Universidad De San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ciencias y Sistemas

Estructura de Datos
Sección “A”


# “MANUAL TÉCNICO”

Nombre: Samuel Alejandro Pajoc Raymundo.

Carné: 201800665

Auxiliar: José Andrés Montenegro Santos.

# Objetivos

## General:

Brindar al lector una guía que contenga la información del manejo de clases, atributos, métodos y del desarrollo de la interfaz, la cual será mostrada a través de la consola del ID utilizado; para facilitar futuras actualizaciones y futuras modificaciones realizadas por terceros.

## Específicos:

- Mostrar al lector una descripción lo más completa y detallada posible del SO, IDE entre otros utilizados para el desarrollo de la aplicación.

- Proporcionar al lector una concepción y explicación técnica - formal de los procesos y relaciones entre métodos y atributos que conforman la parte operativa de la aplicación.

# Introducción

Este manual técnico tiene como finalidad dar a conocer al lector que pueda requerir hacer modificaciones futuras al software el desarrollo de la aplicación “Social Structure” indicando el IDE utilizado para su creación, su versión, requerimientos del sistema, etc…

La aplicación tiene como objetivo leer tres tipos de archivos de extensión (.json), dichos archivos son cargados a la memoria disponible para el sistema, por medio de un usuario Administrador, el cual es el único capaz de cargar los tres archivos json, dichos archivos son: usuarios, publicaciones y relaciones. Con estos datos se genera una simulación de la funcionalidad básica de una red social, por lo que además de haber datos cargados por el Administrados, es posible crear usuarios, enviar solicitudes, realizar publicaciones y si también lo dese, puede eliminar su cuenta. La aplicación permitirá fue desarrollada con el enfoque de comprender mejor la administración y manejo de memoria RAM. De este modo, se realizarán producciones (programas) más optimos.

# Descripción de la Solución

Para poder desarrollar este proyecto se analizó los requerimientos necesarios para la implementación de listas, tales como pilas, listas simples, listas simples doblemente enlazadas y listas circulares; tomando en cuenta siempre futuros cambios en los entornos en los que se realiza la aplicación.

Entre las consideraciones encontramos con mayor prioridad están:

- El único ente capaz de realizar las cargas de los archivos tipo json es el administrador, para poder acceder al menú, en donde se realizan las gestiones de los datos, tales como las cargas; Se debe iniciar sesión con un correo: admin@gmail.com y una contraseña: EDD2S2024.

- El administrador también podrá ser capaz de visualizar de forma gráfica el flujo de los datos encontrados dentro del programa, los cuales son: grafico de Usuarios, grafico de relaciones de amistad, grafico de publicaciones, y además mostrará los 5 usuarios con más publicaciones y los 5 usuarios con menos amigos.

- Al crear una cuenta, el usuario podrá visualizar un menú especial para usuarios, en el cual podrá ver su información, eliminar su cuenta, enviar solicitudes de amistad, ver las publicaciones de sus amigos y realizar publicaciones.

- El usuario podrá tener acceso a un gráfico de solicitudes enviadas y recibidas, un gráfico de relaciones de amistad, un gráfico de publicaciones, y ver un listado de sus amigos.

# Lógica y Flujo del programa

Para poder realizar la carga masiva de ususarios, se solicita que el Administrador ingrese la ruta donde se encuentre cada documento a cargar.

El administrador es el único capaz de eliminar los usuarios dentro de la aplicación.

Cada nuevo usuario se almacena dentro de una lista simplemente enlazada, cada vez que se crea un usuario nuevo o si estos son cargados por medio del Administrador, sus campos:valor se encuentran dentro de una lista simplemente enlazada, en la cual únicamente se utilizan 2 punteros, el primero llamado “primero” y el siguiente llamado “ultimo”, esto con el fin de llevar un mejor control en la inserción o eliminación de un nodo.

Los nodos de la lista simple son llamados nodoSimple; dentro del contructor de dicho nodo se encuentran los campos de: Nombres, Apellidos, Fecha de nacimiento, Correo, Password (Contraseña), junto a un apuntador llamado siguiente, el cual permite mantener enlazado el dato nuevo a crear.

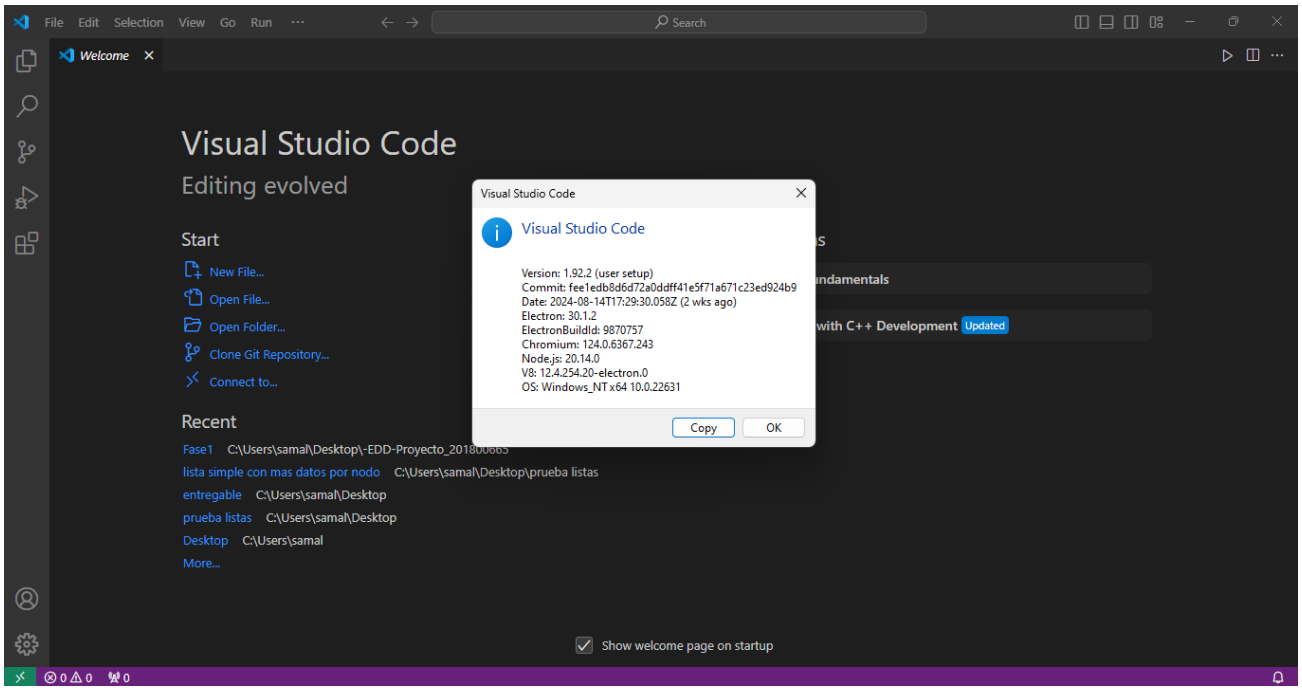
Para poder acceder a cada dato de cada nodo, se crearon sus respectivos métodos setter’s y getter´s (Para mayor detalle verificar lines 14 a 38 del código fuente).

Dentro de la lista simple, se agregan cada nodo con su respectiva información. Los métodos utilizados dentro de una lista simplemente enlazada son: append, buscar, buscar2, eliminarUsuario, eliminarUsuarios y print. Además, dado que el lenguaje utilizado para realizar esta aplicación, permite un amplio manejo de la memoria, es necesario diseñar un método llamado destructor, el cual, al finalizar la ejecución del programa, liberará toda la memoria utilizada dentro del stack y hip. Para diseñar dicho método, únicamente se coloca el símbolo (~), y a continuación el nombre de la clase, similar a un constructor, pero en este caso este se encargará de eliminar los datos ya no utilizados de la aplicación. También es importante mencionar que c++ posee un miniGarbage colector, el cual únicamente libera el espacio utilizado en la declaración de variables, por lo que, como se menciono anteriormente, es necesario implementar el método destructor. Dentro del método destructor, se debe realizar un recorrido entre cada nodo, y luego ir eliminando uno por uno, hasta vaciar la lista.

- El método apped, agrega cada nodo a la lista simplemente enlazada.
- El método buscar, verifica si un usuario existe dentro de la lista simplemente enlazada.
- El método buscar2, buscar el nodo del usuario que inicio sesión para luego ser eliminado, dado que este método se utiliza en dicha acción.
- El método eliminarUsuario, permite eliminar el nodo y por ende la cuenta del usuario que lo desee.
- El método eliminarUsuarios, elimina todos los nodos y por ende todos los usuarios almacenados dentro de la lista simplemente enlazada.
- El método print, permite observar los datos almacenados dentro de la lista simplemente enlazada.

# IDE

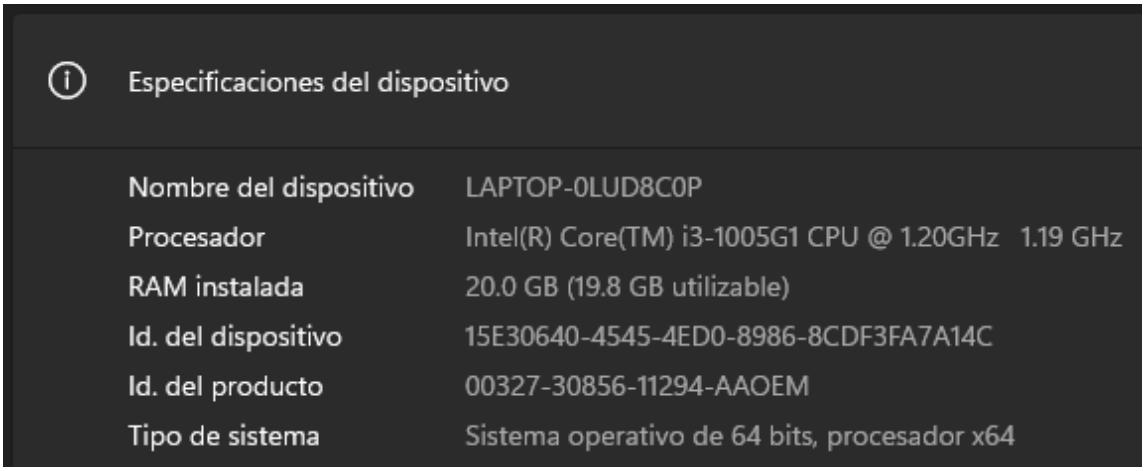
El IDE con el que se desarrolló el proyecto “Social Structure” fue Visual Studio Code versión 1.92.2, debido a que el desarrollo de la aplicación fue más sencilla dato que únicamente se importó un compilador para c++, el cual funciona perfecto con Visual Studio Code, además, dentro del entorno de dicho IDE, se pueden agregar extensiones las cuales apoyan al desarrollador, de tal modo que se pueda aumentar la detección de errores así como aumentar la producción de aplicaciones desarrolladas.




**Requerimientos:**

- Instalar Visual Studio Code.
- Instalar el compilador de c++; Para que los comandos puedan ser ejecutados. Se optó por instalar el compilado: GCC C++ compiler (g++) de mingw-w64.
- Instalar la extensión: C/C++ extensión for VS Code.

- Sistema Operativo:
  El sistema operativo en el que se llevó a cabo la realización del proyecto fue Windows 11 de 64 bits.




# Librerías Utilizadas

Las librerías utilizadas para el desarrollo de este proyecto fueron:

```
#include <iostream>
#include <string>
#include <vector>
using namespace std;
#include <fstream>
```