

Universidad San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ciencias y sistemas
Estructuras de Datos

Ingenieros:

- Ing. Luis Espino
- Ing. Edgar Ornelis
- Ing. Álvaro Hernández

Auxiliares:

- Kevin Martinez
- Carlos Castro
- José Montenegro



Proyecto Fase 2

Social Structure

Implementación de estructuras no lineales

ÍNDICE

Objetivos.....	3
Objetivo general.....	3
Objetivos específicos.....	3
Descripción.....	4
Modificaciones.....	5
Usuarios.....	5
Publicaciones.....	5
Comentarios.....	6
Carga Masiva.....	6
Usuarios.....	7
Solicitudes.....	7
Publicaciones.....	8
Reportes.....	9
Administrador.....	9
Usuario.....	9
Visualización de Reportes.....	9
Observaciones.....	10
Entregables.....	10

Objetivos

Objetivo general

- Aplicar los conocimientos del curso Estructuras de Datos en el desarrollo de soluciones de software.

Objetivos específicos

- Comprender y desarrollar distintos tipos de estructuras no lineales.
- Implementar una aplicación de escritorio utilizando el lenguaje de programación C++.
- Utilizar la herramienta Graphviz para graficar estructuras de datos no lineales.
- Definir e implementar algoritmos de búsqueda, recorrido y eliminación en estructuras de datos.

Descripción

En la primera fase del proyecto, se desarrolló una aplicación de consola que emula el funcionamiento de una red social, utilizando diversas estructuras de datos para gestionar la información de manera eficiente. Esta aplicación permite simular la interacción básica entre usuarios, la publicación de contenido, y otras funcionalidades clave, todo dentro de un entorno de línea de comandos.

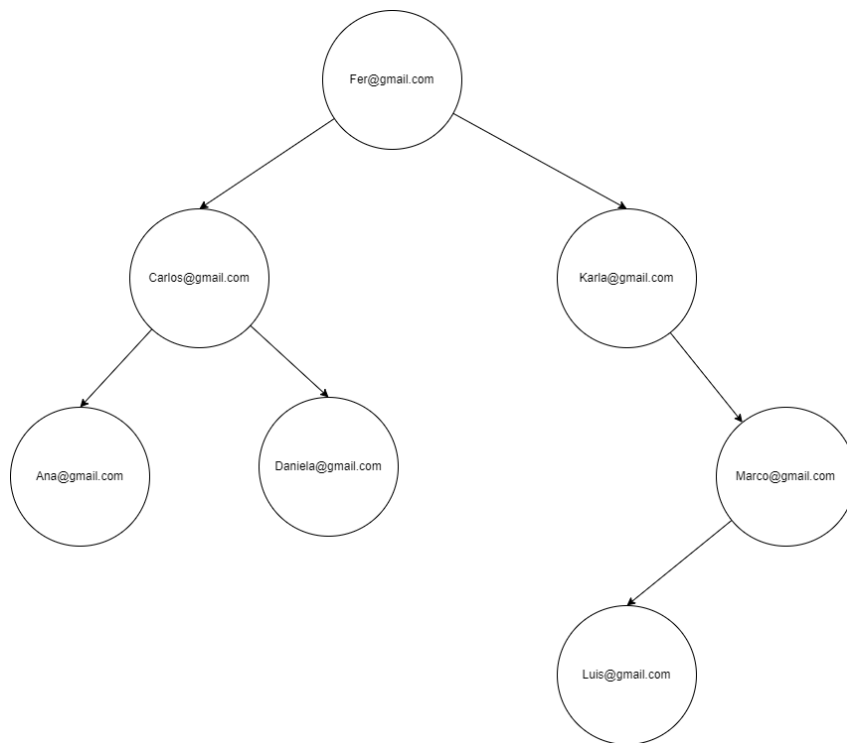
En esta segunda fase, el proyecto evolucionará hacia una aplicación de escritorio con una **interfaz gráfica de usuario (GUI)**, lo que proporcionará una experiencia más intuitiva y accesible para los usuarios. Se migrarán las funcionalidades previamente implementadas en la aplicación de consola a esta nueva interfaz, asegurando que todas las operaciones fundamentales se mantengan y optimicen según sea necesario.

Además, se realizaron ajustes y mejoras en las estructuras de datos empleadas, para adaptarlas a las nuevas exigencias que implica el manejo de una interfaz gráfica. La implementación de esta fase es un paso crucial para el desarrollo del proyecto, ya que representa la transición hacia una aplicación más compleja y robusta, capaz de simular de manera más realista el entorno de una red social moderna.

Modificaciones

Usuarios

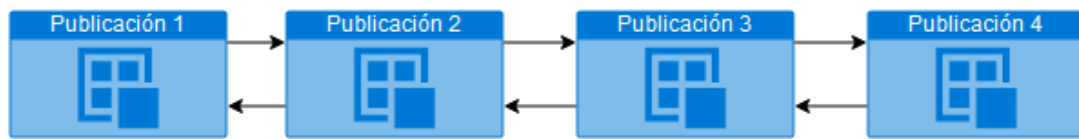
Anteriormente los usuarios se almacenaban en una lista simple, esta estructura será reemplazada por un **árbol AVL**, manteniendo las mismas restricciones de la fase anterior.



Este árbol AVL será implementado para poder optimizar la búsqueda de usuarios, por este mismo motivo se deberá utilizar como criterio para ordenar los nodos del árbol el **correo electrónico** de forma alfabética (A-Z) y cada persona registrada en la aplicación deberá tener la opción de buscar personas a través de dicho correo.

Publicaciones

Al momento de crear una publicación, se mantendrá almacenando en una **lista doblemente enlazada** la cual contendrá todas las publicaciones de la aplicación.



El feed de publicaciones de un usuario será almacenado en un **Árbol Binario de Búsqueda (ABB)**, reemplazando la estructura anterior (lista circular). **El índice de este árbol será la fecha de cada publicación**, permitiendo un orden cronológico de las mismas. Además, cada nodo del árbol contendrá una lista simplemente enlazada que almacenará todas las publicaciones correspondientes a esa fecha específica. Esto permitirá gestionar múltiples publicaciones en un mismo día y mantenerlas organizadas dentro del ABB. El cambio de estructura se debe a que se desea agilizar la búsqueda de publicaciones de una fecha en concreto, **por lo cual la aplicación debe poseer una opción para filtrar publicaciones por fecha.**

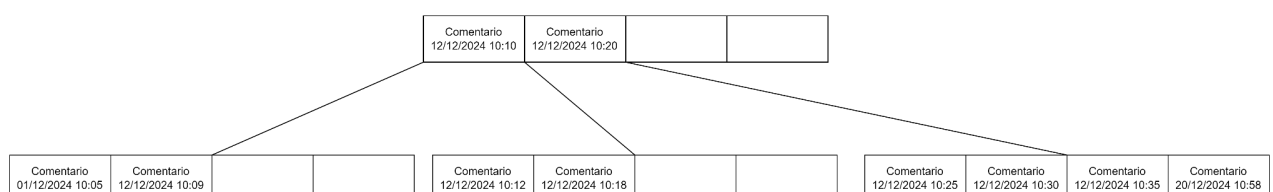
Cada una de estas publicaciones deberá poseer un árbol B, el cual almacenará los respectivos comentarios que se realicen en cada una, la llave que se utilizará como criterio de ordenamiento para este árbol será la fecha en la que se realicen los comentarios.

Además se agrega la función de poder **seleccionar una imagen** para agregarla a las publicaciones.

Comentarios

Para cada publicación, se implementará una funcionalidad de comentarios. Estos comentarios se almacenarán en un **Árbol B de orden 5**, el cual permitirá un acceso eficiente y una estructura balanceada para manejar un gran volumen de comentarios.

Cada nodo del Árbol B contendrá varios comentarios, manteniendo un orden que facilita la inserción de comentarios de manera eficiente. Para insertar en el árbol se debe de tomar como **índice la hora y fecha de inserción**. Por lo tanto, se vería de esta manera:



Carga Masiva

La carga de los datos (usuarios, publicaciones y solicitudes de amistad) se realizará por medio de un usuario administrador el cual tendrá como correo **admin@gmail.com** y como contraseña **EDD2S2024**, por lo tanto al momento de ingresar con este usuario se podrá realizar la carga de los archivos json.

A continuación se muestra un ejemplo de la estructura de los json. El único archivo que cambia respecto a la fase anterior es el json de publicaciones.

Usuarios

```
{} usuarios.json U X
{} usuarios.json > {} 0 > fecha_de_nacimiento
1  [
2    {
3      "nombres": "Juan",
4      "apellidos": "Perez Gomez",
5      "fecha_de_nacimiento": "1990/05/14",
6      "correo": "juan.perez@example.com",
7      "contraseña": "password123"
8    },
9    {
10     "nombres": "María",
11     "apellidos": "Lopez Martinez",
12     "fecha_de_nacimiento": "1985/08/22",
13     "correo": "maria.lopez@example.com",
14     "contraseña": "securepass456"
15   },
16   {
17     "nombres": "Carlos",
18     "apellidos": "Rodriguez Fernandez",
19     "fecha_de_nacimiento": "1992/11/30",
20     "correo": "carlos.rodriguez@example.com",
21     "contraseña": "mypassword789"
22   }
23 ]
```

Solicitudes

```
1  [
2    {
3      "emisor": "maria.lopez@example.com",
4      "receptor": "carlos.rodriguez@example.com",
5      "estado": "ACEPTADA"
6    },
7    {
8      "emisor": "juan.perez@example.com",
9      "receptor": "maria.lopez@example.com",
10     "estado": "PENDIENTE"
11   },
12   {
13     "emisor": "carlos.rodriguez@example.com",
14     "receptor": "juan.perez@example.com",
15     "estado": "PENDIENTE"
16   },
17   {
18     "emisor": "ana.martinez@example.com",
19     "receptor": "maria.lopez@example.com",
20     "estado": "ACEPTADA"
21   }
22 ]
```

Publicaciones

Como cada publicación ahora tendrá comentarios, se agregó un nuevo campo el cual tendrá la lista de comentarios.


```
1  [
2    {
3      "correo": "juan@gmail.com",
4      "contenido": "Hola :)",
5      "fecha": "19/08/2024",
6      "hora": "10:00",
7      "comentarios": [
8        {
9          "correo": "jose@gmail.com",
10         "comentario": "Buen dia",
11         "fecha": "19/08/2024",
12         "hora": "10:00"
13       },
14       {
15         "correo": "maria@gmail.com",
16         "comentario": "Como estas",
17         "fecha": "19/08/2024",
18         "hora": "15:30"
19       }
20     ]
21   },
22   {
23     "correo": "pedro@gmail.com",
24     "contenido": "Tengo hambre",
25     "fecha": "9/07/2024",
26     "hora": "19:00",
27     "comentarios": []
28   }
29 ]
```

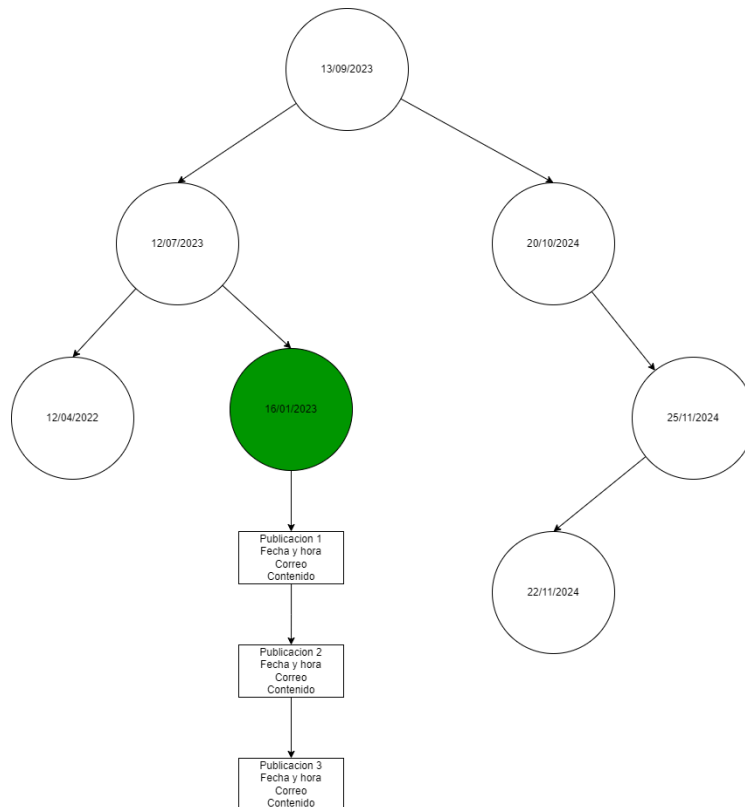
Reportes

Administrador

- **Usuarios:** Árbol AVL de usuarios
- **Filtros:**
 - Buscar usuarios por correo
- **Listar usuarios de la aplicación:**
 - Preorden
 - Inorden
 - Postorden
- **Publicaciones:** lista general de publicaciones(lista doble)

Usuario

- **Publicaciones:** Ver árbol ABB de una fecha en concreto
- **Comentarios:** Cada usuario debe ser capaz de ver los comentarios de cada publicación de su feed.
 - Ver árbol B de comentarios de una publicación concreta.
- **Filtros:**
 - Filtrar publicaciones por fecha
 - Buscar usuarios por correo
- **Ordenar publicaciones en feed:**
 - Inorden limitado
 - Preorden limitado
 - Postorden limitado
- **Reportes**
 - 3 fechas con mayor cantidad de publicaciones
 - 3 publicaciones con mayor cantidad de comentarios
 - Árbol BST: Se deberá graficar el árbol binario de publicaciones propias y de amigos, seleccionando una fecha en concreto, se deberá de graficar la lista de publicaciones que corresponden a esa fecha:



Visualización de Reportes

Todos los reportes generados deben mostrarse directamente dentro de la aplicación, utilizando gráficos que se desplieguen de manera intuitiva y accesible para los usuarios y administradores. No será necesario que el usuario salga de la aplicación para buscar imágenes o gráficos externos, ya que todos los elementos visuales deberán ser generados y presentados internamente en la interfaz de la aplicación. Esto garantizará una experiencia de usuario fluida y centralizada.

Interfaz sugerida

Usuario

Social Structure

Buscar

Publicaciones

Solicitudes

Reportes

Perfil

Cerrar sesión

pdro@gmail.com

Buscar

Datos de usuario

Nombres

Apellidos

Correo

Fecha de nacimiento

Social Structure

Buscar

Publicaciones

Solicitudes

Reportes

Perfil

Cerrar sesión

Fecha

Todos

Aplicar

Recorrido limitado

Orden

PreOrden

Cantidad

Aplicar

Crear nueva publicación

Usuario1

Fecha

Texto de la publicación

Comentar

Ver comentarios

Ver árbol de comentarios

Social Structure

BuscarPublicacionesSolicitudesReportesPerfil

Cerrar sesión

Usuarios

Nombres	Apellidos	Correo	Nacimiento	
Pedro	Suarez	pdro@gmail.com	21/02/2002	Enviar solicitud

Solicitudes recibidas

Correo	
pdro@gmail.com	AceptarRechazar

Solicitudes enviadas

Correo	
pdro@gmail.com	Cancelar

Social Structure

BuscarPublicacionesSolicitudesReportesPerfil

Cerrar sesión

Fechas con más publicaciones

Fecha	Cantidad de publicaciones

Publicaciones con más comentarios

Fecha	Usuario	No. Comentarios

Generar reportes

BST

21/06/2024

Generar

Social Structure

BuscarPublicacionesSolicitudesReportesPerfil

Cerrar sesión

Perfil

Nombres

Apellidos

Correo

Contraseña

Fecha de nacimiento

Modificar datos

Eliminar cuenta

Administrador

Social Structure

BuscarCargas MasivasReportes

Cerrar sesión

pdro@gmail.com

Buscar

Orden

PreOrden

Aplicar

Nombres	Apellidos	Correo	Nacimiento		
Pedro	Suarez	pdro@gmail.com	21/02/2002	Modificar	Eliminar

Social Structure

Buscar

Cargas Masivas

Reportes

Cerrar sesión

Usuarios

File

Cargar usuarios

Solicitudes

File

Cargar Solicitudes

Publicaciones

File

Cargar publicaciones

Social Structure

Buscar

Cargas Masivas

Reportes

Cerrar sesión

Arbol de usuarios

Lista de publicaciones

Generar reportes

Observaciones

- Lenguaje de programación a utilizar: **C++**
- El nombre del usuario administrador será **admin@gmail.com** y su contraseña será **EDD2S2024**.
- Sistema Operativo: Libre
- IDE: Libre
- GUI: Libre
- Herramienta para desarrollo de reportes gráficos: **Graphviz**.
- Durante la calificación se harán preguntas para validar que el estudiante realizó el proyecto, de no responder correctamente anulará la nota obtenida en la o las secciones en la que se aplique tal concepto.
- Cada estudiante deberá utilizar un repositorio de github con el nombre **[EDD]Proyecto_carnet**, ir agrupando cada fase por carpetas dentro del mismo repositorio.
- Apartado de entrega en la plataforma UEDI:
 - Fecha y hora de entrega: **21 de septiembre a las 23:59 horas**.
- Es **obligatorio** entregar esta fase para tener derecho a realizar las demás.
- Es **obligatorio** implementar una interfaz gráfica para tener derecho a calificarse.
- Las copias serán penalizadas con una nota de 0 y castigadas según lo indique el reglamento.

Entregables

- Link a repositorio
 - Código fuente
 - Manual de Usuario
 - Manual Técnico