

Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Estructuras de Datos
Vacaciones Primer Semestre de 2023
Proyecto

Ingeniero:

- Ing. Luis Espino

Auxiliares:

- Cristian Suy



EDD Creative

Proyecto Único - FASE I

Ciudad de Guatemala, 01 de Junio de 2023

Objetivos

Objetivo general:

- Aplicar los conocimientos del curso Estructuras de Datos en el desarrollo de las diferentes estructuras de datos y los diferentes algoritmos de manipulación de la información en ellas.

Objetivos específicos:

- Utilizar el lenguaje Go para implementar estructuras de datos lineales
- Utilizar la herramienta Graphviz para graficar las estructuras de datos.
- Definir e implementar algoritmos de ordenamiento, búsqueda e inserción en las diferentes estructuras a implementar.

Resumen de estructuras a utilizar:

FASE I

- Lista Doble Enlazada: Listado de Imágenes disponibles
- Lista Simple Enlazada: Empleados
- Lista Circular Enlazada: Clientes dentro del sistema
- Cola: Clientes en espera para tomar pedido
- Pila: Bitácora de cliente y pedido de imagen
- Matriz Dispersa: Capas de la imagen

Definición del problema

La empresa EDD Creative, desea que usted como programador realice una aplicación de escritorio capaz de generar imágenes en píxeles, esto debido a que imágenes con extensiones generales como jpg, png, svg, llegan a tener un peso grande en almacenamientos de la nube, por lo cual se opta a generar imágenes creadas por intérpretes de html y css, ya que al ser generadas como código, optimizan el espacio que estos ocupan. Teniendo en cuenta lo anterior, se solicita que como ingeniero en software cree una aplicación capaz de leer archivos csv, los cuales representan las capas de la imagen, estas imágenes pueden ser modificadas aplicando filtros de cada una de las capas según lo solicitado por el cliente. Y mediante mapeo lexicográfico generar los archivos html y css que podrán ser visualizados por los usuarios.

Descripción de la aplicación

FASE I

Login: El sistema contará con un login principal para realizar el inicio de sesión de cada empleado que esté registrado en el sistema, este deberá validar que el usuario y contraseña sean correctos, habrá un usuario especial, el cual será el administrador del sistema este se llamara ADMIN_#Carnet y contraseña Admin, éste será el encargado de realizar la carga masiva de las imágenes y empleados.

```
-----Login-----
1. Iniciar Sesion
2. Salir del Sistema
Elige una opcion: 1
Ingrese usuario: 04933
Password: vacacionesJunio
```

Vista de Administrador: El administrador contará con el siguiente menú donde podrá elegir algunas de las opciones que se explicarán más adelante.

```
-----Dashboard Adminstrador 201700918-----
1. Cargar Empleados
2. Cargar Imagenes
3. Cargar Usuarios
4. Actualizar Cola
5. Reportes Estructuras
Elige una opcion:
```

Administrador: Tendrá como función principal registrar los empleados y las imágenes de prueba que existirán en el sistema, para ello realizará la lectura de un csv con los siguientes campos.

- Imágenes: Nombre de imagen, Cantidad de Capas.
La estructura a utilizar Lista doblemente Enlazada

Imagen	Capas
bmo	3
corrin	3
deadpool	4
hulk	4
mario	6
tux	3

- Empleados: Id, nombre, cargo, contraseña.
La estructura a utilizar Lista Simple Enlazada

id	nombre	cargo	password
6385	Cristian Suy	Ventas	6385_Ventas
5372	Hector Jimenez	Ventas	5372_Ventas
4399	Pablo Velasquez	Diseño	4399_Diseño
1229	Jaquelin Gomez	Diseño	1229_Diseño
3607	Yadira Ruiz	Diseño	3607_Diseño
3518	Paula Fuentes	Ventas	3518_Ventas

- Clientes: Id, nombre

La estructura a utilizar Lista Circular Enlazada

id	nombre
4435	Cristian Suy
5268	Hector Jimenez
2503	Pablo Velasquez
4975	Jaquelin Gomez
1280	Yadira Ruiz
3101	Paula Fuentes

Aparte tendrá un apartado de lectura de cola actual, con los campos ID y Nombre, si el ID posee una X se considera que el cliente atendido actualmente es nuevo cliente y deberá almacenarse en el sistema para futuras compras. La estructura a utilizar es una cola.

id	nombre
2111	Cristian Suy
X	Hector Jimenez
3220	Joel Guzman
2411	Karen Garcia
X	Teodoro Mejia
5257	Alejandra Perez

Sistema Principal: Cuando el empleado a cargo del pedido realice el inicio de sesión contará con un menú especial, el cual tendrá las siguientes opciones:

- Ver imagenes: Mostrará el nombre de las imágenes que estén registradas en el sistema, al seleccionar una imagen se deberá generar la visualización previa de la imagen, esta sección será explicada más adelante
- Realizar pedido: Este apartado solicitará el id del cliente atendido actualmente, ID del empleado que atendió pedido, nombre de la imagen o id. (Nota, si el cliente atendido no está registrado en el sistema, se procede a guardar mediante un identificador único y deberá retornar el id, para entrega del cliente) La estructura a implementar será una pila para esta seccion.

```

-----EDD Creative 79852-----
1. Ver Imagenes Cargadas
2. Realizar Pedido
Elige una opcion:

```

Generación de Imagen: El sistema contará con la generación de la imagen principal, para ello se leerá un csv llamado **inicial.csv** este contará con las configuraciones de la imagen y la capa que contiene la imagen. se manejará el numero de capa en orden ascendente, es decir la capa 1 representa la capa mas profunda de la imagen, mientras que la capa con el numero mas alto representa la capa que se encuentra en el tope de la imagen. (NOTA: la capa 0 esta reservada para el archivo de configuración.).

Layer	File
0	config.csv
1	body.csv
2	hair.csv
3	shirt.csv
4	dungarees.csv
5	boots.csv

Archivo de configuración: El archivo de configuración contendrá 2 columnas: Config y Value, las cuales representan el atributo en config y el valor de dicho atributo en value, se manejaran 4 atributos:

- Image width (ancho de la imagen)
- Image height (alto de la imagen)
- Pixel width (ancho de pixel)
- Pixel height (alto de pixel)

config	value
image_width	16
image_height	16
pixel_width	30
pixel_height	30

Capas: Cada archivo de capas debe de contener la información correspondiente a cada capa, en cada celda se contendrá la información del color correspondiente a dicho pixel en formato rgb de la siguiente manera: R-G-B, ejemplo: 157-118-230, en caso la celda se encuentre una x significa que en dicho lugar no se encuentra

ningún color o simplemente es “transparente”.

[illegible]

Generación de html base: para generar el html se usará el orden lexicográfico el cual ayudará a convertir las matrices en arreglos lineales y generar el archivo html y css más fácil. El cual tendrá una estructura a la siguiente y dicha plantilla será subida al repositorio para que pueda ser utilizada como base:

Archivo HTML

```
<!DOCTYPE html>
<html>
<head>
  <!-- Link to our stylesheet Painting our Pixel Art -->
  <link rel="stylesheet" href="template.css">
</head>
<body>

  <!-- div container representing the canvas -->
  <div class="canvas">

    <!-- div containers representing each pixel of the drawing (16px x 16px = 256 pixels) -->
    <div class="pixel"></div>
    <div class="pixel"></div>
    <div class="pixel"></div>
    <div class="pixel"></div>
    <!-- ...add as many pixels as needed. -->

  </div>

</body>
</html>
```

Archivo CSS

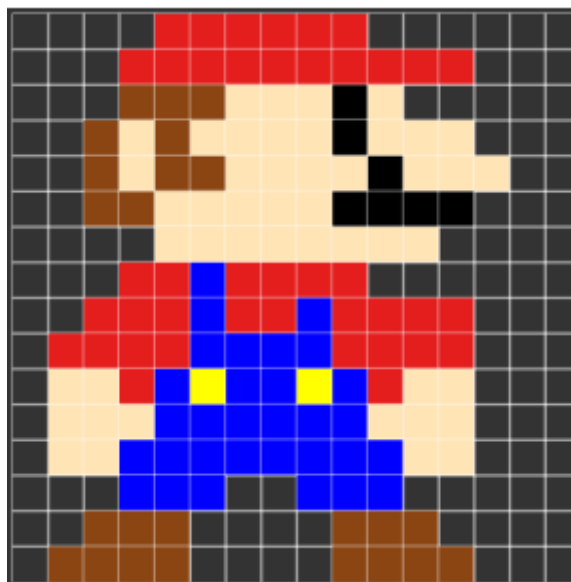
```
body {
  background: #333333; /* Background color of the whole page */
  height: 100vh; /* 100 viewport height units */
  display: flex; /* defines a flex container */
  justify-content: center; /* centers the canvas horizontally */
  align-items: center; /* centers the canvas vertically */
}

.canvas {
  width: 480px; /* Width of the canvas */
  height: 480px; /* Height of the canvas */
}

.pixel {
  width: 30px; /* Width of each pixel */
  height: 30px; /* Height of each pixel */
  float: left; /* Everytime it fills the canvas div it will begin a new line */
  /*box-shadow: 0px 0px 1px #fff;*/ /* Leave commented, showing the pixel boxes */
}

/* list of pixels that will be painted */
.pixel:nth-child(92),
.pixel:nth-child(12) { background: rgb(255,255,255); }
```

Luego de generar el archivo html con su css, tendrá que ser visualizado, otorgando así una vista previa de la imagen, de la siguiente manera:



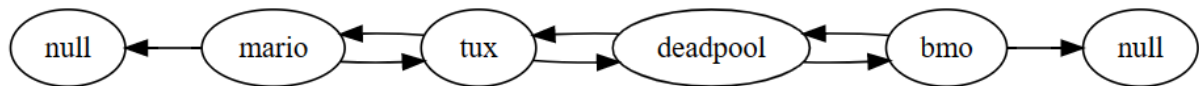
Área de Reportes

Se contará con un apartado especial en el que se puedan mostrar en forma de grafos las estructuras en tiempo real, generando la imagen utilizando el visualizador

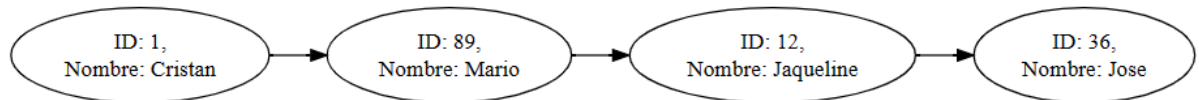
de imágenes del sistema operativo. Los reportes sólo serán generados con la herramienta de Graphviz.

Reporte de Imágenes, Empleados, y Clientes: Este reporte al ser Listas, deben generar mostrando el id y nombre en el caso de empleados y clientes, y para imagenes solo el nombre de la siguiente manera:

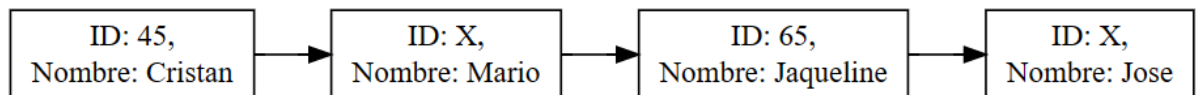
Imágenes:



Empleados y Clientes:



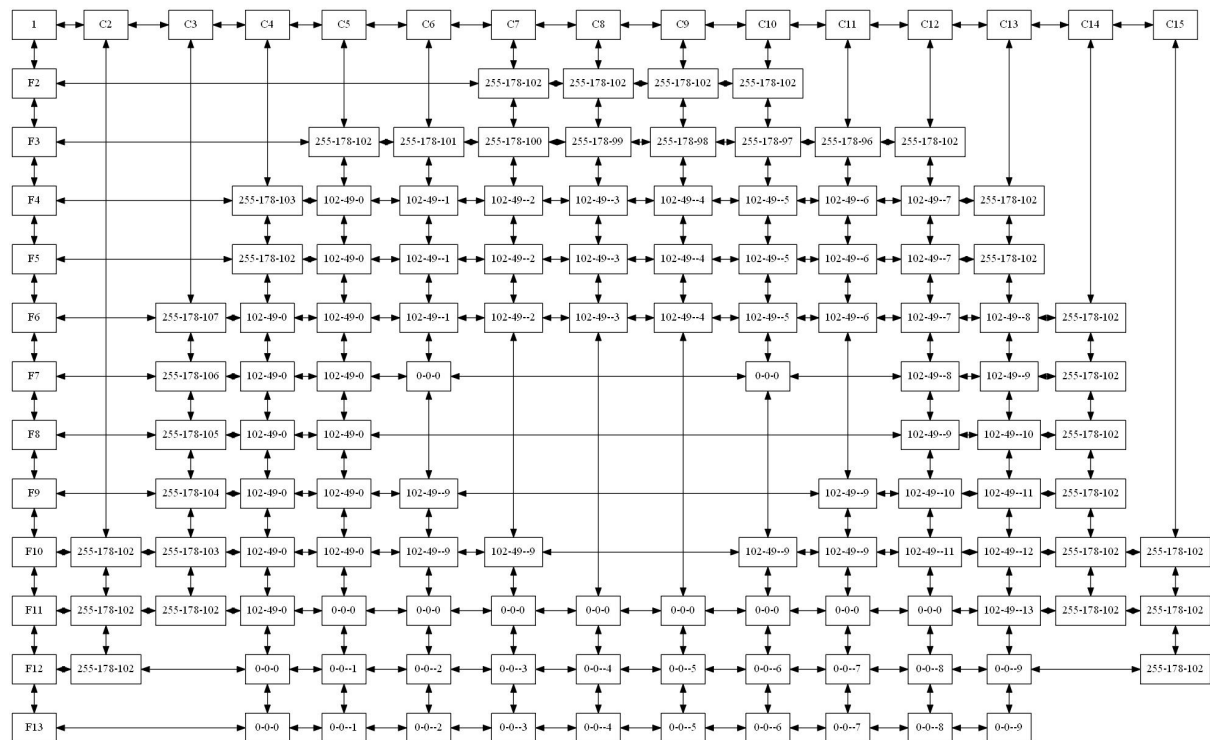
Reporte de Cola: Este reporte representa la cola actual de la tienda, teniendo en cuenta los parámetros ID para las personas que son nuevas dentro de la tienda.



Reporte de Pila: Este reporte representa los pedidos realizados en ese momento.

ID Cliente: 12 Imagen: bmo
ID Cliente: 19 Imagen: mario
ID Cliente: 100 Imagen: ave
ID Cliente: 112 Imagen: bmo

Reporte de Imágenes: Para el caso de las matrices, la aplicación debe ser capaz de generar el reporte de capa de la imagen, para este caso debe solicitar la aplicación que imagen se desea generar el reporte y que capa desea su reporte, tal y como se muestra a continuación.



Reporte JSON: Este reporte sera generado cada vez que se realice un pedido, esto con el fin de usarlo para la segunda Fase como archivo de entrada, este tendra la siguiente estructura.

```
{
  "pedidos":[
    {
      "id_cliente": 12,
      "imagen": "bmo"
    },
    {
      "id_cliente": 12,
      "imagen": "mario"
    },
    {
      "id_cliente": 12,
      "imagen": "deadpool"
    },
    {
      "id_cliente": 12,
      "imagen": "ave"
    }
  ]
}
```

Tecnologías a utilizar

FASE I:

Realizar la aplicación en Golang:

Este contará con una aplicación en consola en el que se deberán de retornar los datos en un formato JSON, se puede utilizar cualquier librería para poder crear el archivo, esto para utilizarlo como base para la segunda fase del proyecto.

Realizar los reportes en Graphviz:

Todos los reportes deberán estar realizados en graphviz y deben estar constantemente generando cuando se realice un cambio, para poder observar de forma visual el estado actual de las estructuras.

Restricciones

- Las estructuras deben de ser desarrolladas por los estudiantes **sin el uso de ninguna librería o estructura predefinida en el lenguaje**.
- Los reportes son esenciales para verificar si se trabajaron correctamente las estructuras solicitadas, por lo que si no se tiene el reporte de alguna estructura se anularán los puntos que tengan relación tanto al reporte como a la estructura en cuestión.
- Se permite el uso de framework o librerías como Angular o React para el desarrollo del proyecto, solamente para el desarrollo de interfaz grafica en Segunda Fase, toda las estructuras deben ser creadas en Go.

Observaciones

- El lenguaje para la fase 1 será Go y todas las salidas mediante Consola
- Herramienta de desarrollo de reportes solamente **Graphviz** y **HTML** para visualizar imágenes.
- La entrega se realizará por medio de Github, el nombre del repositorio debe ser **EDD_VJ1S2023_PY_#carnet**. Y por medio de **UEDI** se hará entrega del link de su repositorio.
- Recordar tener sus repositorios en privado, para evitar copias de código.
- Realizar los **manual de técnico y de usuario** en el README del repositorio
- Toda duda que se tenga durante el proceso, será realizada por medio de los foros de **UEDI**.
- Fecha de entrega:
 - FASE I: Jueves 15 de Junio del 2023 a las 11:59 hrs (Medio Día).