

UNcomedor

Natalia Andrea Quiroga Castillo, Sofia Quimbay Cadena, Laura Camila López Pardo, Andres Fernando Rojas Pedroza, Julián Esteban Villate Pinzón

Grupo de trabajo 1

I. INTRODUCCIÓN

En el presente documento se busca presentar UNcomedor, el cual es un programa que da solución al problema de las aglomeraciones en los comedores de la Universidad Nacional de Colombia sede Bogotá; presentando un programa que utiliza los conceptos de estructuras de datos para ayudar a la comunidad, dentro del documento se encuentra mas explicado tanto el problema como el funcionamiento del programa, a quienes va dirigido, una descripción de la interfaz de usuario, entornos de desarrollo y operación, el prototipo de software inicial, para este prototipo se presenta solo mediante herramientas de desarrollo, dado que la interfaz no será presentada en esta entrega del proyecto.

A su vez se presenta el diseño, implementación y aplicación de las estructuras de datos en el programa, las pruebas del prototipo y análisis comparativo, la información de acceso al video demostrativo del prototipo de software, los roles y actividades, las dificultades y lecciones aprendidas, y se termina con las referencias que fueron usadas en todo el documento.

II. DESCRIPCIÓN DEL PROBLEMA A RESOLVER

Durante el periodo actual del 2022-1 se viene presentando un aforo máximo en los comedores, este es generado por múltiples factores que se presentan desde tiempo atrás, como lo son la alta demanda en los comedores más centrales y la concurrencia en horas particulares por parte de la comunidad, sin embargo se vuelve un problema de solución urgente dadas las nuevas medidas de seguridad en la salud que iniciaron desde la pandemia del 2019, las cuales intentan evitar estos aforos dando soluciones como el distanciamiento, uso constante de tapabocas y lavado de manos.

Sin embargo los aforos no son consecuencia de la pandemia, sino que son eventos que se presentan desde tiempo atrás, y que perjudican el ambiente, bien sea por lo extenuante que resulta hacer la fila larga, o lo incómodo de las aglomeraciones e incluso si se presenta junto con la necesidad de comer rapido por el horario que existe académico.

Uncomedor busca disminuir estos aforos y facilitar a la comunidad una espera adecuada en las filas de los comedores. El propósito es disminuir el tiempo de espera, regular los aforos y brindar al usuario una experiencia controlada de la

situación en tiempo real de los comedores de la Universidad Nacional de Colombia sede Bogotá.

III. USUARIOS DEL PRODUCTO DE SOFTWARE

En el programa se esperan dos roles inicialmente, el de administrador y el de usuario, el perfil de administrador va a tener acceso a todos los datos dentro del programa, puede ver las colas en cada uno de los comedores, puede ver cuantas personas hay en cada comedor, que turno tiene cada persona y puede modificar los datos que se le presentan al usuario, como lo es por ejemplo la disponibilidad de variedad de almuerzo o si está habilitado o no el comedor.

Mientras que en el otro perfil identificado como el usuario se espera poder ver la lista de comedores, la disponibilidad en cada uno de los comedores, ver que almuerzo ofrecen y la disponibilidad de los menús; a su vez la función principal de solicitar un turno en el comedor elegido, y poder realizar el pago (esta función estará disponible en actualizaciones futuras).

Esta clasificación se realiza teniendo en cuenta las funciones que son útiles para cada perfil, a los usuarios en este caso los y las estudiantes necesitan tener acceso a la información de disponibilidad de comedores mientras que los administradores necesitan saber cuánta demanda tendrá cada comedor y tener la posibilidad de cambiar la información que se le brinda al usuario.

IV. REQUERIMIENTOS FUNCIONALES DEL SOFTWARE

● *Guardar usuarios*

Descripción: A través de las listas enlazadas con cola se busca almacenar los datos del usuario, como un objeto de tipo usuario, en un nodo de tipo genérico. Esto con el objetivo de que la ejecución sea más eficiente en términos de tiempo.

- *Acciones iniciadoras y comportamiento esperado:* El usuario deberá diligenciar datos básicos, como nombre, identificación, correo electrónico, tipo de apoyo socioeconómico y al comedor en el que desea hacer la fila y esperar su turno

Requerimientos funcionales: Almacenamiento de datos

- *Buscar usuarios*

Descripción: Se hará una búsqueda lineal a través de la lista enlazada donde se podrá visualizar los datos básicos del usuario al retornar la función que tiene como parámetro de búsqueda el id del usuario, una vez buscado el usuario se puede eliminar el elemento o conservarlo en la lista.

- *Acciones iniciadoras y comportamiento esperado:* Al elegir la opción de búsqueda de usuarios en el menú principal se indaga el id de usuario, en caso de que el usuario no se encuentre en la lista se retornara un aviso y en caso contrario se visualizarán los datos básicos del usuario en cuestión.

Requerimientos funcionales: Búsqueda y eliminación de datos

- *Manejo de comedores*

Descripción: Creación de cada objeto de tipo comedor con sus respectivos atributos, almacenamiento en una lista enlazada, función de búsqueda y eliminación de alguno de los comedores a través de su nombre.

- *Acciones iniciadoras y comportamiento esperado:* El administrador de la aplicación podrá crear un nuevo comedor con los atributos de su nombre y la capacidad de ocupación, actualizarlo, buscarlo o eliminarlo a través del menú principal del usuario administrador.

Requerimientos funcionales: Almacenamiento, búsqueda, eliminación de datos

- *Creación de turnos y encolar este en el comedor correspondiente*

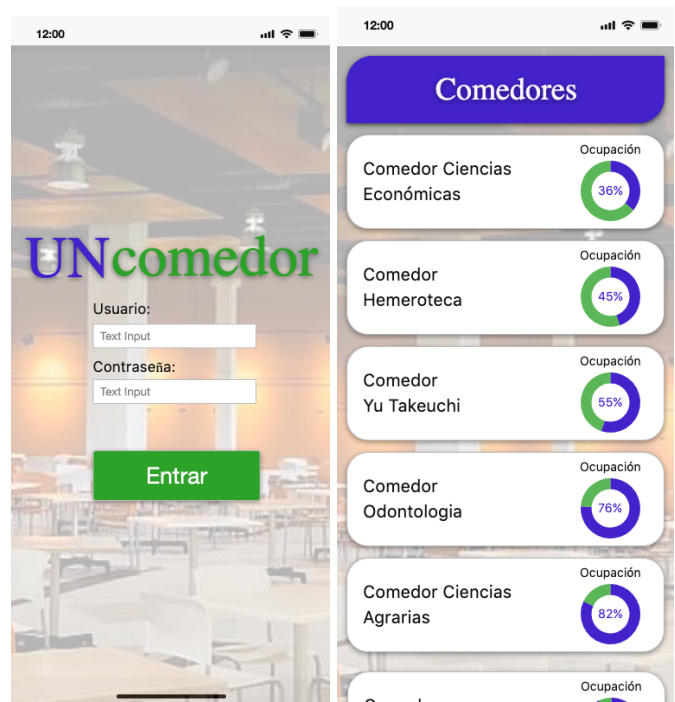
Descripción: A través de esta funcionalidad se crea un objeto de tipo Turno con los datos del usuario y el comedor correspondiente en el que desea tomar el turno. Posterior a esto se accedera a la cola del comedor escogido y se agregara el turno a esta.

- *Acciones iniciadoras y comportamiento esperado:* Al ingresar sus datos el usuario escogerá un comedor, a este se le asignará un turno respecto al orden de llegada y posteriormente se mostrará en la consola la cola entera de cada uno de los comedores con el número de turno y el nombre del usuario.

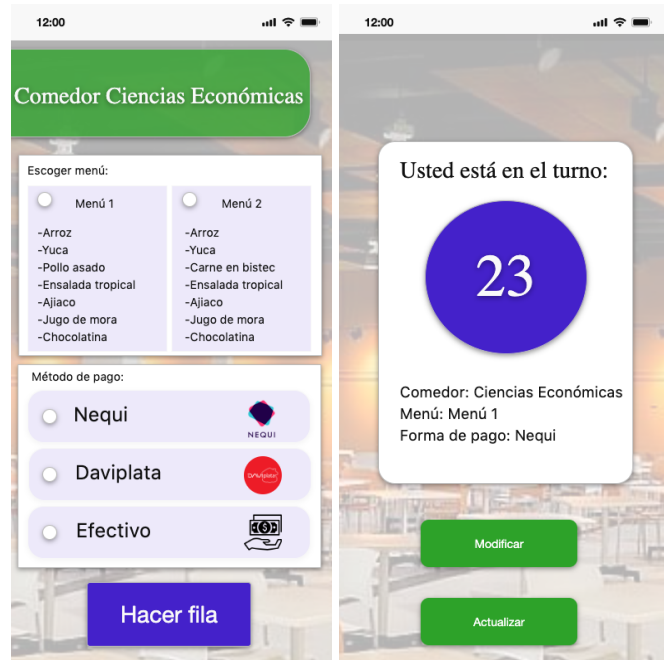
Requerimientos funcionales: Actualización y almacenamiento de datos

V. DESCRIPCIÓN DE LA INTERFAZ DE USUARIO PRELIMINAR

Al ingresar a la aplicación el usuario podrá registrarse o iniciar sesión en su cuenta, posterior al login se visualizará en pantalla la lista de comedores, ordenados de menor a mayor ocupación.



Al escoger el comedor que se desea se abrirá un panel de opciones a escoger como que menú desea el usuario y que método de pago se utilizará para la transacción, una vez estos campos sean completados se procederá a hacer la fila virtual. En la siguiente pantalla se visualizará el turno en el que se encuentra el usuario y los datos del mismo con la opción de actualizar la fila o modificar los datos previamente seleccionados de menú y método de pago.



Por último se encuentra la opción de haber ingresado como administrador del comedor, donde se encontrará el número de usuarios en fila con su respectiva opción de comida y forma de pago; de esta manera se llevará registro de a quien se le ha entregado su comida y de esta manera se actualizará la fila.



VI. ENTORNOS DE DESARROLLO Y DE OPERACIÓN

En el entorno que se utilizó para desarrollar el proyecto fueron los IDE Visual Studio y Netbeans, a su vez se utilizó el lenguaje de programación Java, para las reuniones de grupo se utilizó la plataforma de videollamadas meet. Se espera que el proyecto cuando se ponga en operación pueda ser utilizado como aplicación móvil para dispositivos Android.

VII. PROTOTIPO DE SOFTWARE INICIAL

El primer prototipo se encuentra en el repositorio en Github: <https://github.com/camilall/ProyectoED>

En dicho repositorio se encuentra el README.md con la descripción del prototipo, dentro de la carpeta de UNcomedor se encuentra la carpeta src donde se encuentra almacenado el código fuente, como las clases y el main. En la carpeta de docs hallamos la documentación del proyecto, es decir el informe y las diapositivas con la explicación del prototipo. La carpeta Data contiene los datos de prueba utilizados y en lib las librerías utilizadas.

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

Para el desarrollo de nuestro primer prototipo de Software de UNcomedor decidimos implementar las estructuras de datos de colas con referencias, arreglos unidimensionales y listas enlazadas, para almacenar y administrar los datos de los usuarios del programa, empezando por la lectura y manejo de los inputs en formas de arreglos, donde cada usuario se encola en la estructura lineal de cada comedor, la cual es una cola. A

continuación se visualizan los comedores con sus respectivos turnos en orden de llegada de los usuarios. Por último se implementaron las listas enlazadas en el almacenamiento y la búsqueda de usuarios en el sistema

En la clase cola encontramos su constructor, y los métodos empty para verificar la lista, enqueue para encolar en la respectiva cola de cada comedor según la elección del usuario, dequeue para así mismo desencolar un usuario que ya fue atendido y un último método para imprimir la cola actual de los comedores. Añadiendo el hecho de que se creó una clase nodo, la cual almacena los datos genéricos, con el fin de que los nodos pudieran abordar y trabajar con distintos objetos y clases de datos

Para trabajar con los usuarios se creó una lista enlazada específica para ellos con los métodos para añadir al inicio o al final de la lista, buscar y borrar un elemento y para imprimir a los usuarios en el formato deseado.

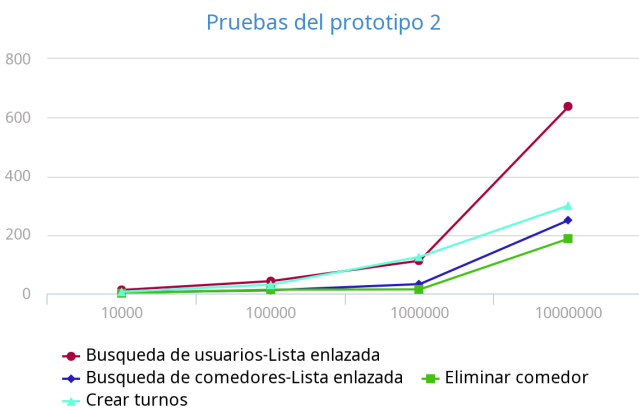
A su vez para el manejo de los comedores se acordó que la estructura más óptima para manejarlos eran las listas enlazadas simples con cola; donde se crean los métodos para su creación, inserción de datos, eliminación y búsqueda, consulta y constante actualización.

IX. PRUEBAS DEL PROTOTIPO Y ANÁLISIS COMPARATIVO

Para realizar el análisis comparativo de las funcionalidades de los datos se tomó el tiempo de 7 funciones, las primeras 3, las cuales tuvieron un tiempo de ejecución mayor fueron las de insertar los usuarios en una lista enlazada, insertar comedores en la lista enlazada e imprimir las colas de los comedores. Las últimas 4 tuvieron un tiempo de ejecución considerablemente menor, por lo que se graficaron por aparte. Las funciones de la gráfica 2 son el de buscar el último elemento de la lista de usuarios, en la de comedores, crear turnos y además una eliminación de los últimos elementos

	10.000	100.000	1'000.000	10'000.000
Insertar usuarios	294	8575	2256957	8545866
Insertar comedor	470	40034	4308098	13094870
Imprimir comedor	408	16300	6453802	25815208
Crear turno	6	30	125	300
Búsqueda comedor	4	6	4	4
Búsqueda comedor inexistente	1	15	32	65

Eliminación comedor	2	10	14	33
Eliminación comedor inexistente	1	4	13	34
Búsqueda usuarios	12	42	112	638



En la gráfica uno podemos evidenciar un aumento exponencial en los tiempos de ejecución, en especial desde el millón de datos a almacenar, evidenciando así tanto en la inserción en las listas enlazadas como con el manejo de las colas un tiempo de $O(n)$.

En cuanto al análisis de la segunda gráfica se evidencia que las funcionalidades son bastante eficientes después de cargados los datos, pero igualmente hay un aumento significativo en el tiempo después del millón de datos. De esta manera vemos que se comporta en un tiempo lineal, es decir, $O(n)$.

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

Video demostrativo del primer prototipo:
<https://youtu.be/vd4cQINb8A4>

XI. ROLES Y ACTIVIDADES

INTEGRANTE	ROL(ES)	ACTIVIDADES REALIZADAS (Listado)
Natalia Andrea Quiroga Castillo	Líder	<ul style="list-style-type: none"> - Coordinación y designación de las tareas a realizar por cada integrante. - Organización de las reuniones de seguimiento del proyecto.
		<ul style="list-style-type: none"> - Creación de las estructuras de datos, lectura de datos, creación de turnos e inserción de estos en las colas. - Documento - Interfaz de usuario preliminar - Pruebas.
Sofia Quimbay Cadena	Experto	<ul style="list-style-type: none"> - Creación de distintos casos de prueba para evidenciar más rápido los errores en el programa.
	Secretaria	<ul style="list-style-type: none"> - Documento - Llevar registro de lo realizado durante las reuniones.
Laura Camila López Pardo	Coordinador	<ul style="list-style-type: none"> -Facilitar el acceso de los integrantes a la plataforma de github.
	Observador	<ul style="list-style-type: none"> -Monitorear las modificaciones al repositorio y comentar cualquier irregularidad.
Andres Fernando Rojas Pedroza	Investigador	<ul style="list-style-type: none"> -Realización de pruebas por distintos métodos de implementación
	Animador	<ul style="list-style-type: none"> -Ser optimista ante las dificultades
Julián Esteban Villate Pinzón	Secretario	-

	Tecnico	-Proporcionar y generar el modelo de los datos usados en las pruebas -Revisar la aplicación de conceptos de programación orientada a objetos y las estructuras de datos
--	---------	--

XII. DIFICULTADES Y LECCIONES APRENDIDAS

Las principales dificultades en la elaboración del proyecto fueron la coordinación por parte de todo el equipo, el planeamiento del proyecto y el correcto manejo del repositorio, ya que se nos dificultaba encontrar la manera de dividimos la programación de las posibles funciones del prototipo del software.

Durante el proceso tuvimos valiosos aprendizajes, como el correcto planteamiento de los problemas a abordar desde un inicio, un correcto mejoramiento de los comandos de git y en el próximo ciclo de trabajo realizar reuniones con mayor frecuencia para ir viendo los avances de cada uno de los integrantes del grupo.

Encontrar herramientas que ayuden a la hora de controlar los tiempos, dado que una de las dificultades que se presentó fue a la hora de cumplir con los plazos establecidos para cada parte de esta primera entrega, tanto por la falta de claridad de estas mismas como por aplazar los tiempos para último momento.

XIII. REFERENCIAS BIBLIOGRÁFICAS

- [1] J. T. Streib y T. Soma, Guide to Data Structures. Cham: Springer International Publishing, 2017. Accedido el 13 de mayo de 2022. [En línea]. Disponible: <https://doi.org/10.1007/978-3-319-70085-4>
- [2] X. F. Gutierrez, Estructuras de Datos. Alfaomega Grupo Editor, 2003.
- [3] División de bienestar Sede Bogotá, Universidad Nacional de Colombia. <http://www.bienestar.unal.edu.co/>