíos a distintas locaciones, también tener la capacidad de guardar toda la información que se necesite de las tiendas, sus productos, sus servicios y sus clientes, estos deben ser almacenados de forma segura y confiable.

Objetivos

1. Que el estudiante se familiarice con el lenguaje Go
2. Que el estudiante conozca el funcionamiento de peticiones REST mediante la implementación de un servidor web en Go
3. Que el estudiante sepa envolverse en el ámbito de manejo de la memoria
4. Comprender y desarrollar distintos tipos de estructuras no lineales
5. Familiarizarse con el manejo de lectura y escritura de archivos JSON
6. Familiarizarse con el uso de Git.
7. Que el estudiante se familiarice con el lenguaje Javascript y sus frameworks.

Especificaciones

Se solicita continuar la aplicación de servidor realizada en la fase 2 agregando cambios y nuevas funcionalidades. En esta parte necesitamos páginas web que se puedan conectar al servidor, el frontend tiene que estar desarrollado en Javascript utilizando cualquier tipo de framework que el estudiante desee utilizar (recomendaciones: Angular o Reactjs), o también puede utilizarse Javascript sin frameworks. La parte del frontend también tiene que estar en el mismo repositorio que el servidor y se desea que sea intuitivo y fácil de usar para todos los tipos de usuarios.

Modificaciones

Usuarios

Para esta fase del proyecto se necesita manejar diferentes tipos de usuarios por lo cual la parte de carritos de compras cambiará de manera que se pueda tener un registro del usuario que está realizando el pedido.

Productos

Los productos tendrán un nuevo campo que nos permite identificar el lugar de almacenamiento dentro de la tienda, esto nos ayudará para realizar el tracking del producto para llegar al sitio de entrega.

```
{
  "Nombre": "s8",
  "Codigo": 1234,
  "Descripcion": "El smartphone del futuro",
  "Precio": 2500.00,
  "Cantidad": 25,
  "Imagen": "https://i.blogs.es/7a4489/galaxy-s8-4/450_1000.jpg",
  "Almacenamiento": "Celulares 1"
}
```

Cuentas

La aplicación debe manejar los diferentes tipos de cuentas para poder tener una jerarquía dentro de la aplicación que se detalla a continuación.

Administradores

Este tipo de cuenta permite realizar toda la carga de los archivos de entrada para llenar las estructuras, visualizar los reportes de las nuevas fases y de las anteriores. Para el usuario administrador por defecto se van a utilizar los siguientes datos:

- DPI: 1234567890101
- Correo: auxiliar@edd.com
- Password: 1234

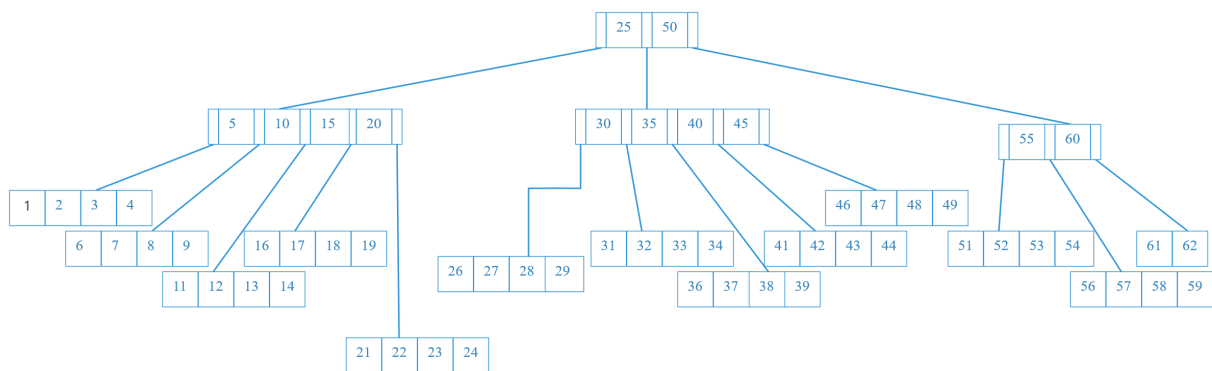
- Nombre: EDD2021

Nota:

Este usuario va a ser el que va a realizar las cargas iniciales.

Cientes

Este tipo de cuenta permite visualizar las tiendas, ver los productos y realizar un pedido de entrega.



Los campos utilizados para las cuentas son los siguientes:

- DPI
- Correo
- Password
- Nombre

La estructura del json se puede encontrar en el siguiente link:

[Usuarios.json](#)

Creación

La aplicación debe permitir registrar nuevos usuarios con cuenta Cliente, por lo cual debe realizar una vista para poder ingresar todos los datos y que se agregue al árbol de cuentas.

Eliminación

La cuenta de un usuario se puede eliminar en cualquier momento, ingresando la contraseña para confirmar la solicitud.

Nota:

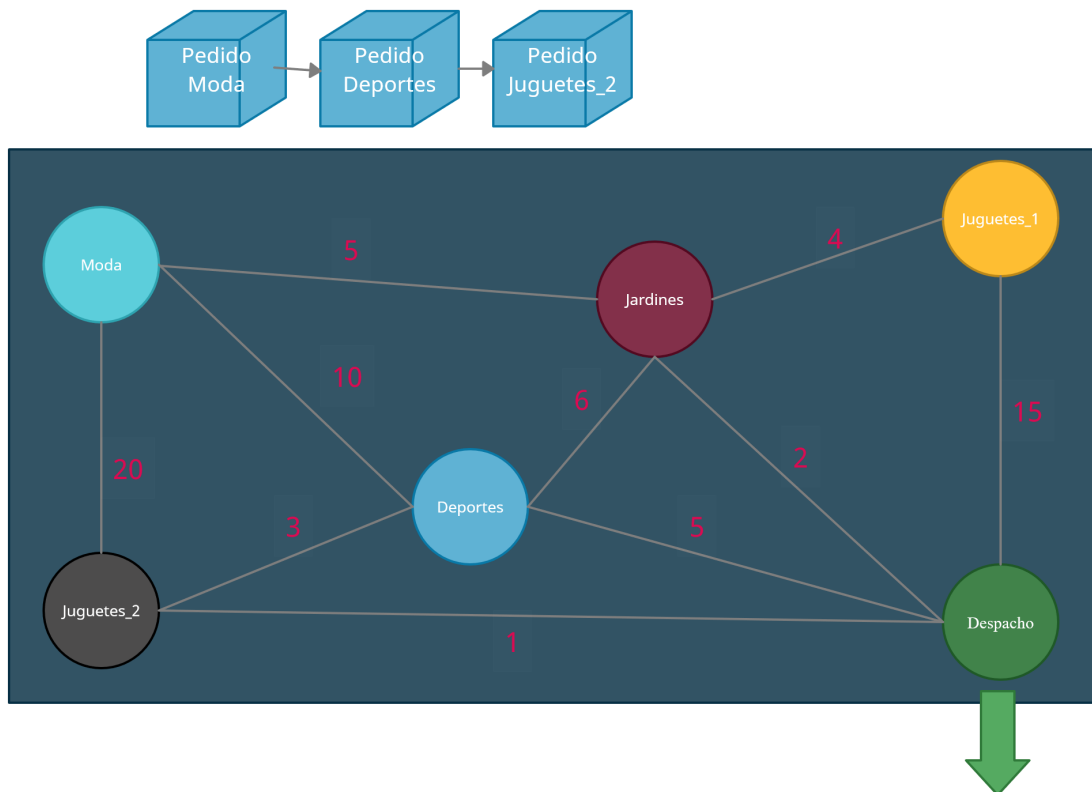
Todas las cuentas van a ir en un mismo árbol, se debe manejar los permisos para que una cuenta no pueda acceder a otras funcionalidades. El grado del árbol B va a ser de grado 5.

Para no eliminar los registros de los pedidos anteriores cuando se elimina un usuario, se debe mostrar "Anónimo" cuando se consulte la información de algún pedido y la cuenta ya no exista.

Servicio de paquetería dentro de los almacenes de Virtual Mall

Virtual Mall maneja sus propios almacenes para poder enviar los productos que llegan por medio de los pedidos, dichos productos son proveídos por las tiendas asociadas, por lo que ahora se cuenta con montañas de productos. Por lo tanto se necesita llevar un orden para poder clasificar los productos; De modo que dentro de los almacenes de virtual mall, los productos son clasificados y distribuidos por departamento.

El sistema de Virtual Mall maneja un apartado de tracking donde se podrá observar la ruta que realizó cada producto para poder ser despachado. Esto se maneja con un grafo en el que el peso de las aristas será la distancia en metros de los departamentos. como se muestra en la siguiente imagen

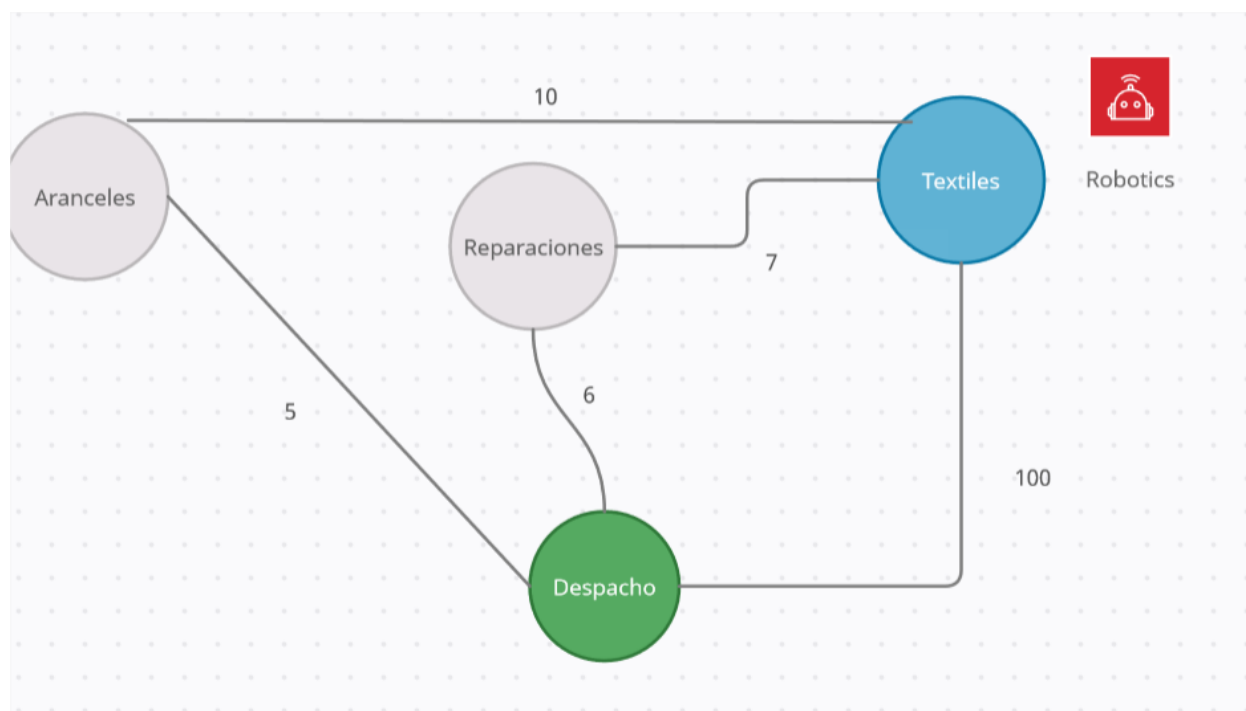


En la imagen anterior se puede observar el sistema que maneja Virtual Mall para el despacho de los productos en relación a los pedidos solicitados por los clientes. Como se puede observar cada departamento es un nodo que está asociado a otro departamento por medio de una arista con un valor, simulado así el camino disponible por donde puede pasar el producto y el esfuerzo que se debe realizar para transportar los productos respectivamente. Por lo tanto se solicita aplicar teoría de grafos para poder encontrar la ruta más corta, logrando con ello una eficiencia en la entrega de pedidos por parte de los almacenes de virtual mall.

Algoritmo

Al iniciar la aplicación se debe configurar la estructura interna para tener un conocimiento del grafo, por lo cual se va a cargar el siguiente archivo:


[Grafo.json](#)



El archivo de entrada cuenta con una lista de nodos los cuales tienen el nombre del nodo y un arreglo de enlaces para poder conectar los caminos con sus respectivos pesos. También trae el nodo inicial en el cual se va a posicionar el robot para iniciar su recorrido.

Recorrido

Para el recorrido del robot siempre va a iniciar en su punto inicial, luego se le va a indicar la lista de productos que tiene que pasar a recoger. El robot tiene que buscar el producto más cercano de su lista de productos e iniciar a desplazarse para recogerlo, luego si aún le faltan productos de su lista debe realizar lo mismo, buscar el más cercano y moverse al



nodo. Luego de que tiene todos los productos debe de buscar el camino más corto para entregar los productos y por último luego de entregarlos se debe regresar a su punto de origen.

Nota:

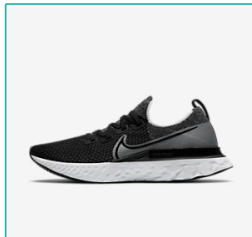
Todos estos recorridos se deben almacenar para indicar los caminos que tomó el robot para posteriormente mostrarlo en el reporte del camino más corto.

La metodología para el despacho de pedidos es el siguiente:

1. Ingreso de productos al carrito de compras

[Inicio](#) | [productos](#) | [Carrito](#)

Carrito de Compras



Q.800

Quitar



Q.800

Quitar



Q.800

Quitar



Q.800

Quitar



Q.800

Quitar



Q.800

Quitar

Comprar

La momento que el cliente decide realizar una compra de los productos seleccionados y guardados en el carrito de compras, automáticamente se realiza una solicitud de pedido de los productos solicitados, por lo tanto se deben realizar las respectivas validaciones sobre los productos solicitados para que haya en existencia y el servidor pueda retornar un mensaje sobre la solicitud de dichos productos. Por lo tanto al analizar el pedido se obtiene todos los productos que solicitó dicho cliente y entra en funcionamiento la

obtención de la ruta más corta por medio del grafo dirigido como se mencionó con anterioridad.

2. Despacho de productos en relación a los pedidos solicitados.

En el paso 1 se describe la manera en que son efectuados los pedidos de productos por parte de los clientes, por lo tanto en el paso dos se obtienen dichos pedidos y son analizados dependiendo del departamento al que pertenecen. Por lo tanto a continuación se describe otra serie de pasos importantes para la distribución de pedidos.

- a. Verificar la categoría o departamento del producto
- b. Ubicarse en el nodo que representa al departamento de producto
- c. Realizar el proceso de la búsqueda de la ruta más corta
- d. enviar el pedido por el nodo de despacho

Seguridad

Para la seguridad en esta fase se plantea la posibilidad de encriptar la información más sensible, esto para evitar el robo de la información asociada a los servicios que se brindan en el sistema, para esto se realizará la implementación de un algoritmo de encriptación simétrica con el cual toda esta información se transformará, y para poder visualizarla el usuario que trate de hacerlo deberá contar con una llave que le permita el acceso a la misma.

Esto será implementado de la siguiente manera, para la encriptación de contraseñas se debe realizar utilizando "Sha256", esto posibilitará que se encripte las contraseñas de una manera segura.

Ahora bien existe además otra información que es sensible como el DPlly correo, estos datos deben ser encriptados también utilizando otro algoritmo, como posible sugerencia puede ser con la librería Fernet en Go, pero el uso de las librerías, se deja libre para esta parte de la encriptación.

La forma de proceder es que será solicitado en el usuario administrador la contraseña maestra para la encriptación y esta contraseña será la utilizada al momento de realizar dicha encriptación, esto puede ser recibido en una entrada de texto en el Frontend.

Reportes

Este módulo solo puede ser accedido por el usuario administrador, en esta área se deben visualizar todos los reportes en la interfaz, también se deben mostrar los siguientes:

Árbol de cuentas (sin Cifrar)

Se debe visualizar el árbol B con todas las cuentas mostrando toda la información necesaria dentro del nodo. El reporte queda a discreción del estudiante, puede utilizar graphviz para generar la imagen o utilizar librerías externas para mostrar el árbol.

Árbol de cuentas (Cifrado)

Este reporte es el mismo que el árbol normal de cuentas, con la diferencia que toda la información del árbol debe ir cifrada con el algoritmo que el estudiante decida utilizar.

Árbol de cuentas (Cifrado sensible)

Este reporte es el mismo que el árbol normal de cuentas, con la diferencia que la información sensible del árbol, como contraseñas, el DPI o correo, debe ir cifrada con el algoritmo que el estudiante decida utilizar.

Nota:

Se va a validar que el estudiante esté realizando el cifrado de datos correspondientemente.

Reporte de camino más corto


Este reporte se debe de mostrar utilizando los pedidos realizados por el cliente, se debe graficar los caminos por el cual tiene que pasar el robot para llegar a recoger todos los productos para finalizar en el área de Despacho. Este reporte puede ser realizado con graphviz pintando los nodos por los cuales tiene que pasar e indicando el recorrido o mostrando de manera tabulada los caminos que debe realizar.

Reportes anteriores

Se debe habilitar la visualización de los reportes de las fases anteriores, árbol AVL, estructura linealizada y la sección de visualizar los pedidos por fecha, mes y la matriz de pedidos.

Consideraciones

- I. Lenguaje a utilizar será **Go y Javascript**
- II. **Puerto** por defecto **3000** para el lenguaje **Go**, y para **Javascript** es **libre**.
- III. Las estructuras serán realizadas por el estudiante.
- IV. IDE a utilizar **Goland** o **Visual Studio Code**.
- V. La entrega será por medio de la plataforma de **UEDI**.

- 
- VI. El estudiante debe tener un repositorio privado en github con el nombre **EDD_VirtualMall_#Carné** y agregar a su tutor como colaborador al repositorio del proyecto (Cada tutor les hará llegar su usuario)
 - VII. Será calificado del último commit realizado antes de la fecha de entrega.
 - VIII. Las copias totales o parciales **serán penalizadas con nota de 0 puntos** y reportadas ante la escuela de ciencias y sistemas.
 - IX. Todos los reportes deben ser generados con graphviz.
 - X. **Fecha de entrega: 17 de abril de 2021 antes de las 23:59 horas.**