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 1

Social Structure

Implementación de estructuras no lineales

ÍNDICE

Objetivos

Objetivo general

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

Objetivos específicos

- Aplicar los conocimientos adquiridos sobre estructuras de datos lineales.
- Implementar una aplicación en consola 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 actualidad, las redes sociales son cruciales debido a su capacidad para conectar personas globalmente, facilitando la comunicación instantánea y la compartición de información. Son herramientas esenciales para el marketing y la publicidad, permitiendo a las empresas interactuar directamente con los consumidores. Además, juegan un papel vital en la movilización social y política, y en la difusión de movimiento y causas. Las redes sociales también sirven como plataformas educativas y de entretenimiento, influyen en la cultura popular proporcionando datos valiosos para la investigación y el análisis de tendencias. Su impacto se extiende a aspectos personales, profesionales y comunitarios, convirtiéndolas en un aparte integral de la vida moderna.

En resumen, las redes sociales se han vuelto parte fundamental del día a día de personas, organizaciones y demás entidades en la actualidad.

Por esta razón es que se le solicita a usted, como estudiante de Ingeniería en Sistemas de la Universidad de San Carlos de Guatemala, realizar de forma gradual una aplicación de escritorio que emule el funcionamiento de una red social en la cual se implementarán diversas estructuras de datos para poder garantizar su funcionamiento óptimo.

Como se mencionó anteriormente, el desarrollo de esta aplicación será incremental por lo que en su primera versión solamente se ejecutará en consola.

Funcionalidades

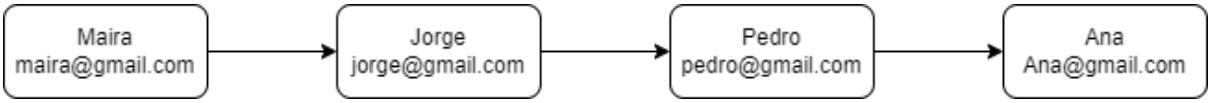
Registro de usuarios e inicio de sesión

Se deberá contar con la funcionalidad de registro de usuarios en la red social, los datos solicitados deberán ser:

- Nombres
- Apellidos
- Fecha de nacimiento
- Correo electrónico
- Contraseña

Al momento de crear una cuenta se deberá verificar que el correo no exista dentro de la aplicación y se procederá a almacenar la información en una **lista simplemente enlazada**.

Adicionalmente, la aplicación deberá poseer una opción para que el usuario pueda iniciar sesión en su cuenta, **los datos que se deberán utilizar para este inicio de sesión deben ser el correo y la contraseña.**

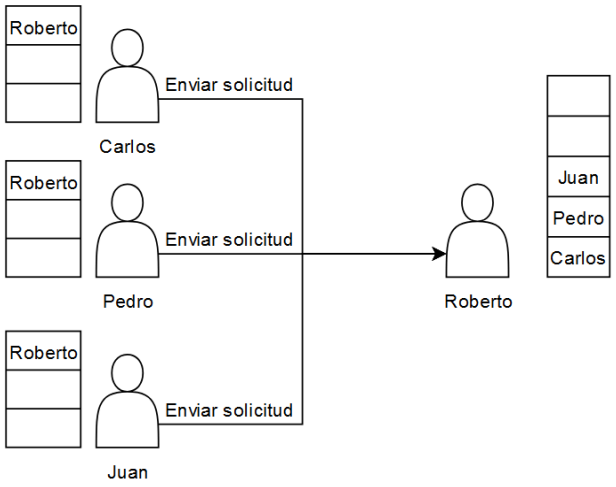


Cada usuario tiene la opción de eliminar su cuenta si así lo desea, tomando en cuenta que estas acciones afectarán a las demás estructuras (lista de publicaciones, matriz de relaciones y solicitudes enviadas y recibidas).

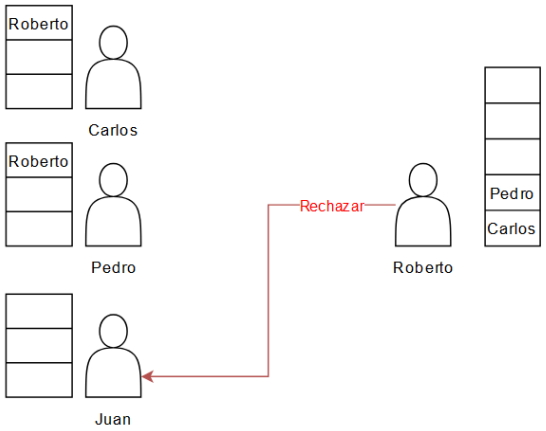
Solicitudes de amistad

- **Envío de solicitudes de amistad**
  La primera acción que se podrá realizar es el envío de solicitudes de amistad a otros usuarios que se encuentren registrados en la red social, para esta mecánica se deberán manejar dos estructuras distintas. Al momento que se envía una solicitud de amistad a un usuario esta deberá ser almacenada en una **pila** que poseerá el usuario receptor, a su vez el usuario emisor poseerá una **lista simple**, en la cual se registrará la solicitud de amistad que envió.

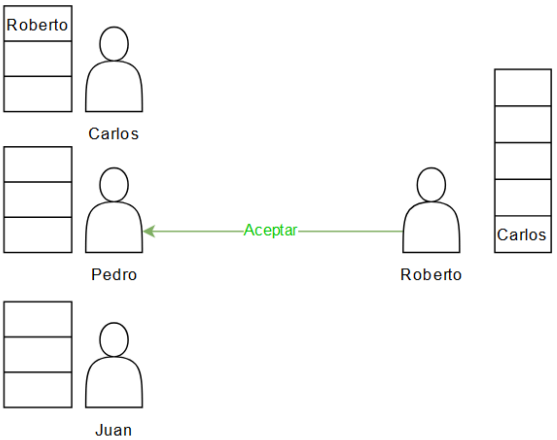
  Al realizar esta acción se deben de tomar en cuenta ciertas consideraciones, la primera es que un usuario no puede enviar una solicitud a otro usuario del cual ya posea una solicitud pendiente de aceptar o rechazar y la siguiente es que un usuario no puede enviar una solicitud de amistad a alguien a quién ya le envió una solicitud anteriormente.



- **Rechazar solicitudes de amistad**
  Los usuarios receptores deben ser capaces de poder rechazar las solicitudes de amistad que se encuentren en su pila de solicitudes recibidas, al momento de que una solicitud de amista sea rechazada, esta deberá de ser eliminada tanto de la pila de solicitudes de amistad recibidas (en el caso del receptor) como de la cola de solicitudes de amistad enviadas (en el caso del emisor).

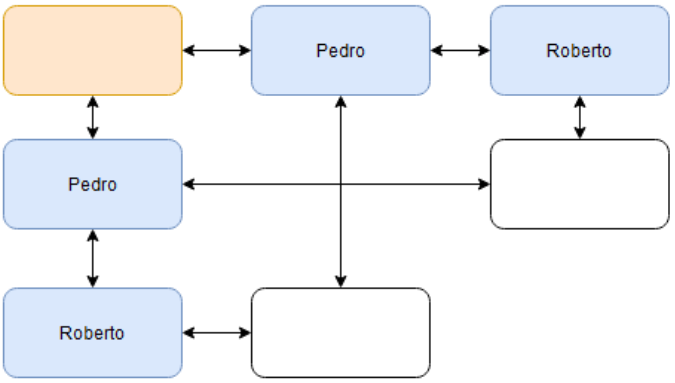


- **Aceptar solicitudes**
  Al igual que puede rechazar solicitudes de amistad recibidas, los usuarios también pueden aceptar solicitudes de amistad recibidas, en el caso de aceptar una solicitud de amistad el comportamiento inicial se realiza al igual que cuando se rechaza una solicitud. Se elimina la solicitud de la cola de solicitudes enviadas (en el caso del emisor) y de la pila de solicitudes recibidas (en el caso del receptor). De manera adicional, esta nueva amistad se deberá de reflejar en una **Matriz de relaciones de amistad**, de la cual se hablará a continuación.

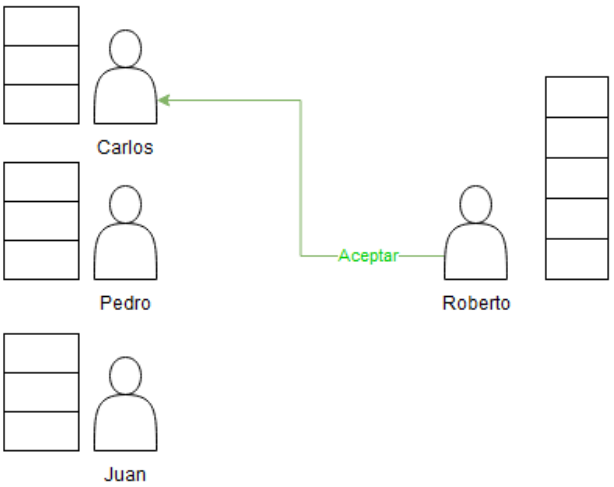


Matriz de relaciones de amistad

Las amistades dentro de la red social se deberán representar haciendo uso de una **Matriz dispersa**. El procedimiento para insertar algún nodo dentro de la matriz dispersa se deberá llevar a cabo después que un usuario acepte alguna solicitud de amistad, después de haber realizado esta acción, se deberán insertar tanto en las filas como en las columnas de la matriz dispersa, los nombres involucrados en la nueva amistad (solamente si estos nombres no han sido insertados con anterioridad).

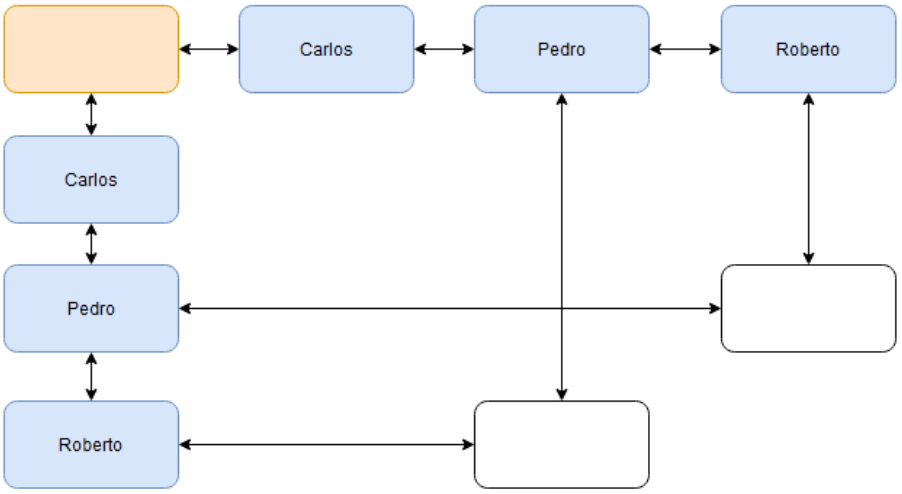


Para ejemplificar lo descrito anteriormente, podemos suponer que Roberto acepta la solicitud de amistad de Carlos (siguiendo el ejemplo de las imágenes).
Lo primero que se realizará al momento que Roberto acepte la solicitud de amistad de Carlos, es eliminar dicha solicitud de amistad de la pila de solicitudes recibidas de Roberto y de la cola de solicitudes enviadas de Carlos para poder indicar que dicha solicitud de amistad ya no se encuentra pendiente.

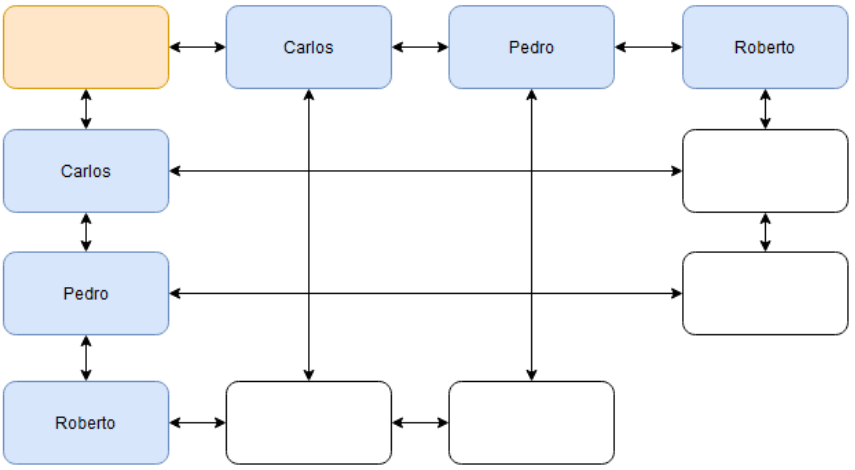


Posteriormente deberemos representar esta nueva amistad en la matriz de relaciones de amistad, esto lo podremos hacer insertando (tanto en filas como en columnas) el nombre de

las personas involucradas, como se puede observar, Roberto ya se encuentra insertado por lo tanto no es necesario insertarlo nuevamente, sin embargo en el caso de Carlos no aparece en la matriz y será necesario agregarlo.



Finalmente se insertan los nodos necesarios para representar la relación de amistad entre las personas involucradas.



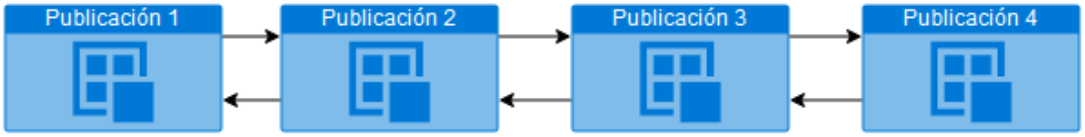
Publicaciones

Los usuarios registrados en el sistema pueden realizar y ver publicaciones suyas o de sus **amigos**. Para crear una publicación solo se necesita el contenido que por ser una aplicación en consola solo se publicará texto. La información que se debe almacenar en la estructura es la siguiente:

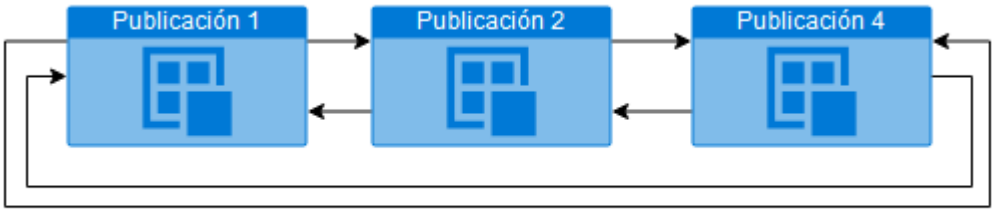
- Correo del usuario que realiza la publicación

- Contenido(texto)
- Fecha
- Hora

Al momento de ser creada, la publicación deberá ser almacenada en una **lista doblemente enlazada** la cual contendrá todas las publicaciones de la aplicación.



Sin embargo, los usuarios no podrán ver completamente la información de esta estructura. Al momento que algún usuario quiera ver su feed de publicaciones, se deberá ir a esta lista, obtener todas las publicaciones que el usuario en concreto tenga permitido ver (es decir, solamente publicaciones suyas y de sus amigos) y posteriormente insertarlas en una lista circular doblemente enlazada, la cual se utilizará para que el usuario pueda explorar en ambas direcciones las publicaciones de su feed.



Por ser una aplicación en consola para visualizar las publicaciones dentro de la aplicación, deberán mostrar la información de la publicación y tener las opciones de anterior y siguiente.

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. En cada una se deberá de solicitar el path donde se encuentre el archivo.

A continuación se muestra un ejemplo de la estructura de los json.

Usuarios

usuarios.json U

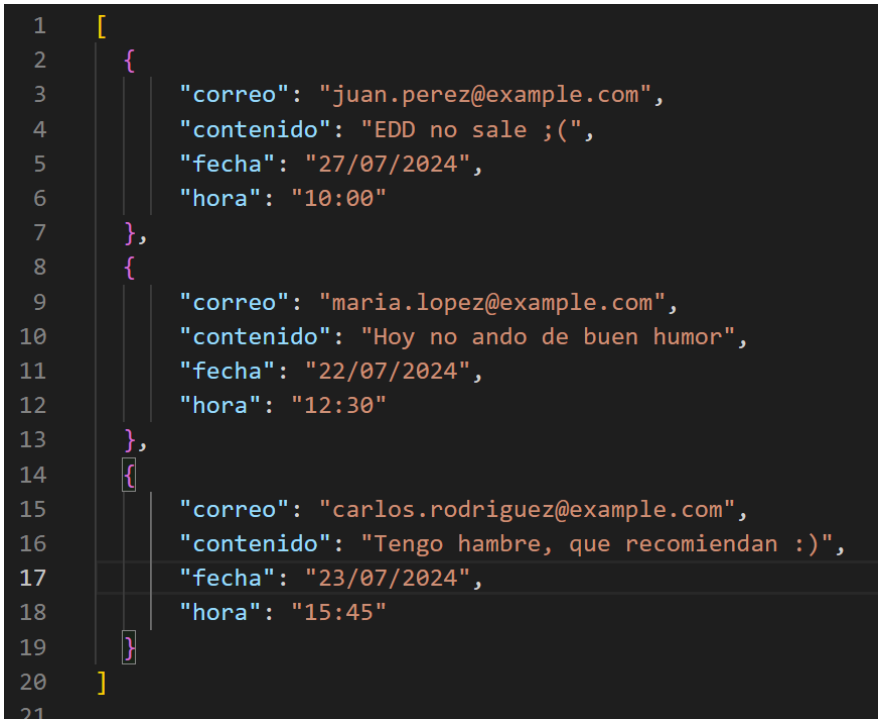
usuarios.json > { } 0 > abc fecha_de_nacimiento

```json
[
    {
        "nombres": "Juan",
        "apellidos": "Perez Gomez",
        "fecha_de_nacimiento": "1990/05/14",
        "correo": "juan.perez@example.com",
        "contraseña": "password123"
    },
    {
        "nombres": "Maria",
        "apellidos": "Lopez Martinez",
        "fecha_de_nacimiento": "1985/08/22",
        "correo": "maria.lopez@example.com",
        "contraseña": "securepass456"
    },
    {
        "nombres": "Carlos",
        "apellidos": "Rodriguez Fernandez",
        "fecha_de_nacimiento": "1992/11/30",
        "correo": "carlos.rodriguez@example.com",
        "contraseña": "mypassword789"
    }
]
```

Solicitudes

```json
[
    {
        "emisor": "maria.lopez@example.com",
        "receptor": "carlos.rodriguez@example.com",
        "estado": "ACEPTADA"
    },
    {
        "emisor": "juan.perez@example.com",
        "receptor": "maria.lopez@example.com",
        "estado": "PENDIENTE"
    },
    {
        "emisor": "carlos.rodriguez@example.com",
        "receptor": "juan.perez@example.com",
        "estado": "PENDIENTE"
    },
    {
        "emisor": "ana.martinez@example.com",
        "receptor": "maria.lopez@example.com",
        "estado": "ACEPTADA"
    }
]
```

Publicaciones



```json
[
    {
        "correo": "juan.perez@example.com",
        "contenido": "EDD no sale ;(",
        "fecha": "27/07/2024",
        "hora": "10:00"
    },
    {
        "correo": "maria.lopez@example.com",
        "contenido": "Hoy no ando de buen humor",
        "fecha": "22/07/2024",
        "hora": "12:30"
    },
    {
        "correo": "carlos.rodriguez@example.com",
        "contenido": "Tengo hambre, que recomiendan :)",
        "fecha": "23/07/2024",
        "hora": "15:45"
    }
]
```

Reportes

Administrador

- **Usuarios:** Gráfico de la lista de usuarios
- **Relaciones de Amistad**: Gráfico de la matriz dispersa
- **Publicaciones**: Gráfico de la lista doblemente enlazada
- **Top:**
    - 5 usuarios con más publicaciones.
    - 5 usuarios con menos amigos.

Usuario

- **Solicitudes Enviadas y Recibidas:** Gráfico de la lista de solicitudes enviadas y de la pila de solicitudes recibidas
- **Relaciones de Amistad**: Gráfico de la matriz dispersa
- **Publicaciones**: Gráfico de la lista circular de publicaciones del usuario y de amigos
- **Mis Amigos**: Listado de amigos

Interfaz Sugerida

1. Iniciar sesión
2. Registrarse
3. Información
4. Salir

Modulo Administrador

1. Carga de usuarios
2. Carga de relaciones
3. Carga de publicaciones
4. Gestionar usuarios
    a. Eliminar usuarios
5. Reportes

Modulo Usuario

1. Perfil
    a. Ver perfil
    b. Eliminar cuenta
1. Solicitudes
    a. Ver solicitudes
        i. Aceptar/Rechazar
    b. Enviar
2. Publicaciones
    a. Ver todas
    b. Crear
    c. Eliminar
3. Reportes
4. Salir

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.
- 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 agosto a las 23:59 horas.**
- Es **obligatorio** entregar esta fase para tener derecho a realizar las demás.
- 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