**📁 Estructura del Proyecto**

📂 Reporter

├── 📄 runme.py # Archivo principal que ejecuta todo

├── 📄 graphs.py # Genera gráficos y el reporte en PDF

├── 📄 leer\_config.py # Lee la configuración del proyecto

├── 📄 installer.cmd # Instala los módulos necesarios

├── 📂 data/ # Carpeta donde se almacena el archivo CSV

├── 📂 output/ # Carpeta donde se guardará el reporte PDF

├── 📄 config.json # Configuración del proyecto

├── 📄 README.md # Documentación del proyecto

**⚙️ Configuración (config.json)**

Puedes modificar el archivo config.json para personalizar el funcionamiento del script.

Json

|  |
| --- |
| {      "input\_csv": "C:/Users/…/File.csv",      "output\_folder": " C:/Users/…/Reports/ ",      "output\_tittle": "Reporte de Datos",      "test\_mode": 1,      "top": 10,      "campo\_fechas": "Creation Date",      "author": "Jhonattan Garcia"    } |

**🔹 Explicación de las Opciones**

| **Parámetro** | **Descripción** |
| --- | --- |
| input\_csv | Ruta del archivo CSV a analizar. |
| output\_folder | Carpeta donde se guardará el PDF. |
| output\_tittle | Nombre base del archivo PDF. |
| campo\_fechas | Columna en el CSV que contiene las fechas. |
| test\_mode | 1 para ver mensajes en consola, 0 para ocultarlos. |
| top | Cantidad de valores a mostrar en los análisis. |
| author | Nombre del autor del reporte. |

**▶️ Cómo Ejecutar**

1. Asegúrate de haber configurado config.json correctamente.
2. Guarda el archivo CSV dentro de data/.
3. Ejecuta el script principal:

sh

python runme.py

1. ¡El **PDF se generará automáticamente** en output/! 🎉

**📜 Explicación de las Clases y Métodos**

**🟢 MainApp (📄 runme.py)**

Clase principal que maneja la lectura de datos y análisis.

| **Método** | **Descripción** | **Opciones** |
| --- | --- | --- |
| leer\_archivo() | Lee el archivo CSV y cuenta los registros. | - |
| obtener\_datos() | Retorna los datos del CSV en memoria. | - |
| obtener\_fechas() | Encuentra la fecha más reciente y la más antigua en el CSV. | - |
| leer\_proactividad() | Calcula el porcentaje de valores "True" en la columna "Proactive". | - |
| llenar\_tablas(campo, orden, campo\_filtro, valor\_filtro) | Obtiene los valores más altos o bajos de un campo. | orden=1 (mayores), orden=0 (menores) |
| llamar\_a\_graphs() | Llama a graphs.py para generar el PDF. | - |

**🔵 ReportePDF (📄 graphs.py)**

Genera el informe en PDF con gráficos.

| **Método** | **Descripción** | **Opciones** |
| --- | --- | --- |
| asignar\_proactividad(proactividad) | Guarda el porcentaje de proactividad. | - |
| asignar\_fechas(fecha\_mayor, fecha\_menor) | Guarda la fecha más alta y más baja del CSV. | - |
| agregar\_valor(datos, tipo\_grafico, titulo) | Agrega datos para graficar. | tipo\_grafico="barras" o "torta" |
| imprimir() | Genera y guarda el reporte en PDF. | - |

**📊 Ejemplo de Salida en Consola (test\_mode=1)**

lua

📂 Se ha abierto el archivo 'data/dataset.csv'

📊 Se han encontrado 1500 registros, para 300 clientes.

📆 Fechas encontradas: Mayor - 2025-02-18, Menor - 2025-02-08

📈 Porcentaje de valores 'True' en 'Proactive': 76.5%

✅ Se ha agregado la variable 'IPs con más intentos de Phishing' con 3 registros.

✅ Se ha agregado la variable 'Clientes con más actividad' con 3 registros.

✅ Se ha agregado la variable 'Ataques menos frecuentes' con 3 registros.

📄 Reporte generado: output/Analisis February 18, 2025.pdf

**📜 Ejemplo de Contenido en el PDF**

less

📌 Reporte de gestión del día 2025-02-18 evaluado desde el 2025-02-08, por Tu Nombre

📈 Nivel de Proactividad: 76.5%

📊 IPs con más intentos de Phishing

[Gráfico de barras]

📊 Clientes con más actividad

[Gráfico de torta]

📊 Ataques menos frecuentes

[Gráfico de barras]

**🛠 Posibles Errores y Soluciones**

| **Error** | **Solución** |
| --- | --- |
| FileNotFoundError | Verifica que input\_csv esté correctamente configurado. |
| UnicodeEncodeError | Usa latin-1 en graphs.py para evitar caracteres inválidos. |
| AttributeError: 'ReportePDF' has no attribute 'imprimir' | Asegúrate de que imprimir() está bien definido en graphs.py. |

**🎯 Conclusión**

Este proyecto permite **automatizar la creación de reportes con análisis de datos y gráficos** en **PDF**.  
Es fácil de configurar y no requiere conocimientos técnicos avanzados. 🚀📊

👨‍💻 **Autor:** [Tu Nombre]  
📅 **Última actualización:** Febrero 2025

¡Espero que este documento te ayude a entender y ejecutar el proyecto sin problemas! 😊🎉

yaml

---

## \*\*✅ Beneficios del README\*\*

✔️ \*\*Explica claramente cómo funciona el proyecto.\*\*

✔️ \*\*Detalla cada clase y método con opciones.\*\*

✔️ \*\*Muestra ejemplos de ejecución y errores comunes.\*\*

✔️ \*\*Facilita la ejecución para personas sin formación técnica.\*\*

Prueba ahora ejecutando:

```sh

python runme.py