

**Instituto Tecnológico de Costa Rica**  
**Escuela de ingeniería en computación**  
*IC-4700* Lenguajes de Programación  
**PY01-** Investigación

**Estudiante(s):**

Dylan Montiel Zúñiga

2023205654

**Profesor(a) a cargo:**

Allan Rodríguez Dávila

II Semestre 2024

## Manual de usuario

Este manual proporciona las instrucciones necesarias para compilar, ejecutar y usar el Sistema de Análisis de Datos de Ventas. Este proyecto fue desarrollado en una máquina virtual Kali Linux de 32 bits y utiliza el lenguaje de programación C junto con la biblioteca cJSON para manipular archivos JSON. El sistema permite importar datos de ventas desde un archivo JSON, analizar atributos, manejar datos faltantes y guardar resultados procesados. Además, ofrece funcionalidades para generar reportes y estadísticas.

### 1. Requisitos del Sistema

- 1.1. **Sistema Operativo:** Kali Linux 32-bits (o cualquier distribución de Linux compatible)
- 1.2. **Compilador de C:** GCC (GNU Compiler Collection)
- 1.3. **Biblioteca cJSON:** Para manejo de archivos JSON
- 1.4. **Editor de Código:** Cualquier editor de texto compatible con C (e.g., Visual Studio Code, Nano, Vim)

### 2. Configuración del entorno

#### 2.1. Actualizar el sistema

```
sudo apt update
sudo apt upgrade
```

#### 2.2. Instalar GCC (si no está instalado):

```
sudo apt install gcc
```

#### 2.3. Instalar la biblioteca cJSON:

```
git clone https://github.com/DaveGamble/cJSON.git
cd cJSON
mkdir build
cd build
cmake ..
make
sudo make install
```

#### 2.4. Crear un directorio para el proyecto

```
mkdir SistemaVentas
cd SistemaVentas
```

## 2.5. Copiar los archivos del proyecto al directorio creado

```
git clone https://github.com/MITTuu/PP1_Est.git
```

## 2.6. Instrucciones de compilación y ejecución

```
cd PP1_Est/Programa  
gcc main.c -o main -lcjson  
./main
```

## 3. Uso del Sistema

Al ejecutar el programa, se presentará un menú principal con las siguientes opciones:

### A. Importar Datos de Ventas

- a. Permite cargar un archivo JSON con datos de ventas al sistema. El usuario debe proporcionar la ruta del archivo al ser solicitado. El sistema verificará si el archivo contiene todos los atributos necesarios y reportará cualquier dato faltante.

### B. Procesamiento de datos

- a. Permite completar datos faltantes en cantidad y precio unitario. Se utilizan diferentes técnicas para los atributos como la moda, la media (promedio) o mediana.
- b. Eliminar datos duplicados.

### C. Análisis de datos

- a. Total de ventas.
- b. Total de ventas mensuales y anuales.

### D. Análisis temporal

- a. Permite ver el mes con mayor venta y el día de la semana más activo.
- b. Permite calcular la tasa de crecimiento o decrecimiento de las ventas en un trimestre específico.

### E. Estadística

- a. Permite visualizar el top 5 de categorías con mayores ventas.

### F. Salir del Programa

- a. Cierra el programa guardando todos los cambios realizados en un archivo JSON definido para mantener la persistencia de los datos procesados.

## Pruebas de funcionalidad

## 1. Compilación y ejecución, y menú principal

Cuando se ejecuta el programa se lee un archivo JSON destinado a la persistencia de los datos trabajados, si este archivo está vacío o hubo algún error en su lectura el programa le indicará al usuario mediante un mensaje, en este caso este archivo se encuentra vacío. El menú principal cuenta con 5 módulos ya detallados anteriormente.

```
(mittuu@kali)-[~/Desktop/PP1_Est]
$ gcc main.c -o main -lcjson

(mittuu@kali)-[~/Desktop/PP1_Est]
$ ./main
El archivo JSON está vacío, no hay datos para importar.

178
179 Sistema de Análisis de Datos de Ventas
180
181 1. Importación de datos
182 2. Procesamiento de datos
183 3. Análisis de datos
184 4. Análisis temporal
185 5. Estadísticas
186 6. Salir
187
188 Seleccione una opción: █
```

## 2. Importación de datos

Se le pide al usuario que indique la ruta del archivo del archivo JSON a leer, el programa busca el archivo y si lo encuentra lo lee y analiza, en caso de que falten atributos obligatorio se le indicará por consola la venta que no se pudo ingresar al sistema y por qué motivo. Al finalizar, si todo terminó correctamente se indicará mediante el mensaje de “Datos importados correctamente”.

```

Seleccione una opción: 1

Ingrese la ruta del archivo JSON: ventas.json

La línea 1 no se pudo importar debido a que faltan los atributos: Identificador de producto.
La línea 3 no se pudo importar debido a que faltan los atributos: Identificador de venta.
La línea 4 no se pudo importar debido a que faltan los atributos: Nombre de producto.
La línea 20 no se pudo importar debido a que faltan los atributos: Identificador de producto, Nombre de producto.

Datos importados correctamente.

```

### 3. Procesamiento de datos

Para este ejemplo arbitrariamente se importó dos veces el mismo archivo JSON, cuando se procesan los datos lo primero que se hace es ver si ids de ventas se repiten, en caso de repetirse, se eliminan. Por otro lado, en caso de que falte la cantidad de una venta, esta se calculará automáticamente con la moda, mientras que el precio unitario se puede calcular con la media o mediana, según lo desee el usuario.

```
Seleccione una opción: 2

Se eliminó el registro duplicado con venta ID 2
Se eliminó el registro duplicado con venta ID 5
Se eliminó el registro duplicado con venta ID 6
Se eliminó el registro duplicado con venta ID 7
Se eliminó el registro duplicado con venta ID 8
Se eliminó el registro duplicado con venta ID 9
Se eliminó el registro duplicado con venta ID 10
Se eliminó el registro duplicado con venta ID 11
Se eliminó el registro duplicado con venta ID 12
Se eliminó el registro duplicado con venta ID 13
Se eliminó el registro duplicado con venta ID 14
Se eliminó el registro duplicado con venta ID 15
Se eliminó el registro duplicado con venta ID 16
Se eliminó el registro duplicado con venta ID 17
Se eliminó el registro duplicado con venta ID 18
Se eliminó el registro duplicado con venta ID 19
Se eliminó el registro duplicado con venta ID 21
Se eliminó el registro duplicado con venta ID 22
Se eliminó el registro duplicado con venta ID 23
Se eliminó el registro duplicado con venta ID 24
Se eliminó el registro duplicado con venta ID 25
Se eliminó el registro duplicado con venta ID 26
Se eliminó el registro duplicado con venta ID 27
Se eliminó el registro duplicado con venta ID 28
Se eliminó el registro duplicado con venta ID 29
Se eliminó el registro duplicado con venta ID 30

Registro 6: Precio unitario faltante. Seleccione el método de imputación:
1. Media
2. Mediana
Seleccione una opción: 1

Registro 6: precio unitario reemplazado por media 29.18

Registro 7: cantidad reemplazada por moda 5
```

#### 4. Análisis de datos

Para este apartado se cuenta con un submenú donde se podrán hacer los distintos análisis vistos a continuación:

```

174 Seleccione una opción: 3
175
176
177
178 1. Total de ventas
179 2. Total de ventas mensuales
180 3. Total de ventas anuales
181 4. Volver al menu principal
182
183 Seleccione una opción: 1
184
185 Total de ventas: 6505.50

```

```

174
175
176
177
178 1. Total de ventas
179 2. Total de ventas mensuales
180 3. Total de ventas anuales
181 4. Volver al menu principal
182
183 Seleccione una opción: 2
184
185 1) Enero 2021 - Total: 125.00
186 2) Marzo 2021 - Total: 120.00
187 3) Marzo 2022 - Total: 157.50
188 4) Julio 2022 - Total: 160.00
189 5) Agosto 2022 - Total: 162.00
190 6) Septiembre 2022 - Total: 360.00
191 7) Enero 2023 - Total: 156.00
192 8) Febrero 2023 - Total: 220.00
193 9) Marzo 2023 - Total: 144.00
194 10) Abril 2023 - Total: 225.00
195 11) Mayo 2023 - Total: 210.00
196 12) Junio 2023 - Total: 200.00
197 13) Julio 2023 - Total: 210.00
198 14) Agosto 2023 - Total: 224.00
199 15) Septiembre 2023 - Total: 500.00
200 16) Noviembre 2023 - Total: 275.00
201 17) Diciembre 2023 - Total: 1886.00
202 18) Mayo 2024 - Total: 395.00
203 19) Julio 2024 - Total: 455.00
204 20) Agosto 2024 - Total: 321.00

```



Para calcular la tasa de crecimiento se le pide al usuario un trimestre (1-4) y el año que desea para el análisis.

```
185 |-----|
186 |               Análisis Temporal               |
187 |-----|
188 1. Mes con mayor venta
189 2. Día de la semana más activo
190 3. Calcular tasa de crecimiento/decrecimiento
191 4. Volver al menú principal
192 |-----|
193 Seleccione una opción: 3
194
195 Ingrese el trimestre (1-4): 2
196 Ingrese el año: 2023
197 Tasa de crecimiento para el trimestre 2 del año 2023: 22.12%
198 Tasa de crecimiento/decrecimiento: 22.12%
```

## 6. Estadística

Se muestra el top 5 categorías con mayor ventas.

```
180 |-----|
181 |               Sistema de Análisis de Datos de Ventas               |
182 |-----|
183 1. Importación de datos
184 2. Procesamiento de datos
185 3. Análisis de datos
186 4. Análisis temporal
187 5. Estadísticas
188 6. Salir
189 |-----|
190 Seleccione una opción: 5
191
192 Top 5 de categorías con mayores ventas:
193 1) Categoría 8 - Total Ventas: 1765.00
194 2) Categoría 2 - Total Ventas: 1432.00
195 3) Categoría 3 - Total Ventas: 1226.00
196 4) Categoría 6 - Total Ventas: 554.00
197 5) Categoría 4 - Total Ventas: 479.50
```

## 7. Salir

Para guardar los datos analizados se debe de terminar el programa con la opción 6.

```
180 |-----|
181 |               Sistema de Análisis de Datos de Ventas               |
182 |-----|
183 1. Importación de datos
184 2. Procesamiento de datos
185 3. Análisis de datos
186 4. Análisis temporal
187 5. Estadísticas
188 6. Salir
189 |-----|
190 Seleccione una opción: 6
191
192 Saliendo del programa...
```



## 8. Persistencia de datos

Si estás trabajando con repositorio del proyecto, los datos analizados se guardarán en el archivo “ventas\_procesadas.json”. Ejemplo:

```
1 [{
2     "venta_id": 2,
3     "fecha": "2021-01-20",
4     "producto_id": 102,
5     "producto_nombre": "Producto B",
6     "categoria": "Categoría 2",
7     "cantidad": 5,
8     "precio_unitario": 25,
9     "total": 125
10 }, {
11     "venta_id": 5,
12     "fecha": "2021-03-01",
13     "producto_id": 105,
14     "producto_nombre": "Producto E",
15     "categoria": "Categoría 2",
16     "cantidad": 6,
17     "precio_unitario": 20,
18     "total": 0
19 }, {
20     "venta_id": 6,
21     "fecha": "2022-03-15",
22     "producto_id": 106,
23     "producto_nombre": "Producto F",
24     "categoria": "Categoría 4",
25     "cantidad": 7,
26     "precio_unitario": 29.181818008422852,
27     "total": 157.5
28 }, {
```

Al compilar y ejecutar el programa, debería de aparecer el mensaje de “Datos importados correctamente.”.

```
(mittuu@kali)-[~/Desktop/PP1_Est]
$ gcc main.c -o main -lcjson
(mittuu@kali)-[~/Desktop/PP1_Est]
$ ./main
Datos importados correctamente.
```

## **Descripción del problema**

El objetivo de este proyecto es desarrollar un sistema de análisis de datos que permita gestionar, procesar y analizar ventas almacenadas en archivos JSON. Este sistema busca resolver los siguientes problemas comunes en la administración y análisis de datos de ventas:

### **A. Mantenimiento de Datos de Ventas:**

- a. Se necesita una forma eficiente de importar datos en lote desde archivos JSON. El sistema debe manejar correctamente la inclusión de múltiples registros, identificando y reportando errores o faltantes en los datos, como atributos obligatorios (venta\_id, fecha, producto\_id, producto\_nombre, categoria).

### **B. Procesamiento y Limpieza de Datos:**

- a. Los datos de ventas suelen contener errores, valores faltantes, o duplicados que deben ser corregidos para garantizar la precisión del análisis. El sistema debe ofrecer herramientas para:
  - i. Completar valores faltantes (como cantidad o precio\_unitario) utilizando diferentes métodos estadísticos.
  - ii. Eliminar registros duplicados que podrían distorsionar los resultados del análisis.

### **C. Análisis y Estadísticas de Ventas:**

- a. Se requiere obtener información estadística relevante a partir de los datos de ventas, como:
  - i. El total de ventas, tanto mensual como anual.
  - ii. Identificar los meses con mayor venta y días de la semana más activos.
  - iii. Calcular tendencias de crecimiento o decrecimiento de ventas.
- b. Además, se debe poder generar informes sobre las categorías de productos más vendidas.

### **D. Persistencia de Datos y Uso Continuo:**

- a. Dado que los datos de ventas pueden actualizarse y analizarse continuamente, es importante mantener un registro persistente de las ventas importadas y procesadas. El sistema debe garantizar que los datos procesados se almacenen de forma permanente en un archivo JSON adicional, permitiendo que los análisis se reanuden desde donde se dejaron al reiniciar el sistema.

## Diseño del programa

### A. Decisiones de diseño

- a. **Modularidad:** El programa está organizado en varios archivos de encabezado y fuentes (.h y .c), cada uno responsable de diferentes aspectos del sistema, como el manejo de archivos JSON (funcs\_json.h), la lógica de ventas (ventas.h), y el control de flujo principal (main.c).
- b. **Uso de Estructuras Dinámicas:** Para manejar los datos de ventas, se utilizan estructuras de datos dinámicas como listas vinculadas o arreglos dinámicos. Esto permite una gestión eficiente de la memoria y facilita la expansión de la lista de ventas a medida que se importan más datos.
- c. **Persistencia de Datos:** Se utiliza la librería cJSON para manejar los datos en formato JSON, permitiendo la importación y exportación de datos de manera estructurada. Al iniciar el sistema, se verifica la existencia de un archivo con datos previos, y al cerrar, se actualiza el archivo con la información más reciente, evitando duplicados.

### B. Algoritmos Usados

- a. **Importación de Datos:**
  - i. Se lee un archivo JSON proporcionado por el usuario.
  - ii. Los datos se cargan en memoria usando estructuras dinámicas.
  - iii. Se verifica la presencia de atributos obligatorios (venta\_id, fecha, producto\_id, etc.).
  - iv. Los registros incompletos son reportados y no se incluyen en la memoria.
- b. **Procesamiento de Datos:**
  - i. **Completar Datos Faltantes:** Utiliza técnicas estadísticas como la moda, media (promedio) o mediana para completar los campos faltantes de cantidad y precio\_unitario.
  - ii. **Eliminación de Duplicados:** Utiliza un algoritmo de comparación para detectar y eliminar registros duplicados basados en el venta\_id.
- c. **Análisis de Datos:**
  - i. **Total de Ventas:** Calcula la suma total de los importes de todas las ventas almacenadas en memoria.

- ii. **Análisis Temporal:** Calcula el mes con mayor venta total y el día de la semana más activo, así como la tasa de crecimiento o decrecimiento de las ventas en un trimestre específico, cuya fórmula utilizada fue:

*T: Total de ventas*

$$\frac{T_{Trimestre Actual} - T_{Trimestre Anterior}}{T_{Trimestre Anterior}} \times 100 = Tasa de crecimiento \%$$

- iii. **Estadísticas:** Determina el top 5 de categorías con mayores ventas aplicando Bubble sort descendente y devolver las primeras 5 categorías.
- d. **Persistencia de Datos:**
- i. Almacena los datos procesados en un archivo JSON de forma ordenada y evita la duplicación al sobrescribir el contenido previo.

## Librerías Usadas en el Proyecto

### A. cJSON:

- a. Propósito: Manipulación y análisis de datos en formato JSON.
- b. Funciones Principales:
  - i. Creación y parsing: Permite crear objetos JSON en memoria y analizarlos a partir de una cadena de texto.
  - ii. Acceso a datos: Proporciona funciones para acceder y modificar elementos dentro de un objeto JSON.
  - iii. Serialización: Convierte estructuras de datos en memoria a formato JSON (cadena de texto) para almacenarlas o transmitirlos.

### B. stdio.h:

- a. Propósito: Entrada y salida estándar en C.
- b. Funciones Principales:
  - i. fopen, fclose, fgetc, fputc: Manejo de archivos para leer y escribir datos.
  - ii. printf: Mostrar información al usuario y reportar errores.

### C. stdlib.h:

- a. Propósito: Proporcionar funciones generales de propósito variado.
- b. Funciones Principales:
  - i. malloc, free: Manejo dinámico de memoria.
  - ii. atoi, atof: Conversión de cadenas a enteros y flotantes.

### D. string.h:

- a. Propósito: Manipulación de cadenas de caracteres.
- b. Funciones Principales:
  - i. strlen, strcpy, strcat, strcmp: Manipulación de cadenas, como calcular longitud, copiar, concatenar y comparar.

### E. time.h:

- a. Propósito: Manejo y manipulación de fechas y horas.
- b. Funciones Principales:
  - i. strftime: Formateo de fechas y horas.
  - ii. mktime: Conversión de estructuras de tiempo a tiempo calendario.

## **Análisis de resultados**

### **Objetivos alcanzados**

#### **A. Importar Datos de Ventas**

- a. Permitir cargar un archivo JSON con datos de ventas al sistema. El usuario debe proporcionar la ruta del archivo al ser solicitado. El sistema verificará si el archivo contiene todos los atributos necesarios y reportará cualquier dato faltante.

#### **B. Procesamiento de datos**

- a. Permitir completar datos faltantes en cantidad y precio unitario. Se utilizan diferentes técnicas para los atributos como la moda, la media (promedio) o mediana.
- b. Eliminar datos duplicados.

#### **C. Análisis de datos**

- a. Total de ventas.
- b. Total de ventas mensuales y anuales.

#### **D. Análisis temporal**

- a. Permite ver el mes con mayor venta y el día de la semana más activo.
- b. Permite calcular la tasa de crecimiento o decrecimiento de las ventas en un trimestre específico.

#### **E. Estadística**

- a. Permite visualizar el top 5 de categorías con mayores ventas.

#### **F. Salir del Programa**

- a. Cierra el programa guardando todos los cambios realizados en un archivo JSON definido para mantener la persistencia de los datos procesados.

## Bitácora

git: [https://github.com/MITTuu/PP1\\_Est](https://github.com/MITTuu/PP1_Est)