UNIVERSIDAD ADOLFO IBÁÑEZ

Tarea 1 Lenguaje y Paradigmas

Pizzería en C

Valentina Roselló
Jesús Salazar
Alexander Vera
María Loreto Gaya
Diego Gajardo

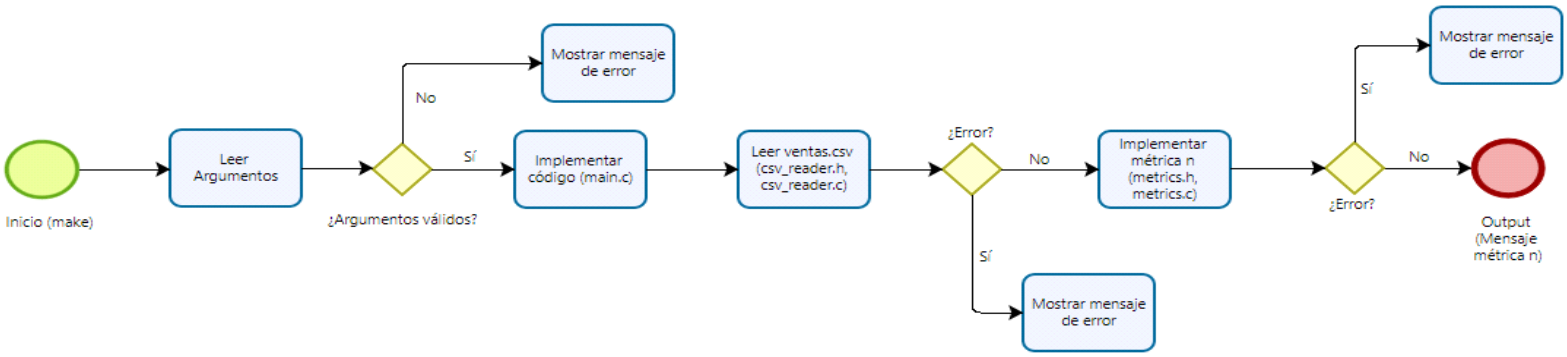
Introducción

Para esta tarea, se ha solicitado el desarrollo de un programa en lenguaje C que analice una base de datos con el historial de ventas de una famosa pizzería en Namekusei. La pizzería solicitó una cierta cantidad de métricas, conformadas por fórmulas matemáticas simples, para analizar su desempeño comercial.

Este informe explica la organización del código, la estructura de los datos a utilizar y la explicación del diseño elegido para el cálculo de las métricas requeridas. Además, se comentarán los desafíos que se encontraron en su desarrollo y las lecciones aprendidas a lo largo del proceso.

Diagrama de Flujo

A continuación, se muestra un diagrama de flujo que refleja el funcionamiento del programa en términos generales:



Razones de Diseño

Método de Parseo del CSV: Se optó por una lectura línea por línea del archivo CSV, utilizando funciones estándar de C para manejar archivos y cadenas. Esta elección se debe a la simplicidad y eficiencia que ofrece para archivos de tamaño moderado, además de evitar dependencias externas.

Almacenamiento de Ingredientes: Los ingredientes de cada pizza se almacenan como cadenas dentro de la estructura de datos que representa una venta. Esta decisión facilita el acceso y manipulación de los ingredientes asociados a cada pedido sin necesidad de estructuras de datos complejas.

Interacción entre archivos:

Archivo	**Explicación**
.vscode/...	Se encuentran varios archivos con configuraciones específicas del entorno de desarrollo en Visual Studio Code y GitHub, como

	propiedades del compilador, configuraciones de lanzamiento y tareas.
bin/app1	Es el ejecutable generado después de compilar el proyecto. Este es el archivo que se ejecuta para correr el programa.
data/ventas.csv	Archivo de datos en formato CSV que contiene las ventas de pizzas. Este archivo es leído por el programa para calcular las métricas en csv_reader.h.
include/csv_reader.h	Archivo de cabecera que contiene las declaraciones de funciones y estructuras utilizadas para leer y procesar el archivo CSV.
include/metrics.h	Archivo de cabecera que contiene las declaraciones de funciones y estructuras utilizadas para calcular las métricas de ventas.
obj/...	Directorio que contiene los archivos objeto (.o) generados durante la compilación. Estos archivos son el resultado de compilar los archivos fuente individuales. Se generan después de escribir "make" en terminal, siempre y cuando el Makefile funcione correctamente.
src/csv_reader.c	Archivo fuente que contiene la implementación de las funciones para leer y procesar el archivo CSV.
src/main.c	Archivo fuente que contiene la función principal del programa. Este archivo coordina la lectura del archivo CSV y la ejecución de las métricas solicitadas.
src/metrics.c	Archivo fuente que contiene la implementación de las funciones para calcular las diferentes métricas de ventas.
Makefile	Archivo que contiene las reglas para compilar el proyecto. Define cómo se deben compilar los archivos fuente y cómo se debe generar el ejecutable. Una vez compilado, se generan los archivos .o mencionados anteriormente.
NUL	Archivo placeholder que puede ser utilizado para propósitos específicos del proyecto o del entorno de desarrollo, en este caso, en Visual Studio Code dentro de Codespace de GitHub.

Uso de IA

Utilizamos Copilot y Chat GPT para hacer la base de datos, ayudarnos con los códigos y resolver errores. Para validar la información, fuimos probando los códigos y a su vez pasamos la base a un Excel, donde comparamos los resultados que nos tenían que dar a través de un método que ya conocíamos. Esto nos permitió saber si los códigos hacían lo que debían y de esta forma nos dimos cuenta de algunos errores.

Reflexiones Finales

Lo más complejo fue que todas las métricas funcionaran como se esperaba, ya que al final teníamos errores pequeños. Un ejemplo de esto fue en la sección del código de la pizza más vendida, en donde el código entregaba como resultado los ingredientes y después el nombre de la pizza, cuando el objetivo era que entregara el nombre solamente. Además, para la cantidad de pizzas vendidas por categoría, el código no entregaba la categoría, a pesar de que el código estaba y según nosotros

estaba correcto. Encontrar el error en los códigos fue lo más complejo, ya que fueron fallos muy pequeños y casi imperceptibles.

Para la búsqueda y resolución de estos errores utilizamos herramientas como Copilot y Chat GPT.

LECCIONES APRENDIDAS

Aprendimos la importancia de la validación de datos con distintos métodos, esto para poder detectar errores ocultos. El uso de inteligencias artificiales fue útil, pero se requirió de una revisión minuciosa. Además, por la forma de la estructura del código esto fue útil para su mantenimiento, y trabajar en C nos ayudó a mejorar en este lenguaje y su comprensión. Finalmente, aprendimos a trabajar en grupo para dividirnos las cargas de trabajo.