Informe de Diseño: Análisis de Ventas de Pizzería

Introducción

En este proyecto, se desarrolló una aplicación en C para analizar las ventas de una pizzería a partir de un archivo CSV (ventas.csv). La aplicación permite calcular métricas, las cuales son útiles para el análisis de las ventas de una pizzería, como la pizza más vendida (pms), fecha con más ventas (dms), entre otras.

Objetivo

Nuestro objetivo es proporcionar un conjunto de métricas sobre las ventas de una pizzería a partir de un archivo CSV que contiene información sobre las pizzas vendidas, el precio, la cantidad, los ingredientes y otros detalles importantes para el negocio. La aplicación busca facilitar la toma de decisiones en el área, al ofrecer detalles sobre el rendimiento de los productos.

Descripción del Proyecto

Entrada

El programa recibe como entrada un archivo CSV que contiene los datos de las ventas. Este archivo sigue la siguiente estructura:

pizza_id	order_id	pizza_name_id	quantity	order_date	order_time	unit_price	total_price	pizza_size	pizza_category	pizza_ingredients	pizza_name
1.00	1.00	hawaiian_m	1.00	01-01-2015	11:38:36	13.25	13.25	M	Classic	Sliced Ham, Pineapple...	The Hawaiian Pizza
2.00	2.00	classic_dlx_m	1.00	01-01-2015	11:57:40	16.00	16.00	M	Classic	Pepperoni, Mushrooms ...	The Classic Deluxe Pizza
3.00	2.00	five_cheese_l	1.00	01-01-2015	11:57:40	18.50	18.50	L	Veggie	Mozzarella Cheese....	The Five Cheese Pizza

Cada fila del archivo CSV representa una venta de pizza y contiene información sobre la pizza.

Salida

La salida de la aplicación son las métricas calculadas sobre las ventas:

- Pizza más vendida (pms)
- Pizza menos vendida (pls)
- Fecha con más ventas en términos de dinero (dms)
- Fecha con menos ventas en términos de dinero (dls)
- Fecha con más ventas en términos de cantidad de pizzas (dmsp)
- Fecha con menos ventas en términos de cantidad de pizzas (dlsp)
- Promedio de pizzas por orden (apo)
- Promedio de pizzas por día (apd)
- Ingrediente más vendido (ims)
- Cantidad de pizzas por categoría vendidas (hp)

Cada una de estas métricas se imprime en la consola con los valores calculados a partir del CSV.

Diseño del Sistema

La estructura del código esta dividida en los siguientes archivos:

- **main.c:** Contiene la función principal y la lógica de ejecución. Se encarga de leer el archivo CSV con las funciones de utils.c y pasar los datos a las funciones correspondientes para el cálculo de las métricas del archivo metrics.c .
- **metrics.c:** Contiene las funciones encargadas de calcular las métricas. Cada función se encarga de un cálculo específico, como el cálculo de la pizza más vendida, la fecha con más ventas, etc.
- **utils.c:** Contiene funciones auxiliares, como la lectura y el análisis del archivo CSV. Convierte los datos en una estructura que puede ser procesada por las funciones de métricas.
- **metrics.h:** Define los prototipos de las funciones del módulo metrics.c.
- **utils.h:** Define los prototipos de las funciones del módulo utils.c.

Funciones implementadas

Lectura y análisis del CSV

La función read_csv() en utils.c se encarga de leer el archivo CSV y mirar cada línea para extraer los datos. Cada venta es convertida en una estructura de tipo Order, que contiene información sobre la pizza, cantidad vendida, etc.

Cálculo de métricas

Las funciones de metrics.c calculan las métricas solicitadas. Algunas de estas métricas incluyen:

- **Pizza más vendida (pms):** Identifica la pizza que ha tenido más unidades vendidas.

- **Pizza menos vendida (pls):** Determina cuál es la pizza con menor cantidad de ventas.
- **Fecha con más ventas (dms):** Calcula la fecha con el mayor valor total en ventas (dinero recaudado).
- **Fecha con menos ventas (dls):** Calcula la fecha con el menor valor total en ventas.
- **Fecha con más pizzas vendidas (dmsp):** Encuentra la fecha con el mayor número de pizzas vendidas.
- **Fecha con menos pizzas vendidas (dlsp):** Encuentra la fecha con el menor número de pizzas vendidas.
- **Promedio de pizzas por orden (apo):** Calcula el promedio de pizzas que se venden por cada orden.
- **Promedio de pizzas por día (apd):** Calcula el promedio de pizzas vendidas por día.
- **Ingrediente más vendido (ims):** Determina cuál es el ingrediente que más veces aparece en las ventas de pizzas.
- **Cantidad de pizzas por categoría (hp):** Clasifica las ventas según las categorías de pizza (por ejemplo, clásica, veggie) y muestra la cantidad de pizzas vendidas por cada categoría.

Estructura de datos

Las ventas de pizza son almacenadas en una lista de estructuras Order, que tiene los siguientes campos:

```c
typedef struct order {
    int pizza_id;
    int order_id;
    char pizza_name_id[100];
    int quantity;
    char order_date[20];
    char order_time[10];
    float unit_price;
    float total_price;
    char pizza_size;
    char pizza_category[50];
    char pizza_ingredients[200];
    char pizza_name[100];
} Order;
```

Implementación

La aplicación hace uso de funciones modulares y punteros a funciones para separar las tareas de lectura de archivo, cálculo de métricas y presentación de resultados.

Llamado al programa

Para ejecutar la aplicación, usa el siguiente comando en el terminal, donde ejemplo.csv es el archivo de ventas y las métricas para calcular son las deseadas:

./app1 ejemplo.csv metrica1 metrica2 .....

Hay que considerar que el Código no funcionara si no se especifica al menos una métrica.

Makefile

El Makefile proporciona una forma sencilla de compilar y enlazar los diferentes módulos del proyecto. Este archivo facilita la compilación del proyecto, asegurando que todas las dependencias se manejen correctamente, sin necesidad de tener que instalar librerías o programas externos.

Explicación de uso de la IA

En este proyecto, utilizamos la herramienta ChatGPT, Claude y Copilot dependiendo del caso, para obtener la orientación sobre la implementación de Makefile en el código, estructura, coherencia entre códigos y cierto contenido de funciones. La IA proporciono ayudas para la organización del código y la interpretación de errores, lo cual nos facilito el debugging. Sin embargo, las decisiones finales fueron tomadas en conjunto, validando las sugerencias de la IA y la comprensión lógica detrás de ellas.

Conclusiones

Este proyecto nos permitió aplicar habilidades de programación en C para manejar datos de ventas y calcular métricas clave como la pizza más vendida, la fecha con más ventas, y el ingrediente más popular. A través del uso de punteros a funciones y la modularización del código, logramos mantener la aplicación flexible y eficiente.

El manejo adecuado del archivo CSV y el control de datos nos permitió cumplir con los requisitos, mientras que el uso de herramientas como GIT y Make facilitó el proceso de desarrollo. Este proyecto no solo consolidó nuestros conocimientos en C, sino que también mejoró nuestra capacidad para organizar y gestionar proyectos de programación. Además de enseñarnos nuevas técnicas para desarrollar software como el uso de Makefiles y el uso de la plataforma GitHub, la cual fue un reto para nosotros al principio, pero nos permitió mejorar nuestras habilidades al usarla.

En conclusión, para nosotros fue una experiencia valiosa que nos preparó para desafíos más grandes en el desarrollo de software y paradigmas.