**Camilo Aguilar, Catalina Goméz, David Nagles, Kelly Solano, Ivana Pedraza**

**No. de Equipo Trabajo: N° 2**

Proyecto Find Ur Chaza

# INTRODUCCIÓN

En la universidad Nacional de Colombia además de destacar como universidad por su excelente calidad educativa a nivel Latinoamérica, se reconoce entre la comunidad universitaria por poseer pequeñas tiendas o carpas dirigidas por los mismos estudiantes, las cuales se encuentran dentro de la universidad. Estas tiendas, o también conocidas popularmente como ‘Chazas’, disponen de la venta de diversos productos como lo son materiales escolares, ropa, accesorios, entre otros. Principalmente, entre ellas se destacan las chazas de venta de comida y almuerzos para los estudiantes. Las chazas se han convertido en indispensables para la vida del estudiantado y una fuente de ingresos para varios de estos estudiantes, que con sus emprendimientos solucionan más de un inconveniente a los demás. Sin embargo, varios problemas persisten debido a que muchas veces algunas chazas no dan abasto para la enorme demanda, lo que se termina manifestando en filas larguísimas y falta de oportunidad para esta cuestión es en esencia un problema de comunicación y organización de la información de la disponibilidad de productos o de personal de las chazas. *Find Ur* *Chaza* es un ejecutable que permitirá solucionar efectivamente estas complicaciones con ayuda de la programación, desarrollado las herramientas necesarias dadas distintas estructuras de datos que nos otorgan la oportunidad de trabajar con grandes cantidades de información.

# DESCRIPCIÓN DEL PROBLEMA A RESOLVER

Las chazas de venta de comida o almuerzos se destacan por vender sus productos a un precio asequible para los mismos estudiantes que muchas veces no pueden traer el almuerzo desde sus casas, de igual forma, las chazas son de gran utilidad para generar ingresos hacía aquellos estudiantes que manejan estas mismas como vendedores. Sin embargo, la falta de un sistema de información sobre la ubicación, los artículos de venta, los horarios de atención, y los precios de los productos que contenga cada chaza, genera una desinformación de estas mismas para los estudiantes como clientes, siendo esto un impacto negativo para los ingresos de los vendedores, generando poca rentabilidad. Por consecuencia, el objetivo principal de este proyecto consta en el manejo, organización y creación de la información confiable sobre cada chaza perteneciente a la Universidad Nacional de Colombia, siendo de gran utilidad para los estudiantes evitando incertidumbres sobre la venta de comida.

# USUARIOS DEL PRODUCTO DE SOFTWARE:

En este proyecto se identifican tres roles principales de usuarios: el cliente, el vendedor y el administrador de aplicación.

El vendedor es aquel que va a ofrecer sus productos en la plataforma, que, a su vez, tendrá que *gestionar*, dicho de otra forma, el vendedor tendrá la responsabilidad de crear y mantener los datos de su negocio actualizados, proporcionando información detallada sobre el negocio, por ejemplo, los productos: su nombre, descripción, imágenes, y precios. Todo esto con el fin de lograr un correcto alcance en el mercado. Por su parte, el cliente es al cual va dirigido el negocio, por tal motivo es importante tener un registro de su actividad dentro del sistema. Dentro de sus oportunidades se cuenta el visualizar la lista de producto ofrecidos por la chaza y su disponibilidad.

El administrador de aplicación es el encargado de administrar, controlar y gestionar las funcionalidades, acciones y permisos que tiene los diferentes perfiles de la aplicación. Este debe asegurarse que todas las estructuras de datos en las que se basan las funcionalidades funcionen correctamente y entreguen los resultados esperados en los tiempos óptimos. Realizar informes de rendimiento del negocio (cantidad de operaciones realizadas, efectividad de venta de productos, rendimiento de las estructuras de datos implementadas, etc.). En resumen:

|  |  |
| --- | --- |
| **Usuarios del producto de software** | **Funcionalidades** |
| Administrador de aplicación | * Gestión de funcionalidades de la aplicación del negocio. |
| Vendedor de la chaza | * Proporciona la información general del negocio. * Presenta actualización de datos. |
| Cliente | * Visualizar productos ofrecidos en conjunto con su disponibilidad. |

# REQUERIMIENTOS FUNCIONALES DEL SOFTWARE

**Funcionalidades para el rol de vendedor:**

Funcionalidad: Registrar negocio.

Descripción: Para que el negocio sea visible para los usuarios sobre la plataforma.

Acciones iniciadoras y comportamiento esperado:

1. El vendedor pueda tener la opción a seleccionar de crear su negocio.
2. Al seleccionar la opción de crear su negocio, automáticamente podrá realizar la funcionalidad del registro de datos sobre el negocio.

Funcionalidad: Registro de datos sobre el negocio.

Descripción: Para que el perfil del negocio cuente con información relevante, como descriptiva y de contacto.

Acciones iniciadoras y comportamiento esperado:

1. El vendedor inicia sesión en el sistema y selecciona en la ventana de inicio “Registro datos negocio”, se presenta una pantalla de registro de los datos necesarios para describir la chaza, como nombre, ubicación, horarios de atención y descripción.
2. Una vez completados los datos, el vendedor selecciona la opción de “Guardar cambios” y el sistema almacena esta información en la base de datos.

Funcionalidad: Borrado del negocio

Descripción: Por circunstancias variadas, el vendedor puede llegar a la decisión de eliminar la cuenta de su negocio.

Acciones iniciadoras y comportamiento esperado:

1. El vendedor al ingresar a la ventana de inicio de su negocio debe seleccionar la opción de “Eliminar negocio”.
2. Al seleccionar la opción de eliminar su negocio, automáticamente le saldrá un cuadro de confirmación, esto con el fin de prevenir una eliminación accidental.

Funcionalidad: Actualización de los datos sobre el negocio.

Descripción: Permite a los vendedores hacer modificaciones de los datos sobre su negocio.

Acciones iniciadoras y comportamiento esperado:

1. El vendedor inicia sesión en el sistema y selecciona en la ventana de inicio “Actualizar datos negocio”, se presenta una pantalla que muestra los datos del negocio que están registrados, situando casillas para ingresar los datos a modificar, como nombre, ubicación, horarios de atención y descripción.
2. Una vez llenado los campos que se quieren modificar, el vendedor selecciona la opción “Guardar cambios” y el sistema almacena la información que fue modificada.

Funcionalidad: Registro de anuncios importantes sobre el negocio.

Descripción: Permite a los vendedores publicar anuncios relevantes a su chaza.

Acciones iniciadoras y comportamiento esperado:

1. El vendedor al iniciar sesión selecciona en la ventana de inicio la opción de “Publicar anuncio”, que presenta una pantalla de registro de anuncios, debe llenarse las casillas de nombre, descripción del anuncio e imágenes a anexar.
2. Cuando se completen las casillas, debe seleccionar la opción “Publicar anuncio” y este se almacenará en el sistema.

Funcionalidad: Borrado de anuncios importantes sobre el negocio.

Descripción: Permite a los vendedores publicar novedades sobre su negocio.

Acciones iniciadoras y comportamiento esperado:

1. El vendedor al iniciar sesión selecciona en la ventana de inicio la opción “Borrar anuncio”, que presenta una ventana que muestra los anuncios publicados y permite seleccionar los que se quieran eliminar.
2. Una vez seleccionado, se debe seleccionar “Guardar cambios”. La modificación quedara almacenada en la plataforma.

Funcionalidad: Registrar nuevos productos de la chaza.

Descripción: Permite al vendedor ingresar los nuevos productos que oferta en su negocio.

Acciones iniciadoras y comportamiento esperado:

1. El vendedor inicia sesión en el sistema y podrá seleccionar la opción de “Registro de productos”.
2. El sistema al seleccionar “Registro de productos” presenta una pantalla de registro de productos donde se pueden ingresar los datos necesarios, como el nombre del producto, descripción, precio, ingredientes y una imagen.
3. Una vez completados los datos, el vendedor selecciona la opción de “Guardar cambios” y el sistema almacena el producto o los datos del negocio en la base de datos.

Funcionalidad: Visualización del total de productos que oferta la chaza.

Descripción: El vendedor puede ver todos los productos que oferta.

Acciones iniciadoras y comportamiento esperado:

1. El vendedor inicia sesión en el sistema y selecciona la opción de “Visualizar productos”.
2. Al seleccionar la opción, el sistema redirigirá a una ventana que cuenta con la tabla de todos los productos almacenados hasta el momento, donde es posible ordenar los productos de acuerdo las columnas de nombre de producto, descripción, precio, ingredientes y fecha de ingreso del producto.

Funcionalidad: Visualización de productos por categoría.

Descripción: El vendedor puede filtrar los productos por categorías como por nombre, descripción, precio, ingredientes y fecha de ingreso del producto.

Acciones iniciadoras y comportamiento esperado:

1. El vendedor al momento de visualizar los productos puede seleccionar las casillas superiores de la tabla y filtrar por dichas categorías. Deben existir casillas para realizar las búsquedas específicas.

Funcionalidad: Actualización de producto(s) ofertado(s).

Descripción: El vendedor mantiene actualizado la información de los productos.

Acciones iniciadoras y comportamiento esperado:

1. El vendedor al ingresar a la ventada de “Visualizar productos” tendrá la opción de “Actualizar producto”, que lo redirigirá a una ventana emergente para actualizar el producto escogido.
2. Una vez modificados los campos a actualizar, selecciona la opción de “Guardar cambios”. La información se guardará en el sistema.

Funcionalidad: Borrado de producto(s) ofertado(s).

Descripción: El vendedor elimina productos que ya no tengan oferta por parte de la chaza.

Acciones iniciadoras y comportamiento esperado:

1. El vendedor al ingresar a la ventana de “Visualizar productos” tendrá la opción de “Eliminar producto” y seleccionar los que desee borrar. Al momento de dar clic a eliminar, sale un mensaje de confirmación.
2. Cuando se confirma la opción “Si” se eliminaran los productos del sistema y automáticamente se actualizará la visualización de los productos.

Funcionalidad: Visualizar ordenes pendientes

Descripción: El vendedor podrá ver las ordenes que tiene el negocio por realizar.

Acciones iniciadoras y comportamiento esperado:

1. El usuario vendedor tendrá la capacidad de visualizar en su ‘menú’ las ordenes de los clientes guardados en el programa.

Funcionalidad: Actualización del estado de las órdenes.

Descripción: El vendedor actualizara los estados de las ordenes que los clientes hayan solicitado.

Acciones iniciadoras y comportamiento esperado:

1. El vendedor inicia sesión en el sistema y selecciona la opción de “Visualizar órdenes”, en esta tiene la opción de actualizar el estado de las ordenes, entre “Pendiente”, “En Proceso” y “Finalizado”. Debe dar clic en “Actualizar”.
2. Una vez seleccionado, se debe guardar en el sistema. La información actualizada se verá automáticamente.

Funcionalidad: Eliminar orden(es) pendiente(s).

Descripción: El vendedor puede eliminar ordenes que por circunstancias determinadas no pueden realizarse, o también cuando una orden ya fue finalizada.

Acciones iniciadoras y comportamiento esperado:

1. El vendedor al ingresar a la ventana de “Visualizar órdenes” tendrá una opción de “Eliminar orden”, deberá seleccionar las ordenes que va a eliminar, y confirmar que si quiere eliminar por medio del mensaje emergente que aparece.
2. Una vez confirmado el borrado, la ventana de visualización de productos se actualizará automáticamente.

|  |  |
| --- | --- |
| **Requisitos Funcionales del sistema** | **Descripción** |
| Inicio de sesión de usuario | El usuario se le permite ingresar al sistema por medio de una ventana de login ingresando su usuario y contraseña. |
| Administración de productos en venta | El usuario vendedor puede controlar los productos ofertados dentro de su negocio y perfil, al realizar operaciones CRUD, que mantengan la persistencia de los datos. |
| Control de modificaciones para productos y ordenes | El sistema debe solicitar confirmación a modificaciones como actualizar y eliminar productos u ordenes, para prevenir errores en la información almacenada. |
| Interfaz de usuario intuitiva | El sistema debe ofrecer una interfaz de usuario amigable tanto para los vendedores como los usuarios, donde la visualización de la información sea entendible. |
| Funcionalidad de búsqueda | Las funcionales permiten que el usuario realice búsquedas personalizadas de acuerdo con los parámetros establecidos. |
| Gestión de usuarios | El administrador gestionara a los usuarios y su acceso a la plataforma. |

|  |  |
| --- | --- |
| **Requisitos No Funcionales del sistema** | **Descripción** |
| Rendimiento | Capacidad del sistema para responder y procesar solicitudes de manera rápida y eficiente, sin importar el volumen de datos o usuarios concurrentes. |
| Seguridad | El sistema debe proteger la información consignada de los usuarios, donde solo se les permita a ellos realizar modificaciones. |
| Usabilidad | El sistema debe ser intuitivo y fácil de usar. |
| Fiabilidad | Capacidad del sistema para funcionar de manera constante y sin interrupciones, asegurando que se cumplan los objetivos de negocio y las expectativas de los usuarios. |
| Mantenibilidad | El sistema tendrá facilidad para modificar, actualizar, corregir y mantener los cambios en el tiempo. |
| Escalabilidad | El sistema poseerá la capacidad de manejar un aumento en la carga de trabajo o en el volumen de datos sin que se produzcan cambios significativos de rendimiento o en su funcionamiento. |
| Interoperabilidad | El sistema tendrá la capacidad de integrar diferentes sistemas y/o aplicaciones mejorando la eficiencia y la productividad, reduciendo el costo y aumentando la calidad del servicio |
| Portabilidad | El programa podrá ser capaz de ser ejecutado en diferentes plataformas de hardware y/o sistemas operativos sin necesidad de realizar cambios importantes en su código o estructura. |
| Operabilidad | El sistema tendrá la capacidad de ser manejado eficientemente por los usuarios, de ser mantenido y actualizado por los administradores del sistema, y de ser operado de manera confiable y efectiva. |

# DESCRIPCIÓN DE LA INTERFAZ DE USUARIO PRELIMINAR

|  |  |
| --- | --- |
| **Vista Inicial de usuarios** | |
| Diagrama  Descripción generada automáticamente con confianza bajaInicio de sesión | Registro de datos usuario |
| Registro datos chaza | Perfil del usuario |

|  |  |
| --- | --- |
| **Vista Administrador** | |
| Vista Reportes | Copia de Ventana  emergente, datos |
| Menu chazas | Vista suspensión de usuario-chaza |

|  |  |
| --- | --- |
| **Vista Vendedor** | |
| Vista productos Chaza | Descripción productos |
| Ordenes vendedor | Facturas ordenes |
| Datos de clientes | Comentarios clientes |

|  |  |
| --- | --- |
| **Vista Clientes** | |
| Menú de chazas | Productos e info. chaza |
| Info. Vendedor y reporte | Info. Productos |
| Vista ordenes cliente | Vista comentarios |
| Factura vista cliente | Vista datos usuarios |

àLink de los Mockups: <https://miro.com/welcomeonboard/NzBiMTlMakFqQjlhMG5IZjFpVlBLenF1UVZIUVhEVkl6R05aVGNHUzdvQkM2NXptVWFnZERlTVhmMm5MbzhVenwzNDU4NzY0NTUzMDI3OTExMjg3fDI=?share_link_id=915233839192>

# Entornos de desarrollo y de operación

Se utilizan los siguientes recursos de software: un repositorio en Git-Hub, NetBeans e Intellij IDEA, se desarrollará un ejecutable funcional para el sistema operativo de Windows, siendo esta una versión de escritorio, además de utilizar la página ‘Miro’ para la creación del diseño de los wireframes del ejecutable.

# PROTOTIPO DE SOFTWARE INICIAL

à<https://github.com/IvanaPedraza/Find-Ur-Chaza.git>

# IMPLEMENTACIÓN Y APLICACIÓN DE LAS ESTRUCTURAS DE DATOS

En este prototipo se implementaron tres tipos de estructuras de datos, arreglos dinámicos con cola, listas enlazadas simples y listas enlazadas con cola, cada una realiza operaciones de creación, búsqueda de único elemento o múltiples elementos, actualización de único elemento o múltiples, eliminación de elemento específico y eliminación por último elemento y primero, usando la lógica LIFO y FIFO.

* **Arreglos dinámicos con cola:** Dicha estructura cumple la función de almacenamiento de datos y de procesamiento de acuerdo con las funcionalidades planteadas. Su estructura de planteamiento permite su uso como pila o como cola, siendo la escogida en el proceso de eliminación de múltiples datos por medio de pilas. Cabe destacar que para la eliminación de datos se utilizó la implementación de las pilas (LIFO)
* **Listas enlazadas simples:** Su estructura consta en nodos enlazados entre sí. Cada nodo contiene un valor y un puntero que apunta al siguiente nodo en la lista, esto hace que la inserción y la eliminación en la cola sean rápidas, pero la búsqueda de elementos es más lenta debido a que es necesario recorrer toda la lista desde el principio para encontrar el elemento deseado. Cabe destacar que para la eliminación de datos se utilizó la implementación de las colas (FIFO)
* **Listas enlazas con cola:** Esta estructura de datos posee en cada nodo un valor específico y un respectivo puntero que nos permitió apuntar al siguiente nodo de la lista. Poseyendo particularmente dos punteros, que permiten conocer el primer elemento de la lista y al último elemento de la lista. Cabe destacar que para la eliminación de datos se utilizó la implementación de las pilas (LIFO)

# PRUEBAS DEL PROTOTIPO Y ANÁLISIS COMPARATIVO

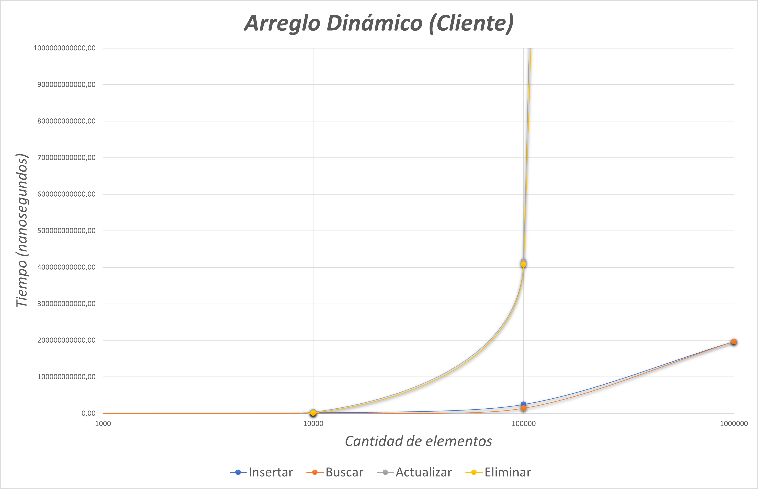
Para realizar pruebas al prototipo se tomaron las estructuras de datos a ensayar y se les asignó un tipo de dato especifico de la siguiente manera:

* **Arreglos dinámicos con cola:** Cliente, el cual cuenta con 5 atributos tipo String.
* **Listas enlazadas simples:** Vendedor, el cual cuenta con 5 atributos tipo String.
* **Listas enlazas con cola:** Chaza, el cual cuenta con los siguientes atributos: 3 Atributos tipo String y un atributo tipo Vendedor

Teniendo esto en cuenta se realizó la implementación de 4 funciones que se corresponden con algunas de las funcionalidades más recurrentes mencionadas anteriormente para cada tipo de implementación: Insertar un elemento, Buscar un elemento, Actualizar la información de un elemento y eliminar un elemento.Se realizaron pruebas de cada una de estas para uno, diez mil, cien mil y un millón de datos. Los tiempos de ejecución de cada estructura para cada cantidad de elementos se muestran a continuación expresados en milisegundos.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Tiempo de Métodos Arreglo Dinámico (Cliente)*** | | | | |
| ***No Elementos*** | ***Insertar*** | ***Buscar*** | ***Actualizar*** | ***Eliminar*** |
| ***1*** | 12,46 | 4,21 | 0,59 | 0,27 |
| ***10000*** | 1701,68 | 482,68 | 3913,85 | 1424,82 |
| ***100000*** | 24446,32 | 14136,11 | 416493,49 | 407812,06 |
| ***1000000*** | 196589,96 | 196589,96 | 43119175,56 | 32658930,00 |
|  |  |  |  |  |
|  |  |  |  |  |
| ***Tiempo de Métodos Lista enlazada (Vendedor)*** | | | | |
| ***No Elementos*** | ***Insertar*** | ***Buscar*** | ***Actualizar*** | ***Eliminar*** |
| ***1*** | 29,44 | 12,39 | 2,02 | 0,61 |
| ***10000*** | 3283,22 | 1614,69 | 11403,00 | 1270,76 |
| ***100000*** | 381711,36 | 329221,44 | 11425,14 | 583370,91 |
| ***1000000*** | 36257252,53 (10 Horas) | 32888447,29 (10 Horas) | No se realizó la prueba | No se realizó la Prueba |
|  |  |  |  |  |
|  |  |  |  |  |
| ***Tiempo de Métodos Lista Enlazada con Cola (Chaza)*** | | | | |
| ***No Elementos*** | ***Insertar*** | ***Buscar*** | ***Actualizar*** | ***Eliminar*** |
| ***1*** | 0,96 | 4,98 | 1,12 | 0,34 |
| ***10000*** | 987286,50 | 691,46 | 2316,30 | 784,81 |

Los datos anteriores muestran el siguiente comportamiento al graficarse:



*Figura 1. Análisis de tiempos implementación Arreglo Dinámico*



*Figura 2. Análisis de tiempos implementación Listas Enlazadas*

Gráfico

Descripción generada automáticamente

*Figura 3. Análisis de tiempos implementación Listas Enlazadas con Cola*

Los métodos para cada estructura de datos tuvieron como resultado la siguiente complejidad:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Complejidad (Big O)*** | | | | |
| ***Estructura*** | ***Insertar*** | ***Buscar*** | ***Actualizar*** | ***Eliminar*** |
| ***Arreglo Dinámico*** |  |  |  |  |
| ***Lista Enlazada*** |  |  |  |  |
| ***Lista Enlazada Con cola*** |  |  |  |  |

La anterior información muestra que para las implementaciones de listas enlazadas tanto con cola o sin cola, las funcionalidades se espera que tengan una complejidad lineal, para los casos de búsqueda, actualización y eliminación. Para el arreglo dinámico, debido a la lógica de duplicación de tamaño, la inserción de información tiende a ser cuadrática. En el caso de la actualización y eliminación en el arreglo dinámico, la tendencia más precisa es la logarítmica. En el caso de la lista enlazada con cola, el máximo de datos permitidos en las pruebas fue hasta 10,000 datos, que permiten inferir que ninguna de las funcionalidades tomará una complejidad constante, pero si lineal o logarítmica.

# roles y actividades

|  |  |  |
| --- | --- | --- |
| **INTEGRANTE** | **ROL(ES)** | **ACTIVIDADES REALIZADAS (Listado)** |
| Camilo Aguilar | Animador (a)  Técnico (a)  Investigador (a) | Realización de pruebas y Redacción de documentos |
| Catalina Gómez | Animador (a)  Técnico (a)  Secretario (a) | Redacción de documentos y creación de frames. |
| David Nagles | Animador (a)  Técnico (a)  Observador (a) | Creación de implementaciones, realización de pruebas. |
| Kelly Solano | Animador (a)  Técnico (a)  Coordinador (a) | Creación de implementaciones, realización de pruebas. |
| Ivana Pedraza | Animador (a)  Técnico (a)  Líder (esa) | Redacción de documentos y creación de frames. |

# DIFICULTADES Y LECCIONES APRENDIDAS

Durante el desarrollo de la primera entrega se presentaron dificultades en la disponibilidad de tiempo de los integrantes lo que retraso en poca medida el avance del trabajo. También se tuvieron dificultades en el manejo de Git en el control de las versiones desarrolladas por los integrantes, lo que incentiva en próximas entregas a estudiar los errores presentados y cómo manejarlos.

# REFERENCIAS BIBLIOGRÁFICAS

1. [1] J. T. Streib y T. Soma, Guide to Data Structures: A Concise Introduction Using Java, 1.a ed. Cham, Suiza: Springer International Publishing, 2018.
2. *Generate Test CSV Data, Test JSON Data and Test HTML Data*. (s.f.). ConvertCSV.com - Convert CSV To JSON, XML, SQL, ... <https://www.convertcsv.com/generate-test-data.htm>

Link presentación Canva: <https://www.canva.com/design/DAFhzDJ2YNs/OkcTHHe2zbbYoEz9rS638g/edit?utm_content=DAFhzDJ2YNs&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton>