

# Entregable Sprint 1

## Requisitos

1. Tema, problema y solución
2. Dataset de referencia: fuente, definición, estructura, tipos y escala
3. Información, pasos, pseudocódigo, diagrama del programa
4. Sugerencias y mejoras aplicadas con copilot
5. programa (.py) que lea y muestre la información de la documentación
6. Archivo (.md) mostrando las instrucciones empleadas por Copilot que sirvieron como apoyo

**NOTA :** en este primer sprint los requisitos apuntan a desarrollar la documentación por lo que los requisitos del 1 al 3 se encuentran consignados en este mismo fichero `Documentacion.md` , el requisito No. 5 se atiende con el fichero python llamado `leer_markdown.py` . Finalmente los requisitos 4 y 6 se atienden con `Instrucciones.md` , para relacionar el contenido de los entregables con los requisitos de entrega se emplea la notación (S<Número del Sprint> : req<Número del Sprint>) en titulares que usen el formato markdown tanto en la documentación como en notebooks.

## Identificación del tema, problema y solución (S1 : req1)

¿Qué factores influyen en el volumen de ventas y cómo pueden optimizarse las estrategias comerciales para aumentar las compras de los productos con menor rotación?

### Problemas hallados en los datos:

- Las categorías están mal asignadas

## Dataset de referencia (S1 : req2)

### Estructura

la información está dividida en 4 tablas o entidades conformadas de la siguiente manera  
NOMBRE DE TABLA(atributos)

- CLIENTES(id\_cliente, nombre\_cliente, email, ciudad, fecha\_alta)
- PRODUCTOS (id\_producto, nombre\_producto, categoria, precio\_unitario)
- DETALLE\_VENTAS (id\_venta, id\_producto, nombre\_producto, cantidad, precio\_unitario, importe)
- VENTAS (id\_venta, fecha, id\_cliente, nombre\_cliente, email, medio\_pago)

## Tipos

Según su estructura los datos suministrados son de tipo **estructurado**.

Según su origen asumimos que los datos los suministra algún tipo de sistema POS, por tanto, son **secundarios**.

## CLIENTES

id_cliente	nombre_cliente	email	ciudad	fecha_alta
Cualitativo	Cualitativo	Cualitativo	Cualitativo	Cualitativo

---

## VENTAS

id_venta	fecha	id_cliente	nombre_cliente	email	medio_pago
Cualitativo	Cuantitativo (fecha)	Cualitativo	Cualitativo	Cualitativo	Cualitativo

---

## PRODUCTOS

id_producto	nombre_producto	categoria	precio_unitario
Cualitativo	Cualitativo	Cualitativo	Cuantitativo

---

## DETALLE\_VENTAS

id_venta	id_producto	nombre_producto	cantidad	precio_unitario	importe
Cualitativo	Cualitativo	Cualitativo	Cuantitativo	Cuantitativo	Cuantitativo

---

# Información, pasos, pseudocódigo y diagrama del programa (S1 : req3)

## Información general

El siguiente programa en Python permite leer un archivo Markdown ( README.md ), identificar los capítulos principales (niveles de encabezado ## ) y mostrar su contenido de manera interactiva en consola.

El usuario puede elegir qué capítulo desea visualizar y salir del programa cuando lo desee.

## Pasos generales del programa

1. Mostrar mensaje de bienvenida.
2. Leer el archivo README.md línea por línea.
3. Detectar los capítulos a partir de los encabezados ## .
4. Guardar los títulos y su contenido en un diccionario.
5. Mostrar un menú con los capítulos disponibles.
6. Permitir al usuario seleccionar un capítulo para ver su contenido.
7. Repetir el proceso hasta que el usuario elija la opción de salir.
8. Mostrar mensaje de despedida.

---

## Pseudocódigo del programa

INICIO DEL PROGRAMA

MOSTRAR "👋 Bienvenido al lector de capítulos del archivo README.md 📄"

DEFINIR ruta ← "README.md"

INTENTAR

    lineas ← LEER\_ARCHIVO(ruta)

```
capitulos ← OBTENER_CAPITULOS(lineas)
```

```
SI capitulos ESTÁ VACÍO ENTONCES
```

```
    MOSTRAR "⚠ No se encontraron capítulos (##) en el archivo."
```

```
    FINALIZAR PROGRAMA
```

```
FIN SI
```

```
REPETIR
```

```
    MOSTRAR_MENU(capitulos)
```

```
    MOSTRAR "Seleccione un capítulo por número: "
```

```
    LEER opcion
```

```
    SI opcion NO ES NÚMERO ENTONCES
```

```
        MOSTRAR "❌ Por favor ingrese un número válido."
```

```
        CONTINUAR CICLO
```

```
    FIN SI
```

```
    SI opcion = (NÚMERO DE CAPÍTULOS + 1) ENTONCES
```

```
        MOSTRAR "👋 Gracias por usar el lector de Markdown. ¡Hasta pronto!"
```

```
        SALIR DEL CICLO
```

```
    SINO SI opcion ENTRE 1 Y NÚMERO DE CAPÍTULOS ENTONCES
```

```
        titulo ← CAPÍTULO EN POSICIÓN (opcion - 1)
```

```
        MOSTRAR "--- " + titulo + " ---"
```

MOSTRAR capitulos[titulo]

MOSTRAR "-----"

SINO

MOSTRAR "⚠ Opción fuera de rango."

FIN SI

HASTA QUE opcion SEA IGUAL A (NÚMERO DE CAPÍTULOS + 1)

CAPTURAR EXCEPCIÓN FileNotFoundError

MOSTRAR "❌ No se encontró el archivo README.md en el directorio actual."

FIN DEL PROGRAMA

FUNCIÓN LEER\_ARCHIVO(ruta)

ABRIR archivo EN modo lectura CON codificación UTF-8

LEER todas las líneas EN una lista

CERRAR archivo

RETORNAR lista de líneas

FIN FUNCIÓN

FUNCIÓN OBTENER\_CAPITULOS(lineas)

CREAR diccionario vacío capitulos

titulo\_actual ← NULO

contenido\_actual ← lista vacía

PARA CADA linea EN lineas HACER

SI linea EMPIEZA CON "## " ENTONCES

SI titulo\_actual NO ES NULO ENTONCES

capitulos[titulo\_actual] ← UNIR contenido\_actual EN TEXTO

contenido\_actual ← lista vacía

FIN SI

titulo\_actual ← linea SIN "## " Y SIN ESPACIOS

SINO SI titulo\_actual NO ES NULO ENTONCES

AGREGAR linea A contenido\_actual

FIN SI

FIN PARA

SI titulo\_actual NO ES NULO ENTONCES

capitulos[titulo\_actual] ← UNIR contenido\_actual EN TEXTO

FIN SI

RETORNAR capitulos

FIN FUNCIÓN

FUNCIÓN MOSTRAR\_MENU(capitulos)

    MOSTRAR "Capítulos disponibles:"

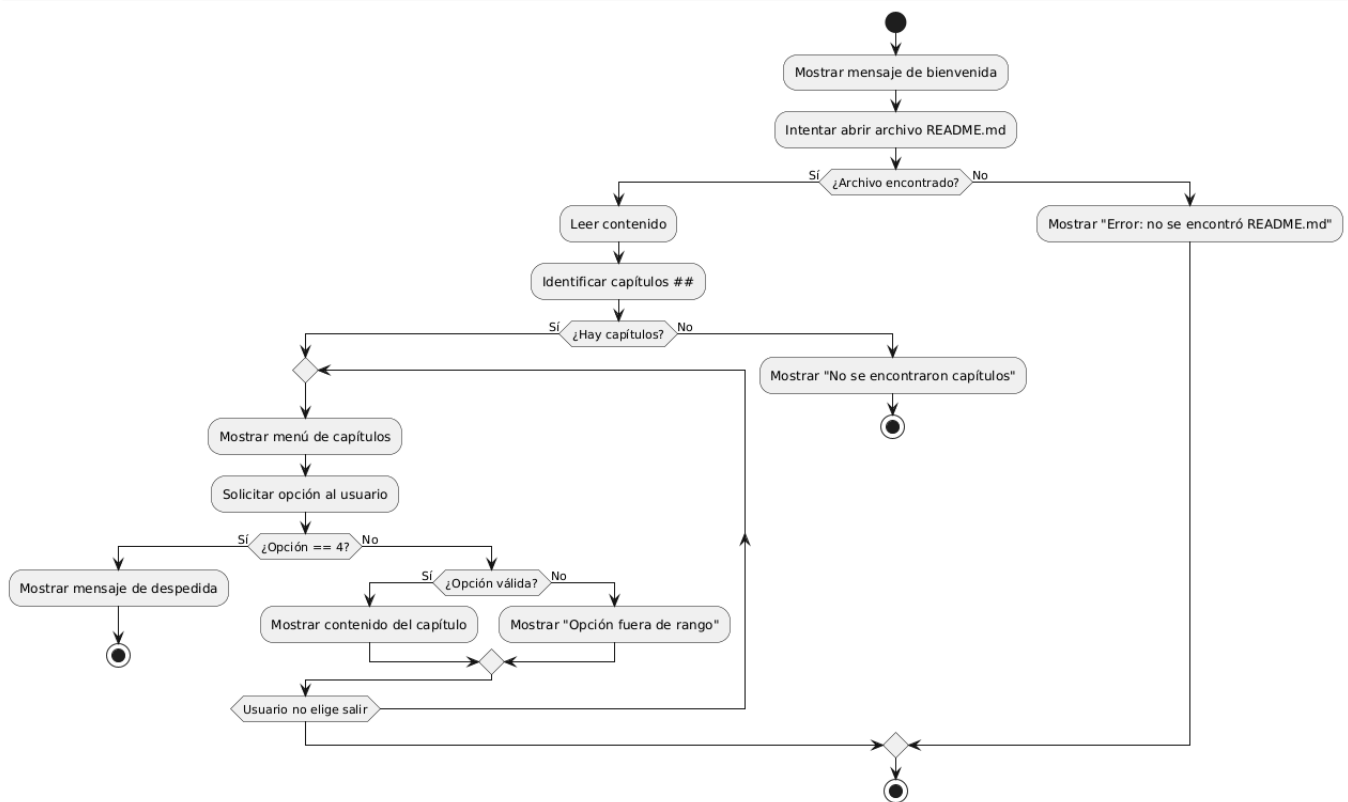
    PARA i DESDE 1 HASTA NÚMERO DE CAPÍTULOS HACER

        MOSTRAR i + ". " + NOMBRE\_DEL\_CAPITULO

    FIN PARA

    MOSTRAR (NÚMERO DE CAPÍTULOS + 1) + ". Salir"

FIN FUNCIÓN



## Entregable Sprint 2

### Requisitos

1. Estadísticas descriptivas básicas calculadas
2. Identificación del tipo de distribución de variables
3. Análisis de correlaciones entre variables principales
4. Detección de Outliers
5. Gráficos representativos (Mínimo 3)
6. Interpretación de resultados orientada al problema

7. Base de datos (.csv) lista y limpia para análisis
8. Información actualizada del proyecto

## Entregable Sprint 3

### Requisitos

1. Objetivo predecir o clasificar
2. Algoritmo elegido y justificación
3. Entradas (X) y salida (y)
4. Métricas de evaluación
5. Modelo ML implementado
6. División train/test y entrenamiento
7. Predicciones y métricas calculadas
8. Resultados en uno o más gráficos

## Entregable Sprint 4

### Requisitos

1. Entrega de dashboard en Power BI

## Especificaciones y notas del desarrollador

El formato original de los datos consistía en archivos extensión .csv usándose así durante 3/4 sprints para su facilidad de manejo en el entrenamiento de modelos y Análisis Exploratorios con python.

Para el sprint 4 se solicitó la creación de dashboard en Power BI por lo que en aras de aportar escalabilidad al ejercicio y aproximándolo a una ejecución bastante cercana a lo que se hace en un proyecto real, los datos fueron migrados a una base de datos PostgreSQL en nube mediante servicio gratuito de aiven (<https://aiven.io/>) posterior a un debido proceso de normalización y modelamiento de datos, puede verse como se realizó la migración mediante `datapipeline.ipynb`. Se utilizó una fuente de datos externas de ciudades de Argentina para poder crear la entidad correspondiente, la fuente de los datos corresponde a <https://simplemaps.com/data/ar-cities>.

El campo de importe se determinó como una medida o atributo calculado por lo cual fue excluido de la base de datos y se implementó como medida mediante Power BI optimizando uso de memoria, la conexión a la base de datos se realizó mediante conector ODBC el cuál puede ser descargado e instalado desde el sitio oficial de PostgreSQL, esto para permitir conexiones SSL ya que los servicios en nube para bases de datos lo exigen.



Se incluyó realización de identidad visual a la marca ficticia de tienda Aurelion para ofrecer no solo un producto funcional si no también estético que reflejara profesionalidad en la presentación del informe visual de Power BI.