### **Título del Proyecto: Social Structure - Implementación de estructuras lineales**

### **1. Resumen Ejecutivo**

El proyecto “Social Structure” básicamente trata del desarrollo de una aplicación de escritorio, en el lenguaje de programación C + +, que busca emular el comportamiento de una red social en la cual se podrán gestionar usuarios, relaciones de amistad entre ellos, solicitudes de amistad, publicaciones, etc. La gestión de todas la información necesaria para las funcionalidades anteriormente mencionadas se deberá almacenar en estructuras de datos.

Este proyecto implica la gestión de usuarios en la aplicación mediante una lista doblemente enlazada, la gestión de solicitudes de amistad enviadas y recibidas a través de pilas y listas circulares, el almacenamiento de publicaciones a través de una lista doblemente enlazada y que, posteriormente, los usuarios puedan ver sus publicaciones a través del uso de listas circulares doblemente enlazadas y la representación de relaciones de amistad a través de una matriz dispersa. Esta implementación busca que, al finalizar, el estudiante sea capaz de comprender e implementar las estructuras de datos lineales en proyectos de software.

### **3. Objetivos del Aprendizaje**

#### **3.1 Objetivo General**

El estudiante será capaz de desarrollar una aplicación en escritorio, en el lenguaje de programación C + +, que permita emular el comportamiento de una red social al gestionar usuarios, relaciones de amistad entre ellos y publicaciones haciendo uso de estructuras de datos lineales como listas enlazadas y matrices dispersas.

#### **3.2 Objetivos Específicos**

1. Desarrollar un sistema funcional que utilice distintos tipos de estructuras de datos lineales para la gestión de datos y el correcto manejo de los mismos.
2. Representar de forma gráfica el estado en memoria de cada una de las estructuras de datos involucradas en la solución de software.
3. Conocer de forma teórica y práctica las estructuras de datos lineales, a manera de emplear este conocimiento en el desarrollo de soluciones de software y discernir qué estructuras de datos son más adecuadas para el óptimo funcionamiento de estas soluciones.

### **4. Enunciado del Proyecto**

El proyecto consiste en el diseño y desarrollo de una aplicación de escritorio en C++ que permita gestionar relaciones de amistad entre usuarios y proporcionar recomendaciones basadas en esas relaciones. El problema a resolver es la falta de una herramienta eficiente que facilite la administración de conexiones sociales, permitiendo a los usuarios agregar, eliminar y visualizar amigos de manera sencilla, así como recibir sugerencias para expandir su red de contactos.

**4.1 Descripción del problema a resolver**

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.

**4.2 Alcance del proyecto**

* **Alcance obligatorio**: A pesar de que el resultado final del proyecto deberá ser una aplicación de escritorio, este enunciado solamente es la primera fase del proyecto en su totalidad por lo cual esta versión de la aplicación debe ser desarrollada solamente en consola. En esta aplicación de consola los estudiantes deberán de implementar de forma obligatoria:
  + Un sistema de registro de usuarios e inicio de sesión para poder acceder a las funcionalidades de la aplicación ya sea bien como usuario normal o como administrador.
  + Los usuarios dentro de la aplicación deberán ser capaces de editar su información de usuario o de eliminar sus perfiles.
  + Todos los usuarios que se encuentren registrados dentro de la aplicación deberán estar almacenados dentro de una lista doblemente enlazada.
  + Cada uno de los usuarios dentro de la aplicación deberá poseer una pila en la cual se almacenarán las solicitudes de amistad recibidas y una lista circular que servirá para almacenar todas las solicitudes de amistad enviadas a otros usuarios.
  + Se deberá implementar una matriz dispersa para poder representar las relaciones de amistad entre todos los usuarios que se encuentren registrados en la aplicación.
  + Las publicaciones realizadas por todos los usuarios de la aplicación deberán ser almacenadas en una lista doblemente enlazada.
  + Los usuarios solamente podrán ver las publicaciones suyas y de sus amigos, por lo cual se extraerá de la lista global de publicaciones aquellas publicaciones que algún usuario en cuestión pueda ver y se insertarán en una lista circular doblemente enlazada, la cual será la que se le presentará al usuario al momento de ingresar a su feed.
  + El usuario administrador deberá disponer de la capacidad de cargar de forma masiva usuarios, solicitudes de amistad y publicaciones a través del ingreso de archivos de tipo JSON.
  + El programa debe ser capaz de graficar todas las estructuras anteriormente mencionadas en algún instante puntual de tiempo.
  + El usuario administrador podrá realizar reportes como: usuarios con más publicaciones, con menos amigos y las amistades más recientes.
  + El usuario normal podrá saber cuales son sus amigos más antiguos.
  + Los usuarios podrán realizar publicaciones, así como interactuar con sus solicitudes de amistad (rechazar o aceptar solicitudes de amistad, enviar solicitudes, cancelar solicitudes, etc.).
  + No se podrán usar ningún tipo de estructuras de datos provenientes de librerías de C + +.
* **Alcance opcional**: Implementar funcionalidades de la solución haciendo uso de la interfaz gráfica. A pesar de que se hizo una lista de funcionalidades obligatorias anteriormente, estas pueden estar estructuradas de cualquier manera en la aplicación de consola por lo que el implementar estilos y demás aspectos de personalización en la visualización de la aplicación en consola de igual manera es opcional.

**4.3 Requerimientos técnicos**Los estudiantes deberán utilizar C + + como lenguaje de programación principal para el desarrollo del proyecto. En esta fase se estará realizando la aplicación en consola, sin embargo, se espera que integren Qt como plataforma para crear una interfaz gráfica de usuario (GUI) en las siguientes fases. Para graficar las estructuras de datos se espera que se use la herramienta Graphviz.

Adicionalmente, se recomienda el uso de herramientas de control de versiones como Git para mantener un historial del desarrollo del proyecto y facilitar la documentación técnica.

**4.4 Entregables**

| **Tipo** | **Descripción** |
| --- | --- |
| **Prototipo** | Desarrollo de una versión funcional inicial del proyecto o sistema, que permite visualizar su funcionamiento básico antes de la implementación final. |
| **Informe Técnico** | Un documento detallado que describe el proceso de desarrollo, análisis de resultados, pruebas realizadas y conclusiones obtenidas. |
| **Documentación Técnica** | Incluye manuales de usuario, guías de instalación, diagramas y descripciones de la arquitectura o flujo del sistema desarrollado. |
| **Código Fuente** | El código completo del proyecto, con una estructura clara y comentada, que refleja las buenas prácticas de programación y uso de control de versiones. |
| **Presentación Final** | Una presentación en la que los estudiantes exponen su proyecto ante la clase y el profesor, incluyendo los resultados, retos y soluciones. |
| **Plan de Pruebas** | Un informe que describe las pruebas realizadas, los casos de prueba, los resultados obtenidos y las mejoras implementadas en el sistema. |
| **Mockups/Diseños** | Bocetos o diseños visuales que representan la interfaz de usuario o las principales pantallas del sistema antes del desarrollo técnico. |
| **Video Demostrativo** | Un video que muestra el funcionamiento del sistema o prototipo, acompañado de una explicación de las características clave. |
| **Lista de Tareas** | Un resumen de las tareas realizadas durante el proyecto, con detalles sobre el progreso, los responsables y el cumplimiento de los plazos. |
| **Evaluación de Usuarios** | Un informe sobre las pruebas realizadas por usuarios reales o simulados, con retroalimentación y mejoras propuestas basadas en sus comentarios. |
| **Manual de Usuario** | Documento que explica cómo usar el sistema desarrollado, incluyendo capturas de pantalla, pasos detallados y resolución de problemas comunes. |
| **Diagrama de Flujo** | Representación gráfica del flujo de trabajo o funcionamiento del sistema, que permite comprender el recorrido de los datos o acciones dentro del proyecto. |
| **Informe de Impacto** | Documento que describe el impacto potencial del proyecto en su contexto de aplicación, como mejoras de eficiencia, reducción de costos, etc. |
| **Plan de Mantenimiento** | Propuesta de cómo se mantendrá y actualizará el sistema a largo plazo, incluyendo posibles mejoras o futuros desarrollos. |

* **Documentación técnica:** Los estudiantes deberán proporcionar un documento en el cual se deberá detallar el proceso de instalación de las herramientas empleadas, así como una descripción de cada una de las estructuras de datos empleadas y su papel en el funcionamiento del programa. Se deberá incluir capturas de pantalla del proceso de instalación de las herramientas y del código fuente de la aplicación.
* **Manual de usuario:** Se deberá entregar un documento donde se debe explicar, a los usuarios finales, como utilizar la aplicación desarrollada. Se deberá incluir capturas de pantalla.
* **Código fuente:** Se entregará el código fuente de la aplicación, el proyecto deberá poseer una estructura clara, en la cual se pueda apreciar las distintas partes del proyecto y el código deberá estar debidamente comentado y estructurado.

### **5. Metodología**

1. **Análisis de requisitos**: Revisión de los requisitos y planificación de la implementación de las estructuras de datos.
2. **Diseño de la solución**: Definición del diseño de las estructuras, detallando el tipo de operaciones, el tipo de dato a almacenar y su justificación.
3. **Desarrollo**: Codificación y pruebas unitarias de cada estructura de datos.
4. **Integración:** Implementar cada una de las estructuras de datos desarrolladas en la aplicación.
5. **Pruebas y ajustes**: Probar y validar que cada una de las estructuras de datos implementadas funcione de forma correcta y se comunique satisfactoriamente con las demás estructuras de datos en la aplicación.
6. **Documentación:** Redacción del informe técnico y la documentación de cada estructura.

### **6. Desarrollo de Habilidades Blandas**

### **6.2 Proyectos Individuales**

**6.2.1 Autogestión del Tiempo**Los estudiantes deben crear un cronograma personal para cumplir con los plazos establecidos. Esto les ayuda a mejorar su disciplina y capacidad de priorización, habilidades esenciales en cualquier entorno profesional.

**6.2.2 Responsabilidad y Compromiso**En un proyecto individual, el estudiante asume la totalidad de las responsabilidades, desde la investigación hasta la entrega final. Esto fomenta el sentido de compromiso y permite una mayor personalización en la solución del problema planteado.

**6.2.3 Resolución de Problemas**Trabajar de manera independiente impulsa a los estudiantes a buscar soluciones de forma creativa y a enfrentarse a los desafíos sin depender del apoyo constante de otros. Esto fortalece su capacidad para resolver problemas de manera autónoma.

**6.2.4 Reflexión Personal**Al concluir el proyecto, el estudiante realiza una autoevaluación, reflexionando sobre sus decisiones, lo aprendido y las áreas en las que podría mejorar. Esta práctica promueve el desarrollo continuo y el autoaprendizaje.

### **7. Cronograma**

El cronograma describe las etapas clave del proyecto, los plazos estimados para cada una, y el proceso de asignación, elaboración y calificación de las tareas. Los estudiantes deberán seguir este plan para asegurar que el proyecto avance de manera organizada y cumpla con los plazos establecidos. Cada fase incluye la asignación de tareas, el tiempo estimado para su elaboración, y el momento de su calificación.

| Tipo | Fecha Inicio | Fecha Fin |
| --- | --- | --- |
| Asignación de Proyecto | Semana 2 | Semana 2 |
| Elaboración | Semana 2 | Semana 5 |
| Calificación | Semana 6 | Semana 6 |

Este **cronograma** es un ejemplo general y puede ser ajustado según las necesidades y el contexto del proyecto.

### **8. Evaluación**

La evaluación del proyecto se realizará en función de varios criterios clave, teniendo en cuenta tanto los aspectos técnicos como las habilidades blandas demostradas a lo largo del desarrollo. A continuación, se presenta un **ejemplo** de cómo podría distribuirse la puntuación, aunque esto puede cambiar según sea necesario y las particularidades del proyecto.

### **Resumen de Puntuaciones**

| **Área** | **Puntos Totales** | **Puntos Obtenidos** |
| --- | --- | --- |
| Aplicación | 7 |  |
| Usuarios | 15 |  |
| Solicitudes de amistad | 18 |  |
| Publicaciones | 20 |  |
| Reportes | 30 |  |
| Documentación | 10 |  |
| **Total** | **100** |  |

### **Detalle de la Calificaciòn**

| **Criterio** | **Descripción** | **Puntos Máximos** | **Puntuación Obtenida** |
| --- | --- | --- | --- |
| **1. Aplicación** |  | **7** |  |
| 1.1 Cargas masivas | La aplicación cumple con tener la opción de carga masiva de usuarios, publicaciones y relaciones de amistad. |  |  |
| **2. Usuarios** |  | **15** |  |
| 2.1 Gestión de usuarios | Se implementó la funcionalidad de inicio de sesión y registro de usuarios. |  |  |
| 2.2 Opciones de edición | Los usuarios son capaces de editar sus datos o de eliminar su perfil y toda la información relacionada al mismo de la aplicación. |  |  |
| 2.3 Implementación | Los usuarios se almacenan en una lista doblemente enlazada. |  |  |
| **3. Solicitudes de amistad** |  | **18** |  |
| 3.1 Implementación | Se usa una pila para gestionar las solicitudes de amistad recibidas, una lista circular para las solicitudes de amistad enviadas y una matriz dispersa para gestionar las amistades entre usuarios en la aplicación. |  |  |
| 3.2 Gestión de solicitudes | Los usuarios de la aplicación son capaces de aceptar y rechazar solicitudes recibidas, así como de cancelar el envío de alguna solicitud de amistad a otro usuario. |  |  |
| **4. Publicaciones** |  | **20** |  |
| 4.1 Implementación | Se hace uso de una lista doble para gestionar las publicaciones de la aplicación y de una lista circular doblemente enlazada para gestionar las publicaciones de cada usuario. |  |  |
| 4.2 Gestión de publicaciones | El usuario es capaz de ver sus publicaciones y las de sus amigos, así como de crear nuevas publicaciones y de navegar entre ellas (hacia adelante o atrás). |  |  |
| **5. Reportes** |  | **30** |  |
| 5.1 Reportes de administrador | El usuario administrador es capaz de visualizar, de forma gráfica, la lista doblemente enlazada de usuarios, la matriz dispersa de las relaciones de amistad y la lista doblemente enlazada de publicaciones. |  |  |
| 5.2 Reportes de usuario | El usuario es capaz de ver, de forma gráfica, su lista circular de solicitudes enviadas y su pila de solicitudes recibidas, además de una matriz con sus relaciones de amistad con otros usuarios, su lista circular doblemente enlazada de publicaciones y su listado de amigos. |  |  |
| 5.3 Ordenamientos | El usuario administrador puede ver el listado de los 5 usuarios con más publicaciones en la aplicación, los 5 usuarios con menos amigos y las 3 relaciones de amistad más recientes.  Los usuarios son capaces de ver sus 5 amigos más antiguos. |  |  |
| 5.4 Precisión y claridad | Los reportes generados se encuentran ordenados y debidamente estructurados de manera que resulte clara la información que presentan. |  |  |
| **6. Documentación** |  | **10** |  |
| 6.1 Manual de Usuario | Claridad y detalle en el manual de usuario, con guías de uso e imágenes que ayuden a su comprensión. |  |  |
| 6.2 Manual Técnico | Calidad y detalle en la documentación técnica, incluyendo guías de instalación de las herramientas y la explicación de la estructura de la solución. |  |  |
| **Total de Puntos** |  | **100** |  |

### **Comentarios Generales:**

Como se mencionó anteriormente, se encuentra prohibido el uso de librerías del lenguaje de programación a usar que provean estructuras de datos ya que todas las estructuras deben ser realizadas por el estudiante. La entrega de esta fase es obligatoria para tener derecho a entregar la siguiente.