



VirtualMall [Fase 2]

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 1 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

Logo de la empresa

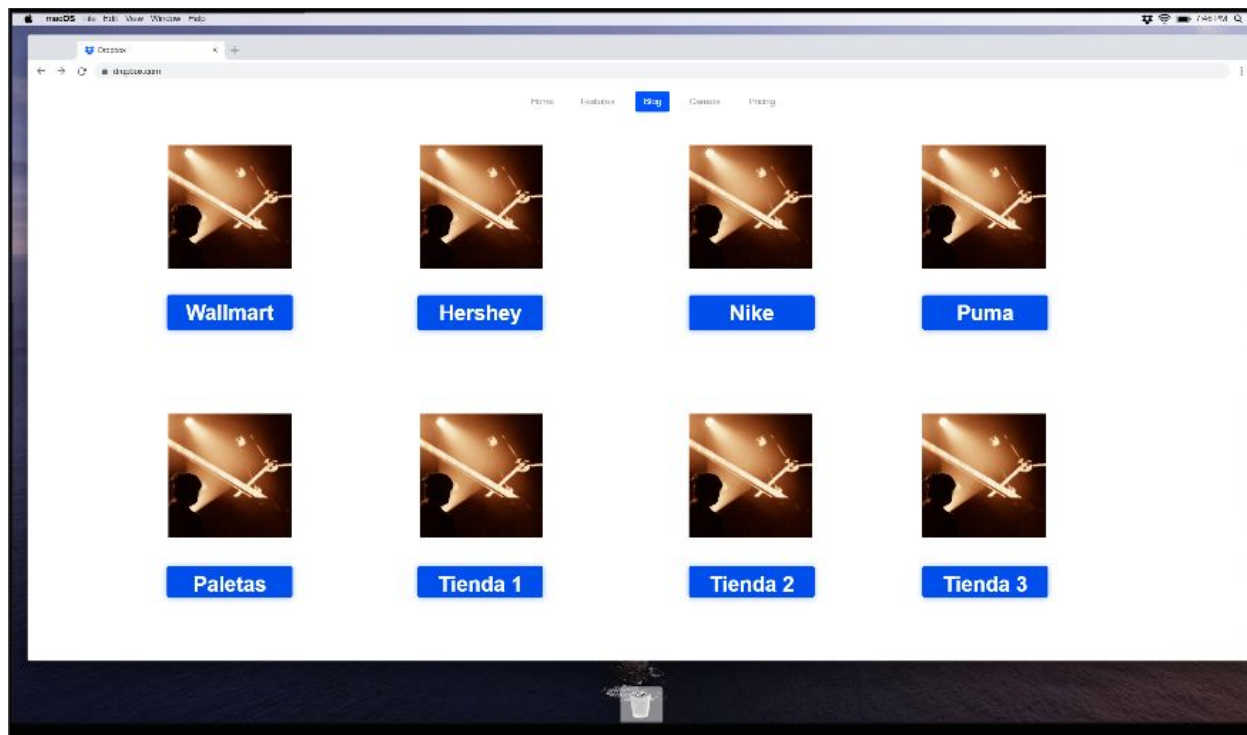
La modificación aplica para el archivo de entrada de las tiendas y se necesita agregar un nuevo campo a las tiendas ya que contamos con una página de catálogo y es necesario ver el logo de la empresa para identificarla más fácilmente, la nueva estructura será de la siguiente manera:

- Nombre
- Descripción
- Contacto
- Calificación
- Logo

El nuevo archivo de entrada es el siguiente:

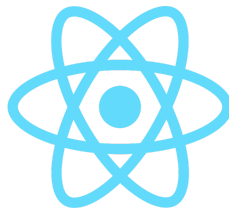
<https://drive.google.com/file/d/1MksyVcPTgQwrJcwjxh5Xt-2mP1bfOGW1/view?usp=sharing>

Nota: queda a discreción del estudiante si descarga la imagen de la url y la almacena en el servidor o consulta la URL siempre que sea necesario.



Cliente

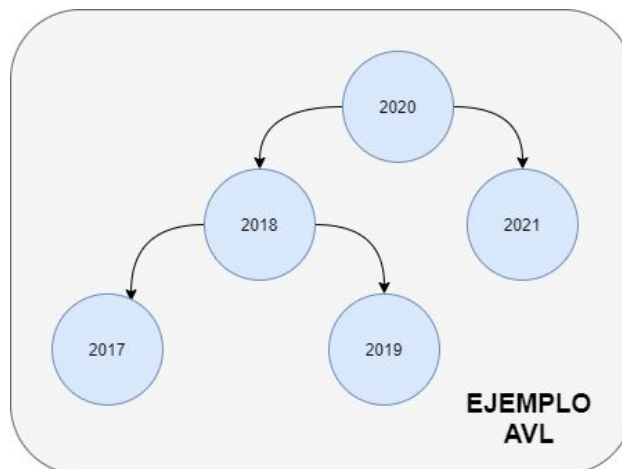
Para la visualización de la información del mall se desarrollará una aplicación web que consuma los servicios realizados en el server de GO el cual estará desarrollado con javascript o con algún framework como por ejemplo **Angular** o **ReactJS**.



Inventario

Para el manejo de inventario, se utilizará la estructura de árbol AVL, donde cada nodo representa 1 producto y deberá guardar la siguiente información:

- Nombre
- Código
- Descripción
- Precio
- Cantidad
- Imagen



Los datos se cargarán al sistema por medio de un archivo JSON con la siguiente estructura:

```
...{
  "Nombre": "algunNombre",
  "Codigo": 23135,
  "Descripcion": "Some description",
  "Precio": 12.00,
  "Cantidad":1231
}...
```

***Nota:** Todos estos productos son correspondientes a las tiendas.

Si la tienda no tiene ningún producto cargado, se deberá crear el árbol e insertar normalmente, si ya existen productos en su inventario, se deberá buscar si el producto que se desea ingresar ya existe y solo aumentar la cantidad, de lo contrario se ingresará al árbol.

Cada vez que se efectúe una compra se deberá buscar el producto, verificar si hay en stock (cantidad) y restar la cantidad que se está comprando. Si no hay en stock no se debe eliminar el producto del árbol únicamente se tomará como cantidad 0.

Carrito de Compras

Para el manejo de las compras de los usuarios, se tiene el carrito de compras, este apartado permitirá visualizar los productos que se han seleccionado previamente.

El funcionamiento de esta vista es el siguiente, luego de que el cliente ha seleccionado todos los productos que desea comprar, podrá entrar al carrito de compras, dentro podrá visualizar un listado de productos, además que tendrá una serie de opciones que le permitirán al cliente tener un mayor control sobre las compras.

Las opciones son las siguientes, en primer lugar puede eliminar productos que no desee comprar por cualquier motivo, en segundo lugar se tendrá la confirmación del carrito de compras, esto hará que todos los productos previamente seleccionados pasen a conformar un pedido, por último se tiene la opción de seguir comprando, lo que le permitirá al cliente retornar de nuevo a donde se encuentren los productos para poder seguir realizando las compras.

| Carrito de Compras | | | |
|---------------------------|----------|----------------|----------|
| Productos | Cantidad | Costo | |
| Videojuego Fallout 4 | 1 | Q255.00 | Eliminar |
| Computadora Dell Inspiron | 1 | Q3275.00 | Eliminar |
| Audifonos Logitech | 2 | Q250.00 | Eliminar |
| Total | | Q3780.00 | |
| Seguir Comprando | | Generar pedido | |

Algo que debe aclararse es que si un producto es seleccionado para pasar al carrito de compras, se debe verificar que cuente con existencias, ya que si no cuenta con existencias debe mostrar un mensaje de error, indicando la falta de existencia.

Vista Calendario-Pedidos

En esta vista el administrador puede ver todos los pedidos que se han hecho en determinada fecha, para esto le han pedido desarrollar una interfaz agradable, robusta y que pueda interactuar con el contenido, **para esto el estudiante debe considerar cuál es la mejor opción para cada caso exceptuando la matriz dispersa que debe ser usada para guardar toda la información de los pedidos de cada día almacenados en una cola dentro del nodo de la matriz; para el encabezado superior se debe usar el día y la categoría para el encabezado lateral, simulando así el mes que se desee visualizar.**

Ejemplo de Vista Web:



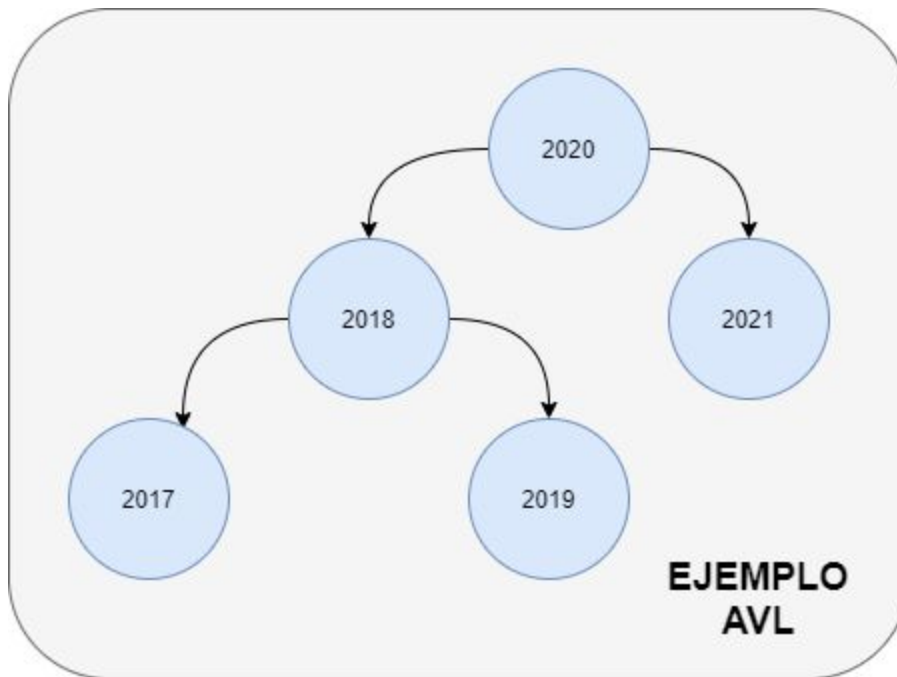
Se podrá guardar la información de las diferentes estructuras, ya sea los años, los meses, la matriz del mes y los pedidos por dia(cola de pedidos), así también poder visualizar los pedidos en una tabla con la información necesaria para poder entender la orden.

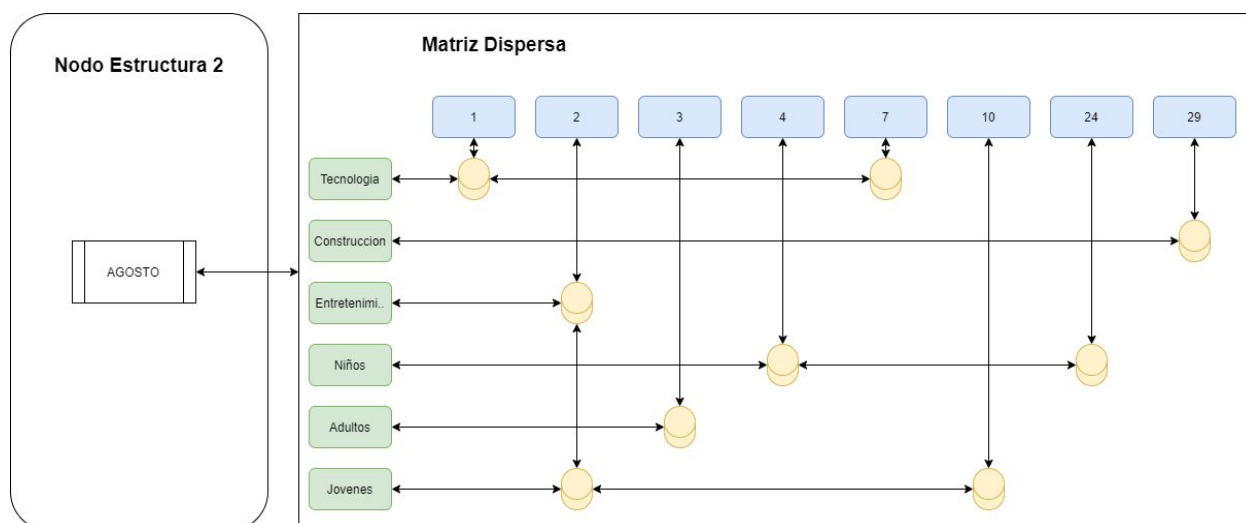
Nota: Los elementos dentro de la matriz deben ser seleccionables para poder mostrar sus detalles, ya sea en el tree de los años como en los nodos de la matriz.

Las líneas solo son guía para saber dónde está cada nodo matriz con su cola de pedidos.

Los reportes deben ser generados por graphviz y deben ser descargados desde la página web (No es permitido rutas quemadas, de lo contrario se tomará como malo este apartado), pueden tomar como base las estructuras mencionadas a continuación.

Ejemplo de Estructuras





Cargas

Cargar Pedidos

Esta función permite cargar el registro de todos los pedidos que se tienen que realizar (esta función es para la Fase 3). El archivo contiene una serie de pedidos indicando la Fecha y una lista de productos.

Archivo de entrada:

https://drive.google.com/file/d/16KnzRRGr6Ym1pfOojo_IO0tgZlbSdBYg/view?usp=sharing

Nota: si el código del producto no existiera se debe eliminar el registro del pedido para que no exista ningún dato invalido.

Cargar Inventarios

Esta es una función por la cual se cargará de forma masiva, los productos que tiene a disposición cada una de las tiendas que ofrece virtual Mall, por lo tanto es importante establecer un formato en el archivo de entrada, de manera que facilite la comprensión y la manipulación de los datos.

Archivo de entrada.

https://drive.google.com/file/d/1UPrRVgbPBChP_JNv_o71gPFbISacleoa/view?usp=sharing

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: 20 marzo de 2021 antes de las 23:59 horas.**