

TU AMIGO UN

Juan Camilo López Bustos, Alejandro Mendivelso Torres, Luis Ángel Rodríguez Valdelamar

No. de equipo de trabajo: {2do}

I. INTRODUCCIÓN

En esta entrega veremos el primer esbozo del esquema del programa “Tu amigo UN”, que espera proveer una solución al problema de socialización y aislamiento presentado por una gran parte de los estudiantes de la universidad nacional.

También veremos cómo se va a llevar a cabo la construcción del programa sus parámetros y las diferentes opciones que presentara para esta primera pre versión, los resultados esperados y una descripción muy precisa de las estructuras de datos implementadas y como se usan dentro del programa.

Además de un análisis muy completo de los casos de prueba y con las diferentes estructuras de datos y cómo se comportan en los diferentes requerimientos como añadir, borrar y buscar un dato.

II. DESCRIPCIÓN DEL PROBLEMA A RESOLVER

I. Es bien sabido que el aislamiento social a causa del covid19 ha dejado muchos problemas por resolver, uno de estos es el foco de este proyecto, y es el causado por el mismo aislamiento y la falta de interacción por parte de los jóvenes pertenecientes a la universidad nacional de Colombia, ya que se ha demostrado que el aislamiento prolongado puede causar problemas de interacción social y física[1], problema que ha tratado de ser mitigado con diferentes actividades culturales y espacios de reunión que ofrece la universidad y la sección de bienestar, pero que a pesar de esto muchos estudiantes se han quejado constantemente del hecho de no poder socializar o simplemente no saber cómo hacerlo ya que carecen tanto de habilidades como de herramientas que les faciliten dicha interacción.

III. USUARIOS DEL PRODUCTO DE SOFTWARE

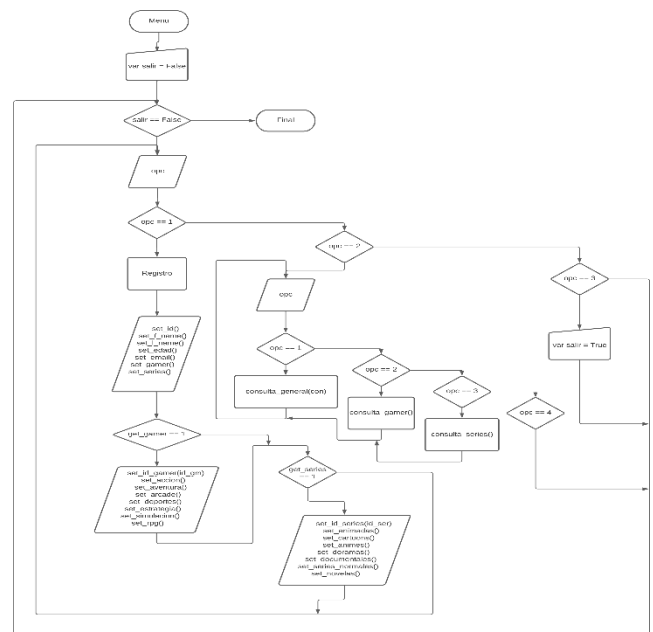
Este producto está destinado a cualquier estudiante de la universidad nacional de Colombia o graduado de la misma, que quiera conocer personas nuevas con sus mismos o diferentes gustos en cuanto a actividades de ocio, no es necesario tener algún conocimiento avanzado en computación ya que el programa está pensado como una plataforma de búsqueda y registro para crear nuevas amistades y fomentar espacios de ocio y esparcimiento.

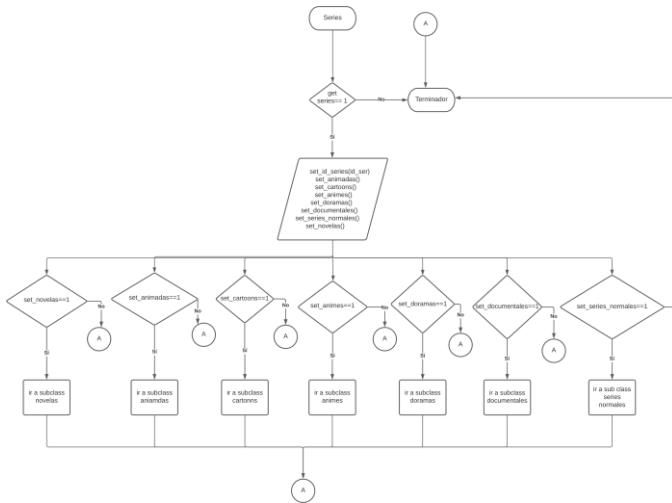
IV. REQUERIMIENTOS FUNCIONALES DEL SOFTWARE

En esta sección veremos los requerimientos mínimos de funcionamiento, y cómo se comporta cada una de las funcionalidades del programa.

- Registro: En esta parte del programa se realiza el registro de los datos del usuario y la selección de sus preferencias, para posteriormente ser almacenadas en una base de datos. El usuario deberá dar clic en la opción de registro, posteriormente diligenciar los datos para ser capturados y guardados temporalmente y dependiendo de su selección de gustos serán guardadas sus preferencias para finalmente ser enviadas a la base de datos.
- Búsqueda: En esta parte del programa el usuario podrá realizar diferentes tipos de búsqueda dependiendo de las opciones que el elija, desde una búsqueda avanzada, hasta una búsqueda específica por cada una de las categorías disponibles. El usuario puede seleccionar diferentes opciones de búsqueda que le enviarán y mostrarán múltiples datos en su pantalla dependiendo de las opciones que este elija, además esta información se guarda de manera temporal para en versiones futuras del programa ser ordenada de manera alfabética, tales como categoría de gamer o series donde hay subcategorías de búsqueda más avanzadas.

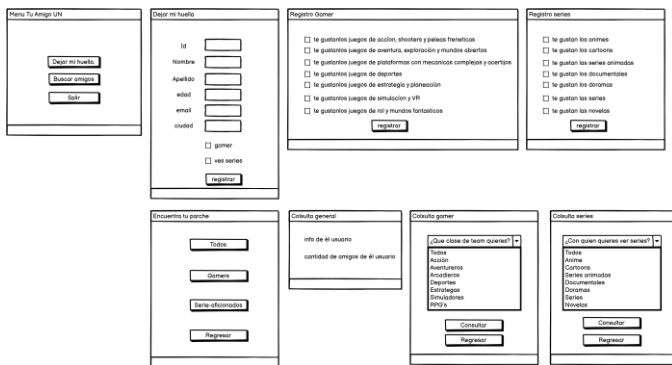
A continuación, se presentan dos diagramas de flujo que muestran la funcionalidad de registro y búsqueda de manera general, se espera para versiones futuras poder implementar una actualización de datos de usuario, y el ordenamiento de datos de manera alfabética.





V. DESCRIPCIÓN DE LA INTERFAZ DE USUARIO PRELIMINAR

Se espera tener una interfaz sencilla e intuitiva, por el momento para esta primera versión no se tiene una interfaz gráfica desarrollada, pero si un concepto de como se espera que se llegue a ver en próximas versiones.



VI. ENTORNOS DE DESARROLLO Y DE OPERACIÓN

El programa es desarrollado en Python, y se espera que se pueda manejar a través de una plataforma web, para todos los sistemas operativos comerciales actuales (Windows, macOS, Linux, Android) y para equipos con características de hardware de bajo desempeño (desde 2gb de RAM, procesador de 2 núcleos 4 hilos), ya que se espera que también se pueda acceder desde dispositivos móviles.

VII. PROTOTIPO DE SOFTWARE INICIAL

A continuación, se deja el enlace de la carpeta de GitHub que contiene el programa y todos los archivos necesarios para que funcione.

https://github.com/Zecro-Darkness/Proyecto_ED

VIII. DISEÑO, IMPLEMENTACIÓN Y APLICACIÓN DE LAS ESTRUCTURAS DE DATOS

Para diseñar este programa se hizo uso de una lista simple enlazada, ya que en esta se podían guardar los datos de manera lineal y crear una lista de listas para almacenar la información de consulta.

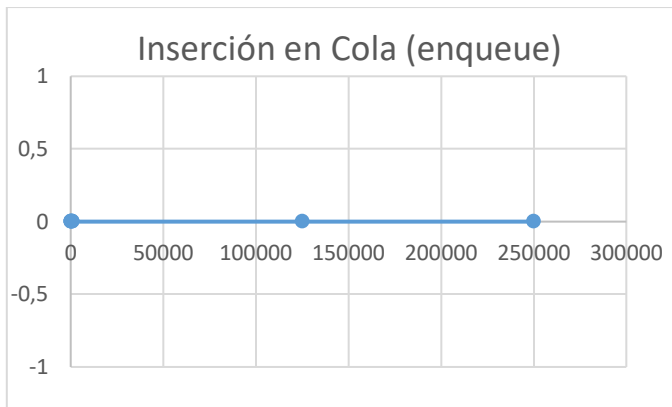
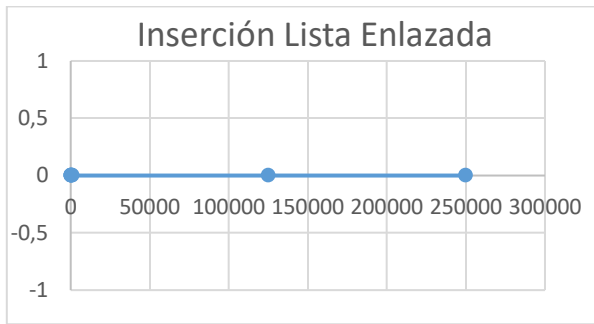
Para poder implementar esta lista enlazada se definió una clase nodo con algunos atributos simples, que son valor (value) y valor siguiente (Next) que ayudan a enlazar los datos almacenados, también se definieron métodos para obtener los datos como caracteres y como enteros para compararlos, luego se definió como tal la clase lista enlazada (LinkedList) que tiene también atributos simples que son el primer ítem (First) y el tamaño (Size), también se definieron métodos para la inserción de datos de manera individual (append) donde el dato se inserta al final de la lista, también uno de eliminación (remove) que recorre la lista hasta encontrar el dato a remover y uno de limpiar (limpiar) que deja vacía la lista, y un método de tamaño (len) que retorna el tamaño de la lista y un método de caracteres (str) que convierte la información guardada en los nodos en caracteres.

Debido a que la lista enlazada simple logró satisfacer la totalidad de las funciones del programa de manera alterna se empleó una cola (Queue) para comparar los tiempos de ejecución y funcionalidad, para esta clase cola se implementó un nodo (Nodo_q) en una estructura de lista doblemente enlazada, por lo que el nodo se compone de las propiedades de valor (data) el nodo siguiente (next) y el anterior (prev). Para la clase cola (Queue) se definió el atributo de cabeza (head) y cola (last) que hacen referencia al primer elemento en la cola y el último respectivamente, luego están los métodos de encolar (enqueue) donde se inserta en la información en un nodo que irá al final de la cola y será referenciado por el anterior y el siguiente, luego está el método de desencolar (dequeue) que saca el nodo de la cabeza y lo reemplaza con el anterior guardando su respectiva referencia, después está el método primero (first) que retorna el primer elemento de la cola (head), y el método último (last) que retorna el último elemento de la cola o encolado, seguido de este está el método tamaño (size) que devuelve la cantidad de elementos encolados y el método vacío (is_empty) para comprobar si hay elementos en la cola, y el método de imprimir que recorre la cola elemento a elemento e imprime su valor.

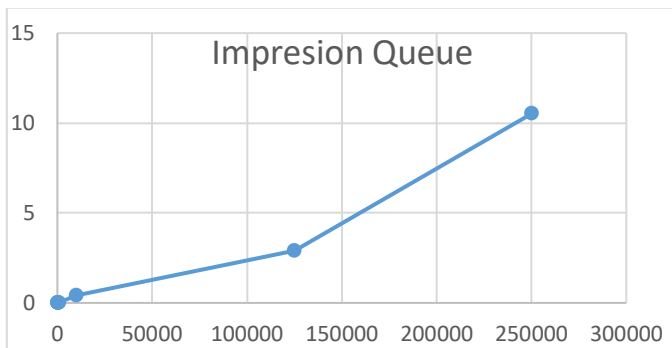
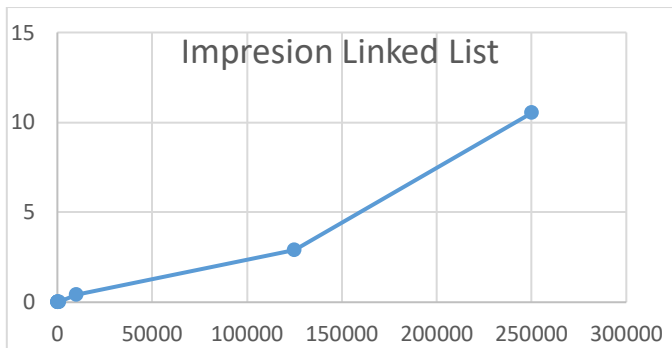
IX. PRUEBAS DEL PROTOTIPO Y ANÁLISIS COMPARATIVO

De acuerdo a las pruebas realizadas en la función de búsqueda que es la que toma más tiempo, ya que el programa debe tomar los datos de la BD y ponerlos en una lista de listas, para luego imprimirlos, para ello se hicieron pruebas tanto en la lista enlazada simple como en la cola.

Para la primera prueba de inserción de n datos se obtuvo un comportamiento de algorítmico de $O(1)$ para ambas estructuras, con muestras de datos máxima de 250.000, los tiempos de inserción correspondientes a la lista enlazada y la cola se muestran a continuación



Para la segunda prueba de la impresión de n datos se obtuvo tanto para la cola como para la lista enlazada un comportamiento de $O(N)$ ya que el tiempo de impresión depende de la cantidad de datos a imprimir, pues recorre uno a uno los datos guardados dentro de la estructura, se usó de igual manera una cantidad máxima de 250.000 datos obteniendo las siguientes gráficas en diferentes pruebas:



X. INFORMACIÓN DE ACCESO AL VIDEO DEMOSTRATIVO DEL PROTOTIPO DE SOFTWARE

El siguiente link muestra un video sobre el software funcionando:

<https://drive.google.com/file/d/1PTK9dNGOVd3xNY4jq-mROaSZudqQlpF4/view?usp=sharing>

XI. ROLES Y ACTIVIDADES

Se deben describir de manera concisa los roles asignados y las respectivas actividades realizadas por cada uno de los integrantes del equipo durante el desarrollo de esta entrega. Para esto, tenga en cuenta los siguientes roles positivos, que puede aplicar durante el desarrollo del proyecto; los mismos deberán ser 'rotados' entre todos los integrantes del equipo en las diferentes entregas.

Rol	Actividades fundamentales
Líder(esa)	Consultar a los otros miembros del equipo, atento que la información sea constante para todos. Aportar con la organización y plan de trabajo.
Coordinador(a)	Mantener el contacto entre todos, Programar y agendar reuniones; ser facilitador para el acceso a los recursos.
Experto(a)	Líder técnico que propende por coordinar funciones y actividades operativas.
Investigador(a)	Consultar otras fuentes. Propender por resolver inquietudes comunes para todo el equipo.
Observador(a)	Siempre está atento en el desarrollo del proyecto y aporta en el momento apropiado cuando se requiera apoyo adicional por parte del equipo.
Animador(a)	Energía positiva, motivador en el grupo.
Secretario(a)	Se convierte en un facilitador de la comunicación en el grupo. Documenta (actas) de los acuerdos/compromisos realizados en las reuniones del equipo.
Técnico(a)	Aporta técnicamente en el desarrollo del proyecto.

Puede utilizar la siguiente tabla para definir los integrantes del grupo, los roles asignados y el listado de actividades durante el desarrollo de la entrega:

INTEGRANTE	ROL(ES)	ACTIVIDADES REALIZADAS (Listado)
Juan Camilo López Bustos	Líder	Desarrollador de la idea del proyecto
	Experto	Crear parte del código y buscar soluciones
	Investigador	Ayuda a resolver inquietudes sobre las

XII. DIFICULTADES Y LECCIONES APRENDIDAS

La dificultad más grande del proyecto fue englobar un interés en común entre las personas a la que está dirigida el proyecto para que puedan relacionarse entre sí de manera más fácil y con más confianza ya que es un tema de interés entre las personas, y también como organizar la información de la persona, los intereses, y sus gustos; y una forma fácil de relacionar los intereses de diversos usuarios para poder integrar a los usuarios. También que el trabajo en grupo es difícil xd.

XIII. REFERENCIAS BIBLIOGRÁFICAS

- [1] Yunier Broche-Pérez, “Consecuencias psicológicas de la cuarentena y el aislamiento social durante la pandemia de COVID-19”, *Scielo.sld.cu*, 2020 [En línea]. Disponible en: http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S0864-34662020000500007
- [2] Weiss, M.A.: *Data Structures and Algorithm Analysis in C++*, 4th Edition, Pearson/Addison Wesley, 2014.
- [3] Hernández, Z.J. y otros: *Fundamentos de Estructuras de Datos. Soluciones en Ada, Java y C++*, Thomson, 2005.
- [4] Shaffer, Clifford A.: *Data Structures and Algorithm Analysis in C++*, Third Edition, Dover Publications, 2013. (En línea.)
- [5] Campos Laclustra, J.: *Apuntes de Estructuras de Datos y Algoritmos*,

Luis Ángel Rodríguez Valdelamar	investigador	Ayuda a resolver inquietudes sobre las funciones del programa
--	--------------	---

		funciones del programa
	Observador	Revisa el progreso del programa cada 2-3 días después de distribuir las responsabilidades
Alejandro Mendivelso Torres	Coordinador	Habla con todos los integrantes y coordinar reuniones
	Experto	Crear parte del código y buscar soluciones
	Investigador	Ayuda a resolver inquietudes sobre las funciones del programa
	Observador	Revisa el progreso del programa cada 2-3 días después de distribuir las responsabilidades
	Observador	Revisa el progreso del programa cada 2-3 días después de distribuir las responsabilidades
	Técnico	Apoya en cada una de las actividades con lo requerido

- segunda edición, 2018. (En línea.)
- [6] Martí Oliet, N., Ortega Mallén, Y., Verdejo López, J.A.: *Estructuras de datos y métodos algorítmicos: 213 ejercicios resueltos*. 2ª Edición, Ed. Garceta, 2013.
- [7] Joyanes, L., Zahonero, I., Fernández, M. y Sánchez, L.: *Estructura de datos. Libro de problemas*, McGraw Hill, 1999.