**FoodAccounting**

**Camilo Andrés Herrera Gutiérrez**

**No. de equipo de trabajo: 9**

# introducción

En este documento se abordará una descripción general de nuestro proyecto, la idea general detrás de él, y los avances en el funcionamiento que se han logrado durante estas entregas.

# descripción del problema a resolver

Los restaurantes, especialmente pequeñas franquicias, pueden tener bastantes problemas logísticos a la hora de atender clientes, incluyendo, pero no limitándose a factores como equivocarse a la hora de tomar y guardar pedidos, al no tener una forma fiable de almacenarlos a la hora de su preparación, además de la cantidad de cálculos complicados que suma cada cliente y cada costo de operación del restaurante.

Esto es un gran problema ya que al no ser eficientes al momento de una atención pierden tiempo valioso que puede significar más clientes y eso monetariamente hablando pesa en cualquier sucursal pequeña.

Además de eso, y después de una pandemia que afectó enormemente la economía, es crucial poder manejar de manera óptima los gastos, costos y ganancias del restaurante.

# usuarios del producto de software

El público objetivo de nuestro proyecto son las personas con un restaurante o negocio de comida que cuente con la necesidad de atender óptimamente a bastantes clientes al día, el objetivo de este software es facilitar todo lo que tenga que ver con la organización logística y monetaria de la franquicia, de este modo buscando una mayor eficiencia al momento de la atención y el manejo propio del lugar.

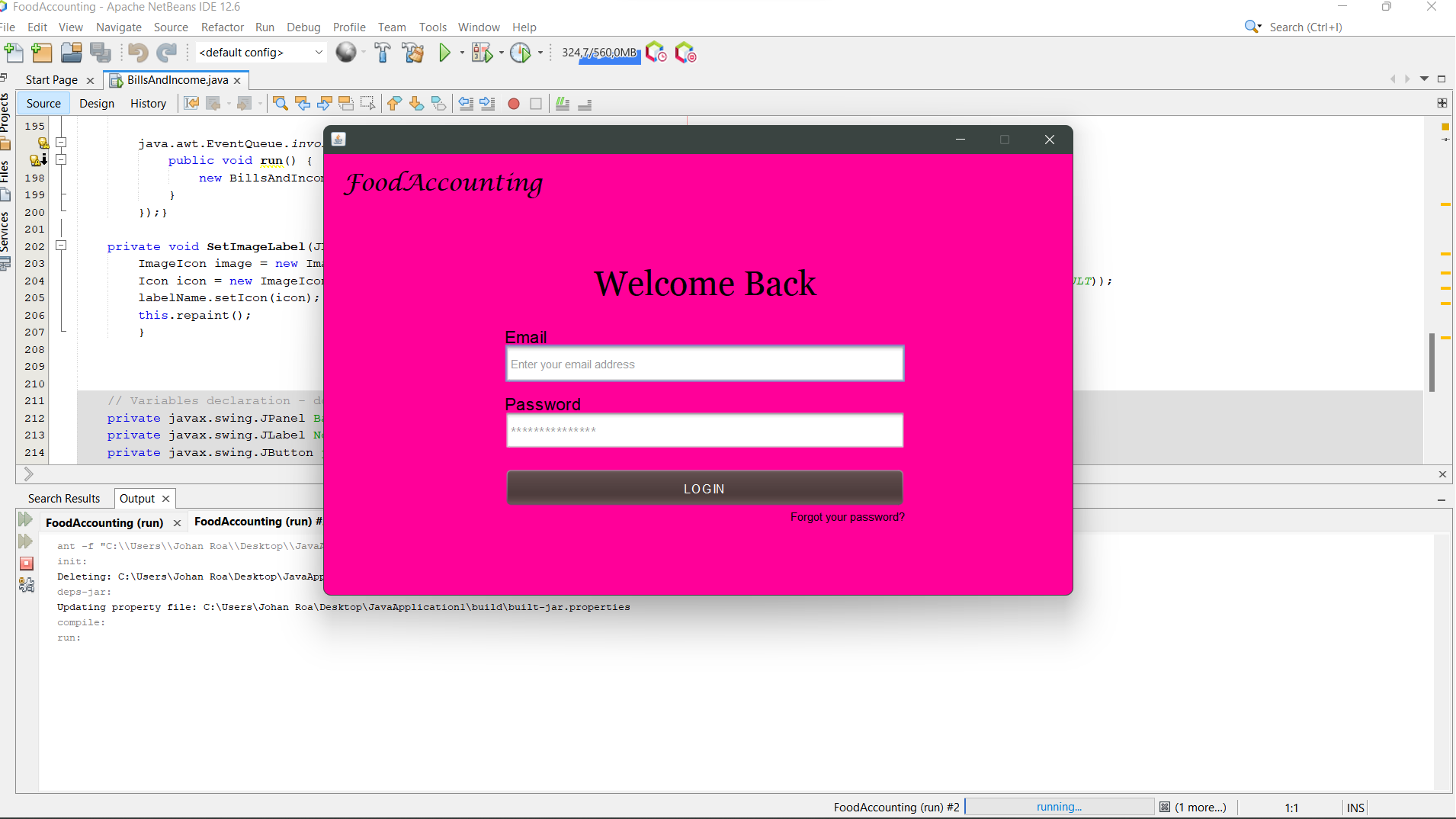
# requerimientos funcionales del software

Como requerimientos funcionales del software tenemos:

* Almacenamiento de datos de usuario.
* Estructuras para el manejo óptimo de inventario, presupuesto, productos y clientes (turnos)
* Interfaz gráfica de usuario para la fácil manipulación de los datos y organización de clientes.
* Alertas de nuevos clientes (turnos) y/o capacidad de crear turnos por atender como se requiera.

# AVANCE EN LA IMPLEMENTACIÓN DE LA INTERFAZ DE USUARIO

El prototipo de front-end proyecta la creación de una interfaz simple pero funcional, donde se encuentre fácilmente la funcionalidad que se requiera, y se maneje de forma eficiente.



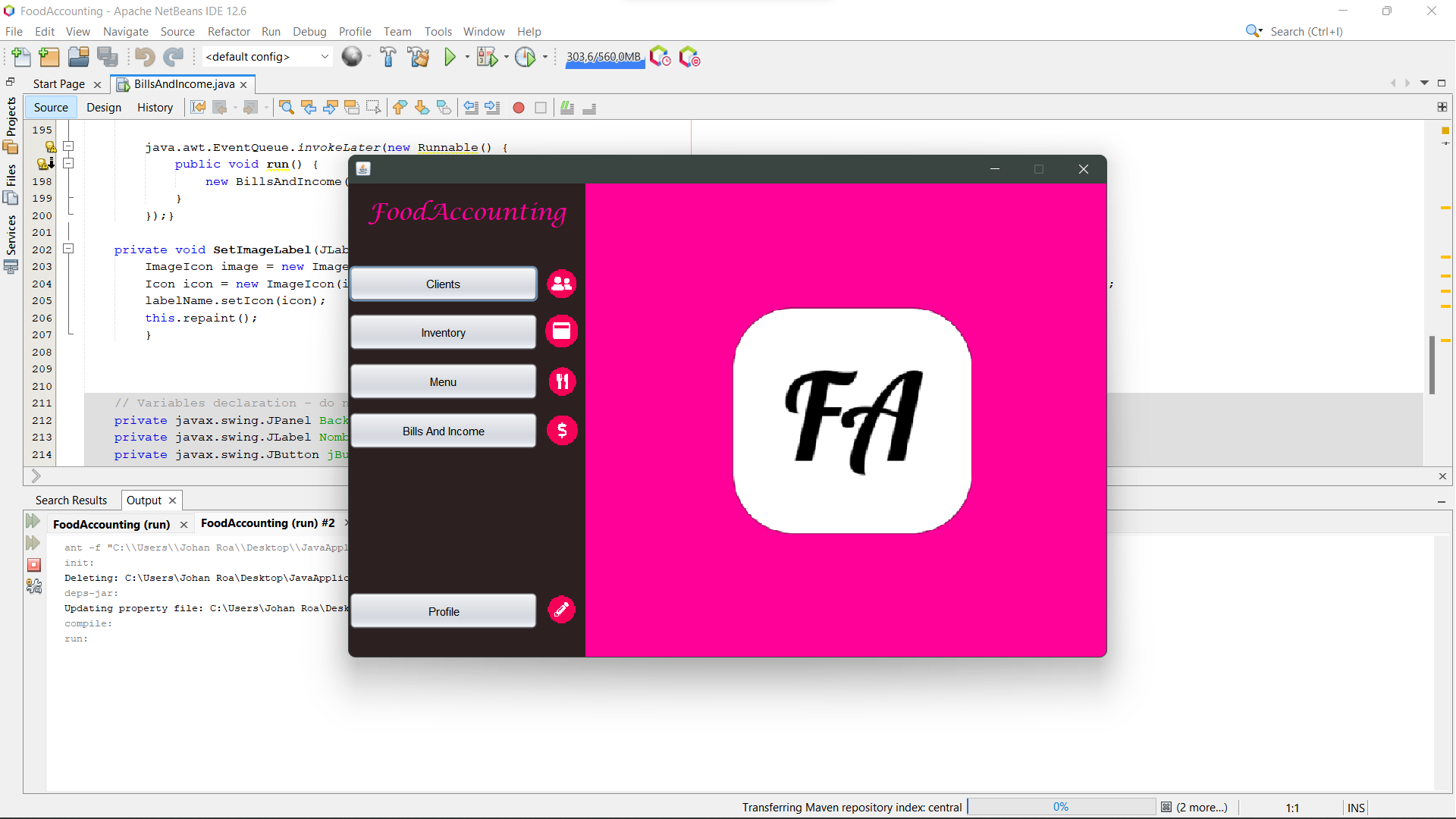


Fig. 1 y 2. Prototipos realizados con JSwing

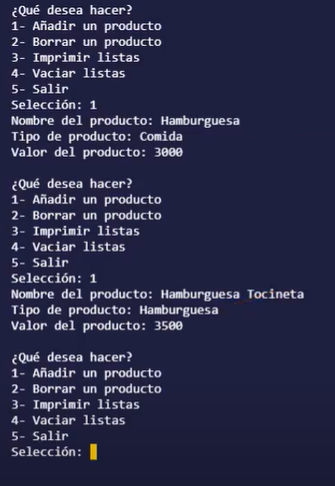
Se puede observar un avance significativo en el tema de la interfaz de usuario al implementarla con , por lo que no son simples mockups sino que ya sea algo, cuanto menos, parcialmente funcional.

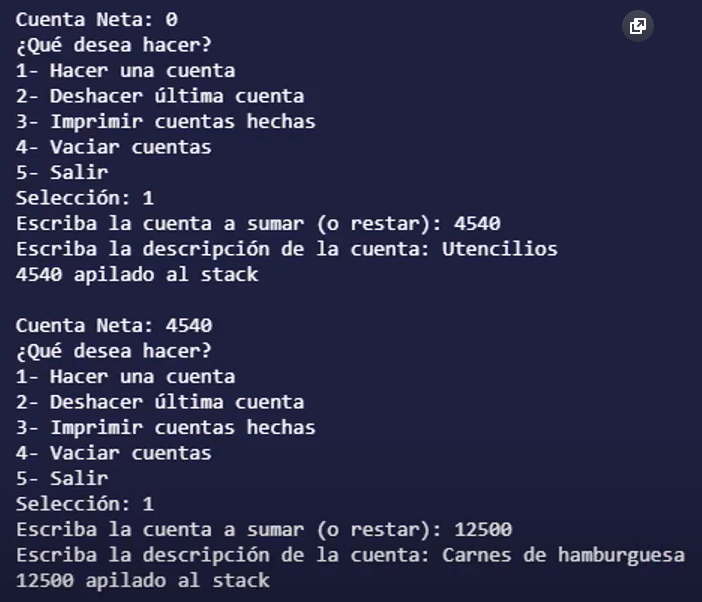
# entornos de desarrollo y de operación

Se ha venido utilizando el lenguaje Java para la implementación tanto en back-end como en front-end del programa. La ejecución se ha llevado a cabo a través de Microsoft Visual Studio Code para back-end, y Apache NetBeans para front-end. Se ha utilizado un PC con Windows 11, un procesador Ryzen 5, 5000 Series y 8GB de RAM.

# Descripción del prototipo de software.

El enfoque para arrancar el proyecto y construir un programa de funcional es que todo funcione “por debajo”. Desde las estructuras de datos, pasando por las operaciones hasta los menús que se planea implementar en JSwing, por ahora, todo corre a través de consola.





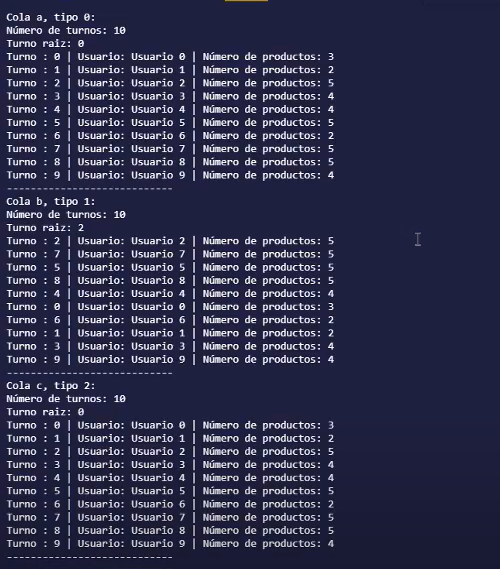


Fig. 3, 4 y 5. Funcionalidades y menús del programa, corriendo en consola

# diseño, implementación y aplicación de las estructuras de datos

Las principales estructuras de datos utilizadas en la implementación han sido, Listas Enlazadas, para el almacenamiento de los productos disponibles en un negocio. También se han utilizado para almacenar las cuentas realizadas en la funcionalidad de manejo de dinero, la cual a su vez utiliza una Pila, para llevar la cuenta neta de la caja del negocio.

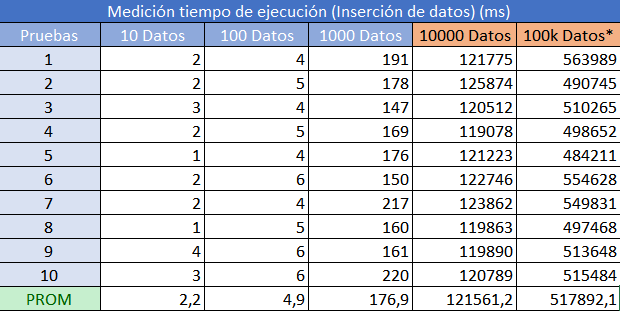
Los turnos de un negocio de comidas requieren priorizar, por lo que los turnos (clientes) del restaurante serán manejados a través de una Cola de Prioridad, la cual, a su vez, podrá priorizar alguno de estos dos ítems:

* Número de elementos en un pedido
* Orden de llegada

Finalmente, para la creación de usuarios, se ha optado por una Tabla Hash, la cual contará con la particularidad de que, los nombres de los usuarios registrados actuarán como Llaves, y estos, a su vez, serán únicos, para tratar de mitigar las colisiones. Con eso se asegura una creación y almacenamiento de usuarios de la aplicación de una forma segura y eficiente.

# pruebas del prototipo y análisis comparativo

Las pruebas realizadas a las diferentes funcionalidades con diversas cantidades de datos se evidencian en las siguientes gráficas:



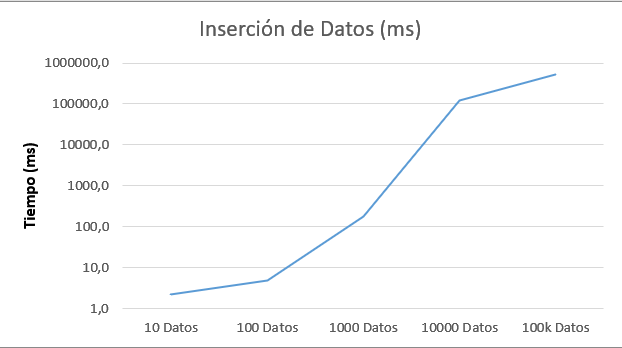
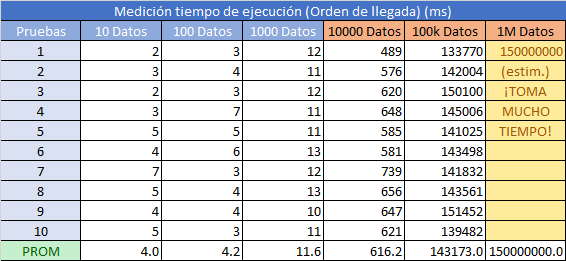


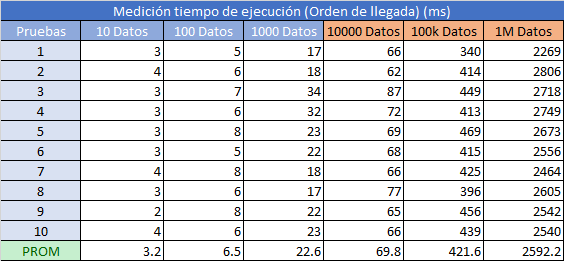
Fig. 5 y 6. Prueba de inserción de datos de productos.



Gráfico, Gráfico de líneas

Descripción generada automáticamente

Fig. 7 y 8. Prueba de cuentas de dinero.



Gráfico, Gráfico de líneas

Descripción generada automáticamente

Fig. 9 y 10. Prueba de creación de turnos. (Las prioridades no varían tiempos significativamente).

Tabla

Descripción generada automáticamente

Gráfico, Gráfico de líneas

Descripción generada automáticamente

Fig. 11 y 12. Pruebas creación de cuentas.

# Acceso al repositorio del software

<https://github.com/camiloandherrera/FoodAccounting>

# Acceso al video demostrativo del software

<https://youtu.be/Y9i0FQc8qcE>

# roles y actividades

Durante el desarrollo del proyecto los siguientes roles fueron realizados por cada uno.

| **INTEGRANTE** | **ROL(ES)** | **ACTIVIDADES REALIZADAS (Listado)** |
| --- | --- | --- |
| Camilo Herrera | Líder, experto, programador | Código, organización, documentación, repositorios, multimedia |

# dificultades y lecciones aprendidas

Algunas implementaciones tomaron más de lo esperado para que pudieran funcionar debido a dificultades técnicas, del lenguaje en sí y de planeación, especialmente la de la estructura de Cola de Prioridad.

Se tiene en cuenta para futuros proyectos que, a pesar de que no es lo ideal, es bueno referenciarse en las bibliotecas del lenguaje para hacer el proceso de *troubleshooting* en un proyecto como este.

# Referencias bibliográficas

[1] GeeksForGeeks. “Estructuras de Datos” [recurso online]: <https://www.geeksforgeeks.org/data-structures/>

[2] Deitel y Deitel. “Cómo Programar en Java”.