

**Instituto Tecnológico de Costa Rica Escuela de
ingeniería en computación***IC4700* Lenguajes de
Programación
PY1 – Proyecto 01

Estudiante(s):

Gadyr Calderón Díaz

2022327328

Profesor(a) a cargo:

Allan Rodriguez Davila

II Semestre 2024

ÍNDICE

Estudiante(s):	1
Profesor(a) a cargo:	1
Manual de usuario	3
Pruebas de Funcionalidad.....	8
Descripción del problema.....	15
Diseño del programa.....	15
. 1. Decisiones de Diseño	15
Librerías Usadas	18
Análisis de Resultados	21
Bitácora.....	21

Manual de usuario

Datos Importantes

- Fue programado en Sistema Operativo LINUX.
- Debe tener dentro del directorio los archivos cJSON

Instrucciones de Compilación: Para compilar y enlazar este programa, se deberán usar el siguiente comando,

```
gadyr@Gadyr:~/Proyecto1-LDP/Proyecto1-LDP/CodigoFuente$ gcc main.c cJSON.c -o main -lm
```

Teniendo en cuenta que hacen un git clone de mi repositorio, ya que es necesario que contengan los archivos de cJSON.

Luego llamamos al main con,

```
gadyr@Gadyr:~/Proyecto1-LDP/Proyecto1-LDP/CodigoFuente$ ./main
```

Con esto ya el programa iniciaría mostrando el siguiente menú,

```
=== Menu Principal ===
1. Importacion de datos
2. Procesamiento de datos
3. Analisis de datos
4. Analisis temporal
5. Estadisticas
6. Salir
Seleccione una opcion:
```

Este es el menú principal, acá, antes de mostrarse este menú, se mostrará un mensaje si existe alguna información pre-existente en el directorio, si existe un JSON ya cargado anteriormente.

En caso de existir, se mostrará este mensaje,

```
Informacion del bloque JSON 1
Info JSON agregada con exito!
Informacion del bloque JSON 2
Info JSON agregada con exito!
Informacion del bloque JSON 3
Info JSON agregada con exito!
Informacion del bloque JSON 4
Info JSON agregada con exito!
Informacion del bloque JSON 5
Info JSON agregada con exito!
```

Esto según la cantidad de contenido que tenga este, en caso de que no haya o no exista. Se mostrar este mensaje,

No se encontró archivo de datos, comenzando con una lista vacía.

Prosiguiendo con el menú, en Importación de Datos, si entra en esta opción se le mostrará el siguiente mensaje,

```
=== Menu Principal ===
1. Importacion de datos
2. Procesamiento de datos
3. Analisis de datos
4. Analisis temporal
5. Estadisticas
6. Salir
Seleccione una opcion: 1
Put the PATH of the .JSON:
```

El cual hace referencia a que introduzca el PATH de la dirección del archivo .json que desea que se cargue,

```
4. Analisis temporal
5. Estadisticas
6. Salir
Seleccione una opcion: 1
Put the PATH of the .JSON:
/home/gadyr/Proyecto1-LDP/Proyecto1-LDP/filename.json
```

Luego de poner el PATH, se mostrarán mensajes de acuerdo a la cantidad de contenido que tenga el json,

```

Informacion del bloque JSON 1
Info JSON agregada con exito!
Informacion del bloque JSON 2
No se incluye el apartado de CANTIDAD!!
Info JSON agregada con exito!
Informacion del bloque JSON 3
Info JSON agregada con exito!
Informacion del bloque JSON 4
Info JSON agregada con exito!
Informacion del bloque JSON 5

```

Como se muestra acá, si en algún json hace falta un atributo, el programa mostrará que hace falta el mismo. Luego en el apartado 2,

```

=== Menu de Procesamiento de Datos ===
1. Completar datos faltantes con moda
2. Completar datos faltantes con mediana
3. Eliminar Datos Duplicados
4. Mostar Contenido
5. Regresar
Seleccione una opcion: 

```

Se desplegará un sub menú para las opciones de este apartado, acá si se desean completar los datos por moda,

```

Cantidad faltante encontrada en venta ID 2. Completando con la moda: 1

```

Se mostrará el mensaje que se completó por moda y el identificador de la venta que se completó. De igual manera con el de mediana,

```

Cantidad faltante encontrada en venta ID 2. Completando con la media: 0.800000

```

En el apartado de eliminar datos duplicados, este mostrará lo siguiente en caso de existir,

```

Duplicado encontrado y eliminado: Venta ID 1
Duplicado encontrado y eliminado: Venta ID 2
Duplicado encontrado y eliminado: Venta ID 3
Duplicado encontrado y eliminado: Venta ID 4
Duplicado encontrado y eliminado: Venta ID 5

```

Acá mostrará los duplicados que se encontraron y fueron eliminados.

Asimismo, en este apartado está la opción de ver el contenido que sería la opción 4,

```

=== Menu de Procesamiento de Datos ===
1. Completar datos faltantes con moda
2. Completar datos faltantes con mediana
3. Eliminar Datos Duplicados
4. Mostar Contenido
5. Regresar
Seleccione una opcion: █

```

```

Seleccione una opcion: 4
Venta ID: 1, Fecha:2023-01-04, ID de producto:109, Nombre de producto:Cámara, Categoria:Muebles, Cantidad:2, Precio unitario:250, Total: 500
Venta ID: 2, Fecha:2023-01-12, ID de producto:106, Nombre de producto:Teléfono, Categoria:Electrónica, Cantidad:1, Precio unitario:500, Total: 500
Venta ID: 3, Fecha:2023-04-12, ID de producto:107, Nombre de producto:Teléfono, Categoria:Ropa, Cantidad:1, Precio unitario:5000, Total: 5000
Venta ID: 4, Fecha:2023-04-12, ID de producto:107, Nombre de producto:Teléfono, Categoria:Tenis, Cantidad:1, Precio unitario:5000, Total: 5000
Venta ID: 5, Fecha:2023-04-12, ID de producto:107, Nombre de producto:Teléfono, Categoria:Carros, Cantidad:1, Precio unitario:5000, Total: 5000
=== Menu de Procesamiento de Datos ===

```

Luego, la opción de Regresar.

En la siguiente opción del menú tenemos el análisis de datos. Este presenta un menú con las opciones disponibles,

```

=== Menu de Analisis de Datos ===
1. Total de Ventas General
2. Total de Ventas Mensuales
3. Total de Ventas Anuales
4. Mostar Contenido
5. Regresar
Seleccione una opcion: █

```

El total de ventas muestra,

```
El valor total de todas las ventas es de: 16000
```

Asimismo, el total de ventas mensuales. Este pide el nombre del mes a buscar

```

Escriba el nombre del mes que desea filtrar:
Abril
El valor total de todas las ventas en el mes ABRIL, es de: 15000

```

Y en Anuales similar, pide el año a buscar,

```

Escriba el año que desea filtrar:
2023
El valor total de todas las ventas en el año de 2023, es de: 16000

```

Pasando al siguiente apartado, el Análisis Temporal, mostrando el siguiente menú,

```

=== Menu de Analisis de Datos ===
1. Mes con mayor venta
2. Dia mas activo
3. Calcular Crecimiento de Ventas
4. Regresar
Seleccione una opcion: 

```

Si se desea ver el mes con mayor venta, se presiona la opción 1,

```

4. Regresar
Seleccione una opcion: 1
El mes con mayor venta es ABRIL, con un total de 15000

```

El día más activo,

```

4. Regresar
Seleccione una opcion: 2
El día de la semana más activo es Miércoles con un total de 15500.00 ventas.

```

En el crecimiento de ventas, se muestra el siguiente mensaje pidiendo el año y el trimestre,

```

4. Regresar
Seleccione una opcion: 3
Ingrese el año: 2023
Ingrese el trimestre (1-4): 1
La tasa de crecimiento o decrecimiento en el trimestre 1 del año 2023 es: 100.00%

```

Y, por último, las estadísticas,

```

5. Salir
Seleccione una opcion: 5
Top 5 categorías con mayores ventas:
1. Ropa: 5000
2. Tenis: 5000
3. Carros: 5000
4. Muebles: 500
5. Electrónica: 500

```

Pruebas de Funcionalidad.

Importación de Datos:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

2. Procesamiento de datos
3. Analisis de datos
4. Analisis temporal
5. Estadísticas
6. Salir
Seleccione una opcion: 1
Put the PATH of the .JSON:
/home/gadynr/Proyecto1-LDP/Proyecto1-LDP/filename.json
Informacion del bloque JSON 1
Info JSON agregada con éxito!
Informacion del bloque JSON 2
No se incluye el apartado de CANTIDAD!!
Info JSON agregada con éxito!
Informacion del bloque JSON 3
Info JSON agregada con éxito!
Informacion del bloque JSON 4
Info JSON agregada con éxito!
Informacion del bloque JSON 5
Info JSON agregada con éxito!

=== Menu Principal ===
1. Importacion de datos
2. Procesamiento de datos
3. Analisis de datos
4. Analisis temporal
5. Estadísticas
6. Salir
Seleccione una opcion: █
```



```

C cJSON.c  C main.c  C ctype.h  filename.json X  .gitignore  C cJSON.h
{} filename.json > {} 1

1  [
2      {
3          "venta_id": 1,
4          "fecha": "2023-01-04",
5          "producto_id": 109,
6          "producto_nombre": "C\u00e1mara",
7          "categoria": "Muebles",
8          "cantidad": 2,
9          "precio_unitario": 250.0,
10         "total": 500.0
11     },
12     {
13         "venta_id": 2,
14         "fecha": "2023-01-12",
15         "producto_id": 106,
16         "producto_nombre": "Tel\u00e9fono",
17         "categoria": "Electr\u00f3nica",
18         "precio_unitario": 500.0,
19         "total": 500.0
20     },
21     {
22         "venta_id": 3,
23         "fecha": "2023-04-12",
24         "producto_id": 107,
25         "producto_nombre": "Tel\u00e9fono",
26         "categoria": "Ropa",
27         "cantidad": 1,
28         "precio_unitario": 5000.0,
29         "total": 5000.0
30     },
31 ]

```

Se cargan correctamente los datos del archivo .json.

Procesamiento de Datos:

Eliminación de duplicados.

```

=== Menu de Procesamiento de Datos ===
1. Completar datos faltantes con moda
2. Completar datos faltantes con mediana
3. Eliminar Datos Duplicados
4. Mostar Contenido
5. Regresar
Seleccione una opcion: 4
Venta ID: 1, Fecha:2023-01-04, ID de producto:109, Nombre de producto:Cámara, Categoria:Muebles, Cantidad:2, Precio unitario:250, Total: 500
Venta ID: 2, Fecha:2023-01-12, ID de producto:106, Nombre de producto:Teléfono, Categoria:Electrónica, Cantidad:1, Precio unitario:500, Total: 500
Venta ID: 3, Fecha:2023-04-12, ID de producto:107, Nombre de producto:Teléfono, Categoria:Ropa, Cantidad:1, Precio unitario:5000, Total: 5000
Venta ID: 4, Fecha:2023-04-12, ID de producto:107, Nombre de producto:Teléfono, Categoria:Tenis, Cantidad:1, Precio unitario:5000, Total: 5000
Venta ID: 5, Fecha:2023-04-12, ID de producto:107, Nombre de producto:Teléfono, Categoria:Carros, Cantidad:1, Precio unitario:5000, Total: 5000
Venta ID: 1, Fecha:2023-01-04, ID de producto:109, Nombre de producto:Cámara, Categoria:Muebles, Cantidad:2, Precio unitario:250, Total: 500
Venta ID: 2, Fecha:2023-01-12, ID de producto:106, Nombre de producto:Teléfono, Categoria:Electrónica, Cantidad:1, Precio unitario:500, Total: 500
Venta ID: 3, Fecha:2023-04-12, ID de producto:107, Nombre de producto:Teléfono, Categoria:Ropa, Cantidad:1, Precio unitario:5000, Total: 5000
Venta ID: 4, Fecha:2023-04-12, ID de producto:107, Nombre de producto:Teléfono, Categoria:Tenis, Cantidad:1, Precio unitario:5000, Total: 5000
Venta ID: 5, Fecha:2023-04-12, ID de producto:107, Nombre de producto:Teléfono, Categoria:Carros, Cantidad:1, Precio unitario:5000, Total: 5000

=== Menu de Procesamiento de Datos ===
1. Completar datos faltantes con moda
2. Completar datos faltantes con mediana
3. Eliminar Datos Duplicados
4. Mostar Contenido
5. Regresar
Seleccione una opcion:

```

```

4. Mostar Contenido
5. Regresar
Seleccione una opcion: 3
Duplicado encontrado y eliminado: Venta ID 1
Duplicado encontrado y eliminado: Venta ID 2
Duplicado encontrado y eliminado: Venta ID 3
Duplicado encontrado y eliminado: Venta ID 4
Duplicado encontrado y eliminado: Venta ID 5

=== Menu de Procesamiento de Datos ===
1. Completar datos faltantes con moda
2. Completar datos faltantes con mediana
3. Eliminar Datos Duplicados
4. Mostar Contenido
5. Regresar
Seleccione una opcion: 4
Venta ID: 1, Fecha:2023-01-04, ID de producto:109, Nombre de producto:Cámara, Categoria:Muebles, Cantidad:2, Precio unitario:250, Total: 500
Venta ID: 2, Fecha:2023-01-12, ID de producto:106, Nombre de producto:Teléfono, Categoria:Electrónica, Cantidad:1, Precio unitario:500, Total: 500
Venta ID: 3, Fecha:2023-04-12, ID de producto:107, Nombre de producto:Teléfono, Categoria:Ropa, Cantidad:1, Precio unitario:5000, Total: 5000
Venta ID: 4, Fecha:2023-04-12, ID de producto:107, Nombre de producto:Teléfono, Categoria:Tenis, Cantidad:1, Precio unitario:5000, Total: 5000
Venta ID: 5, Fecha:2023-04-12, ID de producto:107, Nombre de producto:Teléfono, Categoria:Carros, Cantidad:1, Precio unitario:5000, Total: 5000

=== Menu de Procesamiento de Datos ===
1. Completar datos faltantes con moda
2. Completar datos faltantes con mediana
3. Eliminar Datos Duplicados
4. Mostar Contenido
5. Regresar
Seleccione una opcion:

```

Se eliminan correctamente los datos duplicados.

Completar por Moda:

```

3. Eliminar Datos Duplicados
4. Mostar Contenido
5. Regresar
Seleccione una opcion: 1
Valor fuera de rango detectado: -1
Valor fuera de rango detectado: 5000
Valor fuera de rango detectado: 5000
Valor fuera de rango detectado: 5000
Cantidad faltante encontrada en venta ID 2. Completando con la moda: 1

=== Menu de Procesamiento de Datos ===
1. Completar datos faltantes con moda
2. Completar datos faltantes con mediana
3. Eliminar Datos Duplicados
4. Mostar Contenido
5. Regresar
Seleccione una opcion:

```

Se agrega correctamente por moda, como se podía ver arriba en el .json de prueba, la venta con ID 2 no tenia la cantidad y se le agregó con un 1.

Por mediana:

```

=== Menu de Procesamiento de Datos ===
1. Completar datos faltantes con moda
2. Completar datos faltantes con mediana
3. Eliminar Datos Duplicados
4. Mostar Contenido
5. Regresar
Seleccione una opcion: 2
Cantidad faltante encontrada en venta ID 2. Completando con la media: 0.800000

=== Menu de Procesamiento de Datos ===
1. Completar datos faltantes con moda
2. Completar datos faltantes con mediana
3. Eliminar Datos Duplicados
4. Mostar Contenido
5. Regresar
Seleccione una opcion: █

```

De igual manera, se le asigna un valor al cálculo de mediana.

Análisis de datos:

Total de la ventas.

```

=== Menu de Analisis de Datos ===
1. Total de Ventas General
2. Total de Ventas Mensuales
3. Total de Ventas Anuales
4. Mostar Contenido
5. Regresar
Seleccione una opcion: 1
El valor total de todas las ventas es de: 16000

=== Menu de Analisis de Datos ===
1. Total de Ventas General
2. Total de Ventas Mensuales
3. Total de Ventas Anuales
4. Mostar Contenido
5. Regresar
Seleccione una opcion: █

```

```
cliente > datos.json > ...
[{"cantidad": 2,
  "precio_unitario": 250,
  "total": 500
}, {"venta_id": 2,
  "fecha": "2023-01-12",
  "producto_id": 106,
  "producto_nombre": "Teléfono",
  "categoria": "Electrónica",
  "cantidad": -1,
  "precio_unitario": 500,
  "total": 500
}, {"venta_id": 3,
  "fecha": "2023-04-12",
  "producto_id": 107,
  "producto_nombre": "Teléfono",
  "categoria": "Ropa",
  "cantidad": 1,
  "precio_unitario": 5000,
  "total": 5000
}, {"venta_id": 4,
  "fecha": "2023-04-12",
  "producto_id": 107,
  "producto_nombre": "Teléfono",
  "categoria": "Tenis",
  "cantidad": 1,
  "precio_unitario": 5000,
  "total": 5000
}, {"venta_id": 5,
  "fecha": "2023-04-12",
  "producto_id": 107,
  "producto_nombre": "Teléfono",
  "categoria": "Carros",
  "cantidad": 1,
  "precio_unitario": 5000,
  "total": 5000
}]
```

Este es el json de prueba, acá se puede validar los valores del total.

De igual manera el de mensual y Anual

```
3. Total de Ventas Anuales
4. Mostar Contenido
5. Regresar
Seleccione una opcion: 2
Escriba el nombre del mes que desea filtrar:
Abril
El valor total de todas las ventas en el mes ABRIL, es de: 15000
```

```
=== Menu de Analisis de Datos ===
1. Total de Ventas General
2. Total de Ventas Mensuales
3. Total de Ventas Anuales
4. Mostar Contenido
5. Regresar
Seleccione una opcion: █
```

```
3. Total de Ventas Anuales
4. Mostar Contenido
5. Regresar
Seleccione una opcion: 3
Escriba el año que desea filtrar:
2023
El valor total de todas las ventas en el año de 2023, es de: 16000
```

```
=== Menu de Analisis de Datos ===
1. Total de Ventas General
2. Total de Ventas Mensuales
3. Total de Ventas Anuales
4. Mostar Contenido
5. Regresar
Seleccione una opcion: █
```

Análisis Temporal.

El análisis temporal en el mes más activo

```
=== Menu de Analisis de Datos ===
1. Mes con mayor venta
2. Dia mas activo
3. Calcular Crecimiento de Ventas
4. Regresar
Seleccione una opcion: 1
El mes con mayor venta es ABRIL, con un total de 15000
```

```
=== Menu de Analisis de Datos ===
1. Mes con mayor venta
2. Dia mas activo
3. Calcular Crecimiento de Ventas
4. Regresar
Seleccione una opcion: █
```

Igual en el día

```

=== Menu de Analisis de Datos ===
1. Mes con mayor venta
2. Dia mas activo
3. Calcular Crecimiento de Ventas
4. Regresar
Seleccione una opcion: 2
El día de la semana más activo es Miércoles con un total de 15500.00 ventas.

```

Igualmente, con la tasa de crecimiento

```

1. Mes con mayor venta
2. Dia mas activo
3. Calcular Crecimiento de Ventas
4. Regresar
Seleccione una opcion: 3
Ingrese el año: 2023
Ingrese el trimestre (1-4): 1
La tasa de crecimiento o decrecimiento en el trimestre 1 del año 2023 es: 100.00%

=== Menu de Analisis de Datos ===
1. Mes con mayor venta
2. Dia mas activo
3. Calcular Crecimiento de Ventas
4. Regresar
Seleccione una opcion: 

```

Estadísticas.

```

5. Estadísticas
6. Salir
Seleccione una opcion: 5
Top 5 categorías con mayores ventas:
1. Ropa: 5000
2. Tenis: 5000
3. Carros: 5000
4. Muebles: 500
5. Electrónica: 500

```

Descripción del problema

Hoy en día, la habilidad para analizar y extraer información relevante de grandes volúmenes de datos es esencial para la toma de decisiones en las empresas. En el entorno empresarial, uno de los aspectos más críticos es la gestión y análisis de los datos de ventas, ya que estos ofrecen información valiosa sobre el rendimiento de los productos, las tendencias del mercado y las preferencias de los consumidores. El "Sistema de Análisis de Datos de Ventas" se presenta como una solución diseñada para obtener insights clave a partir de los datos generados por las transacciones diarias de ventas. Este sistema permitirá a los usuarios visualizar, analizar y comprender los patrones de ventas, facilitando así la toma de decisiones estratégicas que pueden mejorar tanto la rentabilidad como la eficiencia operativa.

Diseño del programa

. 1. Decisiones de Diseño

1. Estructura Modular:

- **Importación de Datos:** Carga de datos desde un archivo JSON y almacenamiento en una lista de ventas.
- **Procesamiento de Datos:** Limpieza y corrección de datos (completar datos faltantes y eliminar duplicados).
- **Análisis de Datos:** Cálculo de estadísticas generales como el total de ventas y ventas mensuales/anuales.
- **Análisis Temporal:** Cálculo de métricas temporales como el mes con mayor venta y el día de la semana más activo.
- **Estadísticas:** Cálculo y visualización de estadísticas como el Top 5 de categorías con mayores ventas.
- **Persistencia de Datos:** Guardar los datos procesados en un archivo JSON para persistencia entre ejecuciones.

2. Algoritmos Utilizados:

- *Cargar Datos JSON:*
 - Leer un archivo JSON y convertirlo a estructuras en C utilizando la librería cJSON o json-c.
- *Procesamiento de Datos:*
 - **Completar Datos Faltantes:** Utilizar la moda, media o mediana para rellenar valores faltantes.
 - **Eliminar Duplicados:** Usar un algoritmo de comparación para identificar y eliminar registros duplicados.
- *Análisis de Datos:*
 - **Total de Ventas:** Sumar el monto total de todas las ventas.
 - **Ventas Mensuales/Anuales:** Agrupar y sumar ventas por mes y año.
- *Análisis Temporal:*
 - **Mes con Mayor Venta:** Agrupar ventas por mes y calcular el total.
 - **Día de la Semana Más Activo:** Contar transacciones por día de la semana.
- *Estadísticas:*
 - **Top 5 Categorías:** Acumular ventas por categoría y ordenar en orden descendente.

3. Diagrama de Archivos

Aquí está el diagrama de archivos que describe cómo se organizan los archivos en el

CódigoFuente/

cJSON.c

cJSON.h

main.c

.gitignore

Readme.md

2. Algoritmos Usados

- *Importación de Datos:*
 - Utiliza la librería cJSON para analizar el archivo JSON.
 - Itera sobre el array JSON para leer y almacenar cada venta en una estructura JSON INFO
- *Procesamiento de Datos:*
 - *Completar Datos Faltantes:*
 - Calcular la moda, media o mediana de los datos disponibles y reemplazar los valores faltantes con estos cálculos.
 - *Eliminar Duplicados:*
 - Iterar sobre la lista de ventas y comparar cada venta con las demás para identificar y eliminar duplicados.
- *Análisis de Datos:*
 - *Total de Ventas:*
 - Iterar sobre todas las ventas y sumar los totales.
 - *Ventas Mensuales/Anuales:*
 - Agrupar ventas por mes y año utilizando estructuras de datos.

- *Análisis Temporal:*
 - *Mes con Mayor Venta:*
 - Agrupar ventas por mes y calcular el total para cada mes.
 - *Día de la Semana Más Activo:*
 - Contar las transacciones por día de la semana utilizando una estructura de array.
- *Estadísticas:*
 - *Top 5 Categorías:*
 - Acumular ventas por categoría en una estructura de datos (como un array o lista) y ordenar los resultados.

Librerías Usadas

1. Librerías para Manejo de Archivos y JSON

- *cJSON:*
 - **Propósito:** Procesar y analizar archivos JSON en C.
 - *Funciones Clave:*
 - `cJSON_Parse()`: Analiza una cadena JSON y devuelve una estructura cJSON.
 - `cJSON_GetObjectItem()`: Obtiene un objeto JSON a partir de una clave.
 - `cJSON_AddItemToArray()`: Añade un ítem a un array JSON.
 - `cJSON_Print()`: Imprime la representación JSON de una estructura cJSON.
 - **Uso:** Utilizada para leer datos de un archivo JSON y convertirlos en

estructuras C, y para convertir estructuras C en JSON antes de guardar los datos.

- **json-c** (si se prefiere en lugar de cJSON):
 - **Propósito:** Procesar y analizar archivos JSON en C.
 - *Funciones Clave:*
 - `json_object_from_file()`: Lee un archivo JSON y devuelve un objeto JSON.
 - `json_object_to_json_string()`: Convierte un objeto JSON a una cadena JSON.

2. Librerías para Manejo de Fechas y Tiempos

- *time.h*:
 - **Propósito:** Manipular y obtener información sobre fechas y tiempos.
 - *Funciones Clave:*
 - `mktime()`: Convierte una estructura `tm` a un valor `time_t`.
 - `localtime()`: Convierte un valor `time_t` a una estructura `tm` que representa el tiempo local.
 - **Uso:** Utilizada para convertir fechas en formato de cadena a estructuras de tiempo y para calcular métricas temporales.

3. Librerías para Procesamiento de Datos

- *stdlib.h*:
 - **Propósito:** Proveer funciones para la gestión de memoria y procesamiento de datos.

- *Funciones Clave:*
 - malloc(), free(): Gestión de memoria dinámica.
 - qsort(): Ordenamiento de arrays.
- **Uso:** Utilizada para gestionar la memoria dinámica para las estructuras de datos y ordenar listas de categorías por monto total.
- *stdio.h:*
 - **Propósito:** Proveer funciones de entrada y salida estándar.
 - *Funciones Clave:*
 - printf(), scanf(): Operaciones básicas de entrada y salida.
 - fopen(), fclose(), fread(), fwrite(): Operaciones de archivo.
 - **Uso:** Utilizada para la entrada/salida de datos y para manejar archivos.
- **Math.h**
 - Se utilizó para realizar round() a la hora de calcular datos ventas totales, meses, etc..
- **Time.h**
 - Se utilizó para sacar los días en el calculo de días más activos

Análisis de Resultados

Objetivo	Alcanzado	Razón
Implementación de la estructura de datos y manejo de JSON	Sí	-
Implementación de funciones de procesamiento y análisis	Sí	-
Exportación de resultados a PDF	No	Errores al importarse las librerías, además de la instalación.
Interfaz de usuario y menú	Sí	-
Persistencia de datos	Sí	-
Integración completa con todos los datos del archivo JSON	Sí	-
Pruebas exhaustivas con datos variados	Sí	-

Bitácora

<https://github.com/gadyrcdz/Proyecto1-LDP>