

Instructivo para la Instalación y actualización del prototipo

Versión de la herramienta: 1.0.

Fecha de publicación: Mayo de 2024.

Objetivo:

En este manual se presenta una guía para el personal técnico de ElectroDunas con el fin de apoyarlos en la instalación y actualización del prototipo.

1. Descripción del Artefacto

La solución analítica desarrollada como parte de la iniciativa para satisfacer las necesidades de ElectroDunas se implementó exitosamente, logrando cumplir con los objetivos mediante un artefacto que permite:

- Visualizar datos históricos de las variables para cada cliente.
- Desplegar un resumen descriptivo por cliente de su comportamiento histórico.
- Identificar consumos anómalos que no coincidan con los consumos esperados dado el comportamiento histórico.
- Mostrarle al usuario cuando un cliente presente un comportamiento anómalo y la criticidad de éste de acuerdo con los modelos utilizados
- Ser amigable con el usuario, con una interfaz clara y comprensible.

Nota: Para mayor detalle de los atributos del Artefacto remitirse a la tabla de requerimientos. [enlace](#)

1.1 Esquema general de la solución:

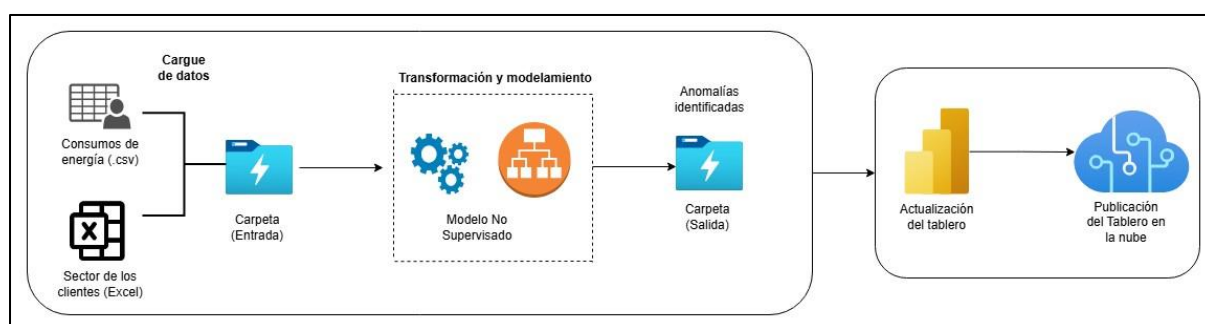


Figura 0. Estructura general del Artefacto

Este esquema presenta una solución integral para la identificación de anomalías en los consumos de energía utilizando modelos de aprendizaje no supervisado. El proceso comienza con la carga de datos provenientes de archivos CSV de consumos de energía y un archivo de Excel con el sector de los clientes, los cuales se almacenan en una carpeta de entrada. Estos datos se transforman y se preparan adecuadamente para ser analizados por un modelo no supervisado que identifica patrones y anomalías sin necesidad de etiquetas predefinidas.

Una vez procesados los datos, las anomalías identificadas se guardan en una carpeta de salida específica. Estos resultados se utilizan para actualizar un tablero de control, que permite visualizar y analizar de manera intuitiva las irregularidades encontradas. Este tablero,

implementado en la herramienta de Power BI, se actualiza automáticamente con los nuevos datos y las anomalías detectadas, facilitando el monitoreo continuo de los consumos de energía.

Finalmente, el tablero actualizado se publica en la nube, permitiendo el acceso remoto a los resultados desde cualquier ubicación con conexión a internet. Esta publicación asegura que los usuarios puedan consultar y analizar la información en tiempo real, mejorando la toma de decisiones y la detección temprana de posibles problemas en el consumo de energía. Este flujo de trabajo proporciona una solución eficiente y adaptable para la gestión y análisis de datos energéticos, optimizando el proceso de identificación de anomalías.

A lo largo de este instructivo se explica paso a paso la instalación de la herramienta y su debida actualización con datos nuevos.

2. Instalación del Artefacto

Requisitos Previos

- Conocimientos básicos de Python y manejo de entornos virtuales.
- Familiaridad con la instalación de software en Windows.
- Conocimientos básicos de Power BI.

Ventajas

- Automatización: Simplifica el proceso de análisis y predicción de datos.
- Accesibilidad: Fácil de usar a través de un acceso directo en el escritorio.
- Escalabilidad: Permite manejar grandes volúmenes de datos y ajustar el modelo de identificación de anomalías según sea necesario.

Limitaciones

- Dependencia de Anaconda: Requiere la instalación de Anaconda para gestionar el entorno de Python.
- Dependencia de Power BI: Requiere la instalación de Power BI.
- Compatibilidad: Puede tener limitaciones en diferentes sistemas operativos no especificados (principalmente diseñado para Windows).

Advertencias

- Requisitos del Sistema: Asegúrese de tener suficiente memoria RAM (4GB mínimo) y espacio en disco para manejar los datos y modelos.
- Actualizaciones: Los scripts y modelos pueden necesitar actualizaciones periódicas para mantener la precisión y eficiencia.

2.1 Acciones para la instalación y puesta en marcha del aplicativo que corre los script de python

La instalación del artefacto inicia con el siguiente paso a paso:

2.1.1 Descarga e Instalación de Anaconda

1. Descargar Anaconda: Ir a la página oficial de Anaconda y descargar la versión adecuada para su sistema operativo. [Enlace](#)

2. Instalar Anaconda: Seguir las instrucciones de instalación proporcionadas por Anaconda. Asegúrese de seleccionar la opción de agregar Anaconda al PATH del sistema.

2.1.2 Creación del Entorno Virtual

1. Abrir Anaconda Prompt: Buscar "Anaconda Prompt" en el menú de inicio y abrirlo.
2. Crear el Entorno: Ejecutar el siguiente comando:

```
conda create -n ElectroDunas python=3.8
```

3. Activar el Entorno:

```
conda activate ElectroDunas
```

2.1.3 Instalación de Dependencias

1. Navegar al Directorio del Proyecto y abrir la carpeta "App": Asegúrese de estar en el directorio donde se encuentran los siguientes archivos:

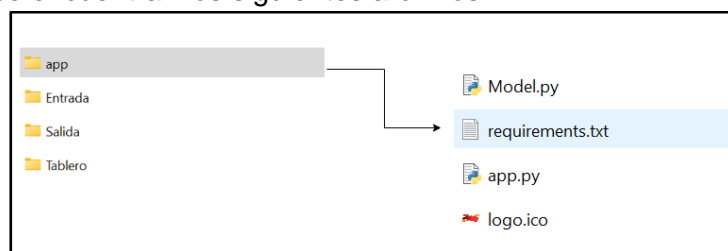


Figura 1. Ubicación del archivo requirements.txt

Nota: para navegar en la consola de "Anaconda Prompt" hasta el directorio del proyecto copia la ruta donde se haya guardado y utiliza el siguiente comando:

```
cd direccion_proyecto
```

2. Instalar Paquetes Requeridos: ejecutar los siguientes comandos

```
pip install -r requirements.txt
```

Nota: El archivo requirements.txt contiene todas las librerías necesarias para correr con satisfacción los scripts de python para el Artefacto.

2.1.4 Creación del Ejecutable

1. Crear el Ejecutable: ejecutar los siguientes comandos

```
pyinstaller --onefile --icon=logo.ico app.py
```

Esto generará un ejecutable en la carpeta 'dist', además de otros archivos, como se puede ver en la siguiente imagen:

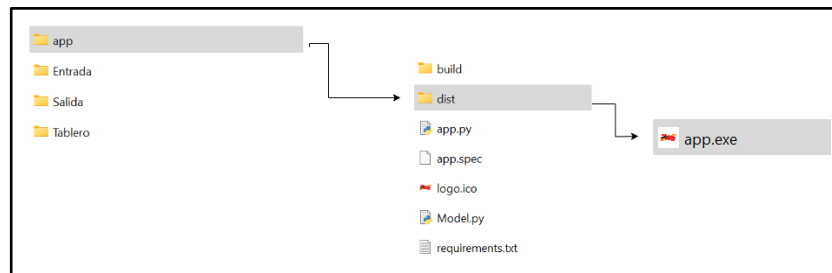


Figura 2. Ubicación del ejecutable *app.exe*

Antes de pasar al siguiente paso se debe mover a la carpeta 'dist' el archivo 'Model.py' para evitar conflictos al momento de correr la app, quedando de la siguiente manera:

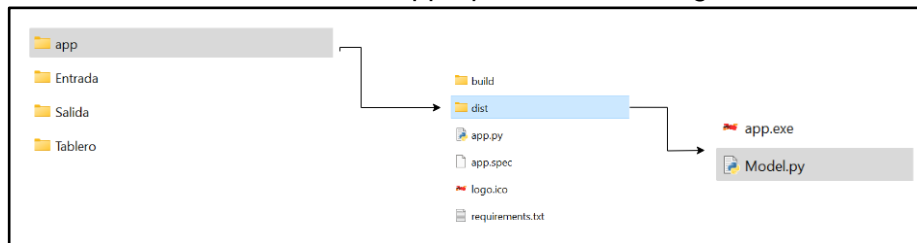


Figura 3. Ubicación del ejecutable *app.exe* y el Script *Model.py*

2.1.5 Creación del Acceso Directo

1. Crear Acceso Directo: En el directorio 'dist', encontrar el archivo ejecutable 'app.exe'. Hacer clic derecho sobre él y seleccionar "Enviar a > Escritorio (crear acceso directo)".



Figura 4. Icono del aplicativo creado en el escritorio como un acceso directo

Ejecutando todos los pasos anteriores se estaría listo para ejecutar el código sin problemas y de una manera sencilla sin necesidad de abrir interfaces de código.

2.2 Detalle de los Script de Python

En este punto se procede a aclarar el código que se encuentra corriendo de manera automática cuando se despliega el aplicativo, dicho código está compuesto por dos script de python que son:

app.py: es el script principal que se ejecuta a través de un acceso directo en el escritorio. Este script crea una interfaz gráfica usando Kivy que permite al usuario ejecutar el script Model.py con un botón. Los componentes principales son:

1. Importaciones: Se importan las bibliotecas necesarias para construir la interfaz gráfica y ejecutar el script externo. Esto incluye 'kivy.app' para crear la aplicación, 'kivy.ui.button' para crear botones, 'kivy.ui.boxlayout' para organizar los widgets en un layout, 'kivy.ui.label' para mostrar texto, y 'subprocess' para ejecutar procesos externos.

2. Clase 'MyApp': La clase principal de la aplicación que hereda de 'App' de Kivy. Contiene métodos para construir la interfaz gráfica y manejar la ejecución del script externo.

3. Método 'build': Este método construye la interfaz gráfica de la aplicación. Crea un 'BoxLayout' vertical que contiene una 'Label' con un mensaje inicial y un 'Button' que, al ser presionado, ejecuta el script 'Model.py'. El botón está vinculado al método 'run_script'.

4. Método 'run_script': Este método se ejecuta cuando se presiona el botón. Utiliza 'subprocess.run' para ejecutar el script 'Model.py' y captura la salida estándar y de error. Si la ejecución del script es exitosa, actualiza la 'Label' con el mensaje "Script ejecutado correctamente" y la salida del script. Si hay un error, muestra el mensaje de error en la 'Label'.

5. Punto de entrada de la aplicación: La aplicación se inicia ejecutando el método 'run' de la clase 'MyApp' dentro de un bloque 'if __name__ == '__main__'', asegurando que la aplicación solo se ejecute si el script se ejecuta directamente y no si se importa como un módulo.

Se puede consultar el código en el siguiente [enlace](#) de GitHub

Model.py: Es un script escrito en Python que contiene el modelo predictivo encargado de procesar datos de archivos CSV y realizar análisis de detección de anomalías. Utiliza un modelo de Machine Learning No Supervisado para identificar consumos anómalos de Energía Activa en clientes no regulados y reglas duras como la relación de Energía Reactiva sobre la Energía Activa para identificar excesos en estos consumos puesto que disminuyen la calidad de la energía. Los componentes principales son:

1. Importaciones:

Se importan las bibliotecas necesarias para la manipulación de datos y la implementación del modelo de machine learning:

- *pandas* - para la manipulación y análisis de datos estructurados.
- *numpy* - para operaciones matemáticas y manejo eficiente de arreglos.
- *os* - para interactuar con el sistema de archivos y listar los archivos en un directorio.
- *IsolationForest* de *sklearn.ensemble* - para la detección de anomalías mediante un enfoque no supervisado.

2. Carga y Concatenación de Archivos CSV:

- Se define la ruta a una carpeta que contiene múltiples archivos CSV con datos de consumo de energía de diferentes clientes.
- Se listan todos los archivos CSV en la carpeta especificada.
- Se crea un DataFrame vacío llamado 'all_data' para almacenar los datos combinados.
- Se itera sobre cada archivo CSV, extrayendo el ID del cliente del nombre del archivo y leyendo su contenido. Los datos se concatenan en el DataFrame 'all_data', añadiendo una columna con el ID del cliente para mantener la referencia.

3. Carga y Procesamiento de Archivo Excel:

- Se carga un archivo Excel que contiene información adicional sobre los clientes, como su sector económico.
- Se extrae el ID del cliente de una columna del archivo Excel y se agrega esta información como una nueva columna en el DataFrame.

4. Fusión de Datos:

Se fusionan los datos concatenados de los archivos CSV con la información adicional del sector económico obtenida del archivo Excel. La fusión se realiza utilizando el ID del cliente como clave, integrando toda la información en un único DataFrame 'data_merged'.

5. Filtrado y Ordenación de Datos:

- Se filtran los valores negativos en la columna de energía activa, reemplazándolos por cero para evitar errores en el análisis.
- Se ordenan los datos por ID de cliente y fecha para facilitar el análisis temporal y la detección de patrones anómalos en el consumo de energía.

6. Detección de Anomalías en la Relación Energía Reactiva/Activa:

- Se define una función 'anomalias_ER_EA' que detecta anomalías en la relación entre energía reactiva y activa para cada cliente.
- Esta función agrupa los datos por mes, calcula la relación entre energía reactiva y activa, y marca como anómalos aquellos meses donde la relación supera el 30%.

7. Evaluación de Todos los Clientes:

- Se define una función 'evaluar_todos_los_clientes' que aplica la detección de anomalías a todos los clientes.
- Esta función itera sobre cada cliente, aplica la función 'anomalias_ER_EA', y concatena los resultados en un DataFrame final 'df_anomalias_ER_EA'.
- Los resultados se guardan en un archivo CSV para su posterior análisis para ser consumido en el Power BI.

8. Modelo Isolation Forest para Detección de Anomalías:

- Se utiliza el modelo Isolation Forest para detectar patrones anómalos en el consumo de energía activa de los clientes.
- Se define un diccionario con los mejores hiperparámetros para cada cliente ('best_params_dict').
- Se define una función 'apply_isolation_forest' que aplica Isolation Forest a los datos de un cliente específico, utilizando los parámetros predefinidos.
- Se define una función 'detect_anomalies_criticidad' que evalúa la criticidad de las anomalías detectadas, asignando niveles de criticidad (leve, moderada, crítica) basados en el Z-Score de la energía activa.

9. Guardar Resultados:

- Se itera sobre todos los clientes, aplicando las funciones de detección de anomalías y criticidad.
- Los resultados se almacenan en un DataFrame final 'df_anomalias_IF'.
- Este DataFrame se guarda en un archivo CSV para ser consumido en el Power BI.

Este script está diseñado para proporcionar un análisis integral del consumo de energía de los clientes, utilizando técnicas de detección de anomalías para identificar patrones inusuales que podrían indicar problemas o áreas de mejora en la gestión del consumo energético para consultar el código a detalle te invito a revisarlo en el siguiente [enlace](#) de GitHub.

2.3 Acciones para la instalación y puesta en marcha de la herramienta Power BI

2.3.1 Descargar e Instalar Power BI

1. Visitar el sitio web de Power BI: Abre el navegador web de tu preferencia y dirígete a la página oficial de Power BI. [enlace](#)

2. Descargar Power BI Desktop:

- En la página de inicio, haz clic en el botón "Descargar gratis".
- Selecciona "Power BI Desktop" y haz clic en "Descargar".
- El archivo de instalación se descarga en tu equipo.

3. Instalar Power BI Desktop:

- Navega hasta el archivo descargado (generalmente en la carpeta de Descargas).
- Haz doble clic en el archivo de instalación para abrir el asistente de instalación.
- Sigue las instrucciones del asistente de instalación para completar la instalación.

2.3.2 Crear la conexión del Tablero de Power BI con las fuentes de datos

1. Dirígete a la carpeta del proyecto y ubica el archivo “.pbix” como se muestra en la siguiente imagen:

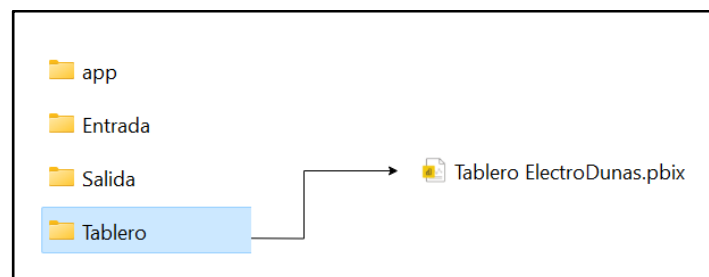


Figura 5. Ubicación del tablero (archivo .pbix)

2. Conectar a Datos:

- Haz doble clic sobre el archivo “Tablero ElectroDunas.pbix” y esto abrirá el tablero.
- En la parte superior del tablero selecciona "Transformar datos" en la pestaña hacia abajo y se desplegará un menú como el que sigue, en el selecciona la opción “Configuración de origen de datos”:

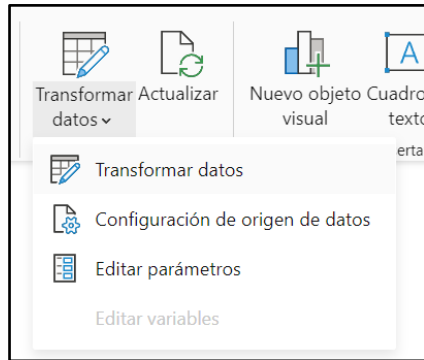


Figura 6. Cambio de orígenes de datos

- En la nueva interfaz que se despliega cambia las rutas apuntadoras a donde se encuentran los archivos. Los archivos están divididos en dos carpetas una la carpeta con los archivos crudos llamada “Entrada” y otra con los archivos procesados por el algoritmo llamada “Salida” por lo cual se deben cambiar con las rutas donde se haya dispuesto la carpeta del modelo:
 - a. tu_ruta_del_proyecto\Entrada
 - b. tu_ruta_del_proyecto\Entrada\sector_economico_clientes.xlsx
 - c. tu_ruta_del_proyecto\Salida\df_anomalias_ER_EA.csv
 - d. tu_ruta_del_proyecto\Salida\df_anomalias_IF.csv

Las rutas por defecto lucen como la siguiente imagen:

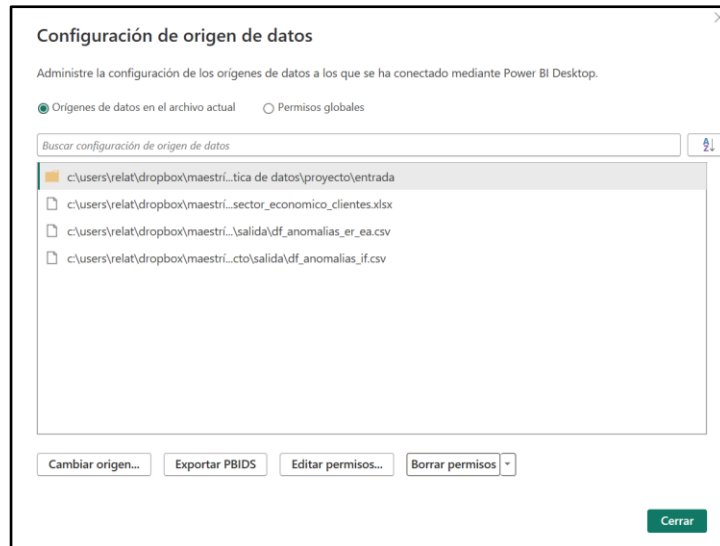


Figura 7. Cambio de rutas en los orígenes de datos

Una vez se cambian las rutas se debe hacer clic en el botón “Cerrar”.

- Para finalizar en la parte superior del tablero de Power BI presionar el botón “Actualizar”

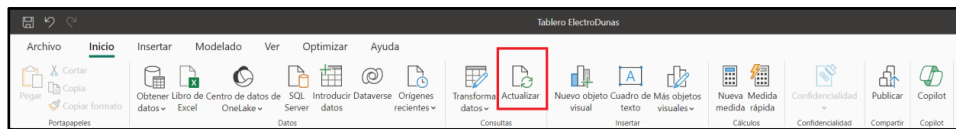


Figura 8. Actualización del tablero con el cambio de orígenes de datos

Con todo el procedimiento anterior el Tablero queda actualizado con toda la información disponible.

3. Actualización del Artefacto

Para actualizar el artefacto es necesario llevar a cabo los siguientes pasos de manera general:

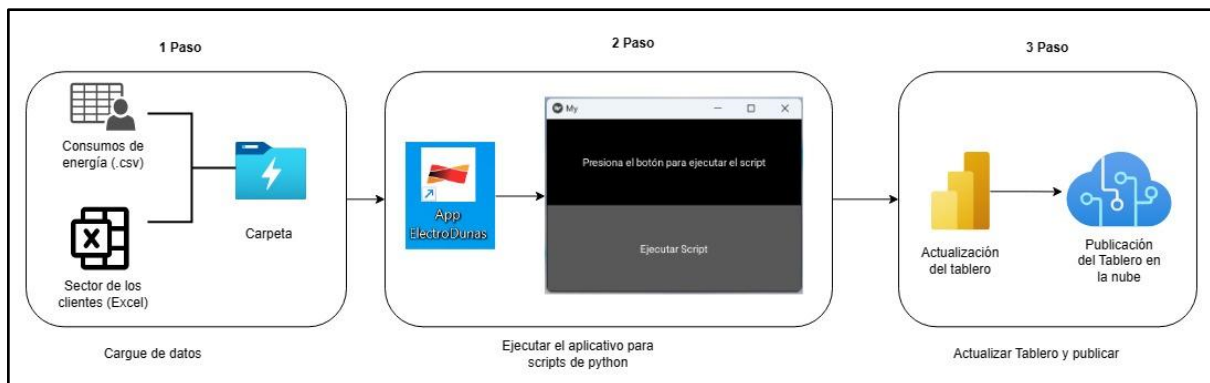


Figura 9. Diagrama paso a paso del proceso de actualización del Artefacto

A lo largo del desarrollo de este instructivo se dará mayor detalle de los componentes involucrados en cada paso y lo que se debe hacer para actualizar de manera exitosa los documentos:

Paso 1: Cargue de datos

El artefacto presenta la siguiente estructura de Carpetas:

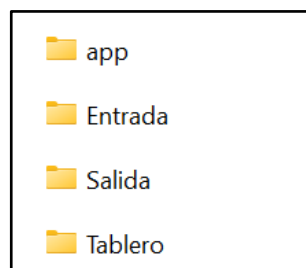


Figura 10. Estructura de carpetas del Artefacto

El usuario debe colocar en la carpeta llamada “*Entrada*” los siguientes archivos:

a. **DATOSCLIENTE:** De cada cliente que se desea estudiar en formato “.csv” y que inicializan con la descripción “DATOSCLIENTE” seguido del número asignado al interior de la compañía “1”, a modo de ejemplo se ilustra el contenido de los primeros clientes en la carpeta “*Entrada*”

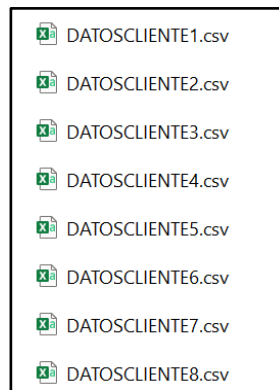


Figura 11. Nomenclatura necesaria de los archivos contenidos en la carpeta “Entrada”

Es importante resaltar que cada archivo “.csv” debe estar construido bajo la misma estructura de datos en que fue entregada la información por parte de la empresa para al construcción del desarrollo, a continuación se ilustra un ejemplo de la estructura interna del archivo llamado “DATOSCLIENTE1” (nótese que son datos separados por coma y la separación de decimales es con punto):

```

Fecha,Active_energy,Reactive_energy,Voltaje_FA,Voltaje_FC
2021-01-01
00:00:00,0.357840980804741,0.2827883838970557,455.139170896401
9,510.5610016387444
2021-01-01
01:00:00,0.3722642351844329,0.4313773208478047,469.97878735610
61,469.91717779181977
2021-01-01
02:00:00,1.0446868278946968,0.3386255117629699,468.72112010681
343,546.9491471624839
2021-01-01
03:00:00,0.5664245271391556,0.4957914849574403,452.32925499200
31,444.12298923388977
2021-01-01

```

Figura 12. Estructura del archivo llamado “DATOSCLIENTE1”

b. sector economico clientes: Es un archivo “.xlsx” en el cual reposan solo dos columnas relevantes “Cliente” y “Sector Económico”. Este archivo de atributos debe de corresponder con el nombre asignado al cliente en el punto a, puesto que es el que permite rastrear el sector de pertenencia del cliente y se debe colocar siempre el mismo nombre “sector_economico_clientes” como se muestra a continuación:

sector_economico_clientes.xlsx

	A	B
1	Cliente:	Sector Económico:
2	Cliente 1	Elaboración de cacao y chocolate y de productos de confitería
3	Cliente 2	Elaboración de cacao y chocolate y de productos de confitería
4	Cliente 3	Elaboración de cacao y chocolate y de productos de confitería
5	Cliente 4	Elaboración de cacao y chocolate y de productos de confitería
6	Cliente 5	Elaboración de cacao y chocolate y de productos de confitería
7	Cliente 6	Cultivo de Árboles Frutales y Nueces

Figura 13. Nombre del archivo “sector_economico_clientes.xlsx” (Fig sup) y la estructura del archivo en su interior (Fig inf)

Una vez se encuentren los archivos respectivos en las carpetas asignadas, se procede con el siguiente paso.

Paso 2: Ejecutar el aplicativo para scripts de python

En este punto se procede a abrir el acceso directo que se puede colocar en el escritorio de manera local para correr la aplicación, siguiendo los siguientes pasos:

- Se hace doble clic en el acceso directo de la aplicación llamada “App ElectroDunas”

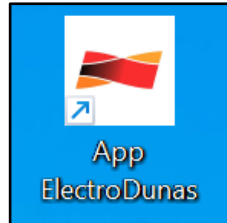


Figura 14. Acceso directo del ejecutable “App ElectroDunas”

- Una vez ejecutado el paso anterior se despliega una interfaz con la siguiente apariencia:

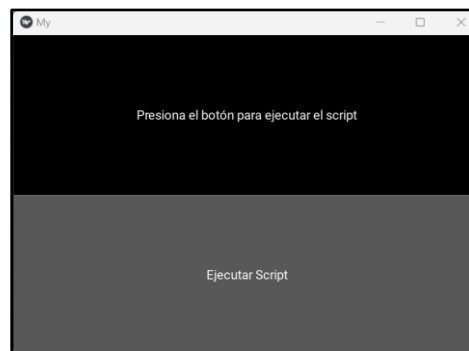


Figura 15. Interfaz de la aplicación “App ElectroDunas”

En ella se puede observar dos secciones, la parte superior es de carácter informativo y da cuenta del proceso ejecutado. En la parte inferior, se encuentra el botón llamado “Ejecutar Script” y al ser presionado corre el script de python llamado “Model.py” a lojado en la carpeta llamada app como se ilustra en la siguiente figura:

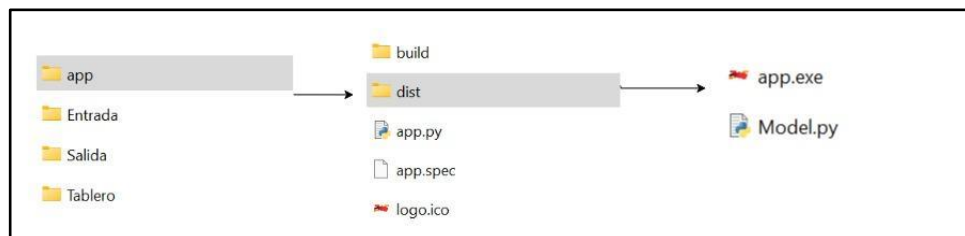


Figura 16. Estructura de carpetas de la aplicación que corre el modelo (Model.py)

Una vez el modelo termina de correr en la parte superior de la interfaz aparece el siguiente mensaje:

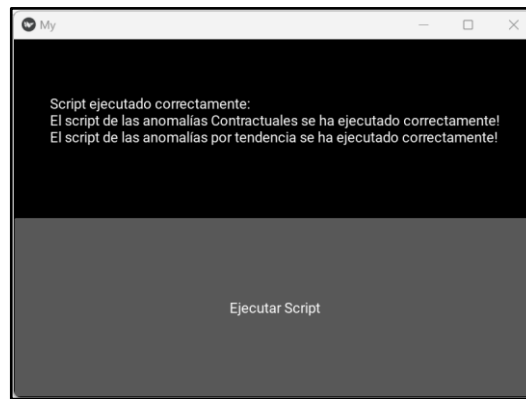


Figura 17. Mensaje de salida luego de correr el modelo (Model.py)

Al correr dicho modelo ya se puede cerrar la interfaz y se puede consultar la carpeta llamada “Salida” en donde se encontrarán dos archivos necesarios para el proceso siguiente de actualización del tablero de Power BI.

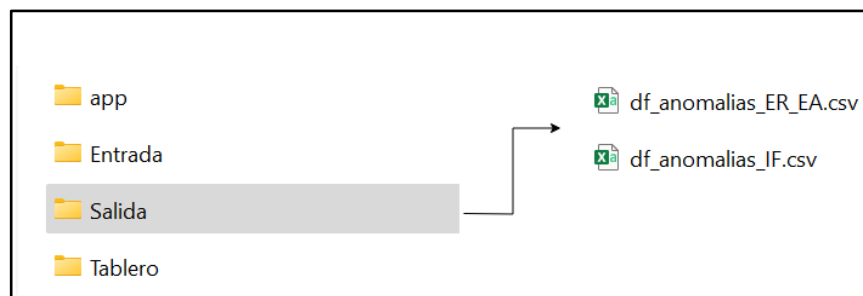


Figura 18. Salida de la ejecución del Script de python Model.py

Paso 3: Actualizar Tablero y publicar

En este punto se procede a abrir el ejecutable del Dashboard llamado “Tablero ElectroDunas” en la carpeta “Tablero”

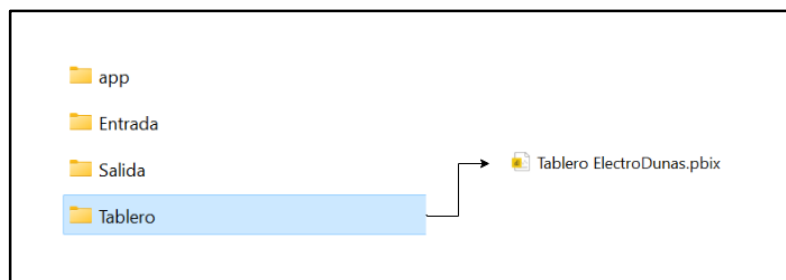


Figura 19. Carpeta que contiene el tablero de Power BI

Cuando se da doble clic en el ejecutable, se abre la interfaz del Artefacto con la siguiente vista:

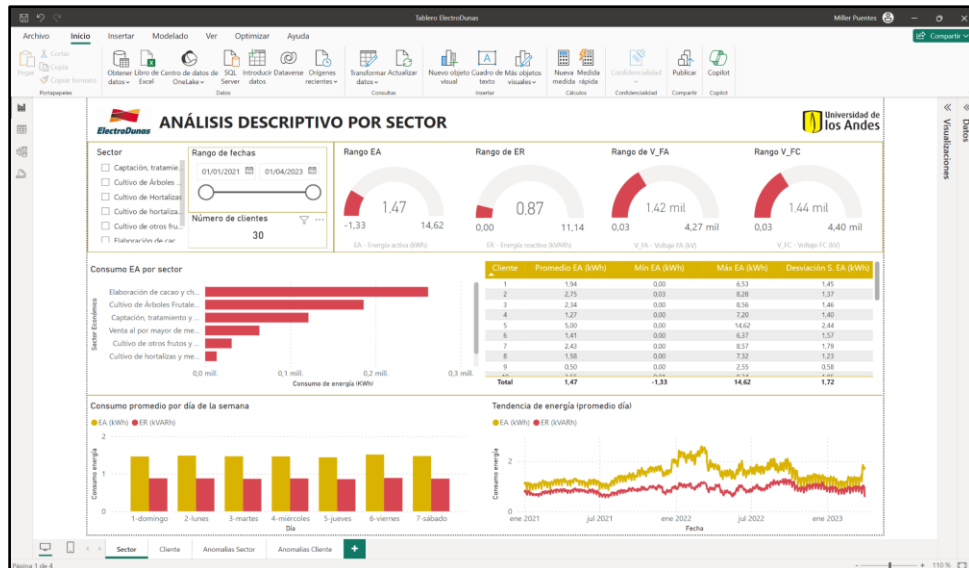


Figura 20. Vista inicial del Artefacto (Tablero ElectroDunas)

Posicionado en esta vista en la parte superior se debe hacer clic en el icono llamado “Actualizar”:

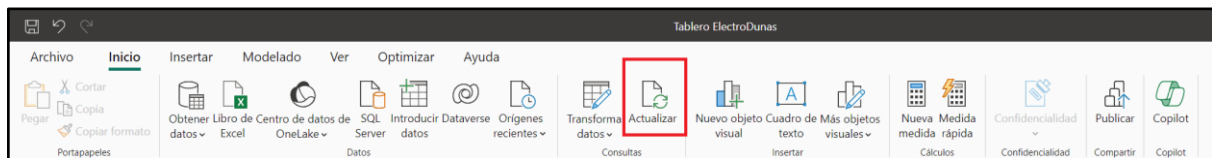


Figura 21. Botón “Actualizar” de la interfaz de Power BI

Posteriormente se abrirá una ventana informativa con un proceso similar al que se puede ver aquí, pero eventualmente desaparece cuando se termina el proceso de actualización:



Figura 22. Actualización de los datos del Artefacto en Power BI

Una vez termine el proceso de actualización, se está listo para publicar la solución en la nube y para ello se procede a dar clic en la opción superior que dice “Publicar”

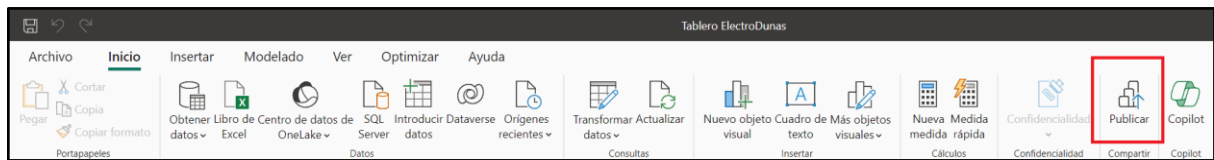


Figura 23. Publicación de los datos del Artefacto en Power BI (Nube)

Cuando se le indica que publique primero nos pregunta si deseamos guardar los cambios y en este caso se le indica “Guardar”

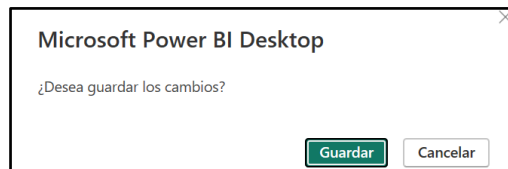


Figura 24. Guardar los cambios antes de publicar

En la nueva pestaña que se despliega se le indica que se publique en el área de trabajo “Mi área de trabajo” (por defecto) y “Seleccionar”:



Figura 25. Elección del área de trabajo

Cuando el proceso termina se ilustra de la siguiente manera:



Figura 26. Proceso de publicación finalizado

En los enlaces proporcionados por Power BI se debe abrir el primer enlace llamado “Abrir ‘Tablero ElectroDunas.pbix’ en Power BI”, mostrando en el navegador predeterminado el Artefacto:

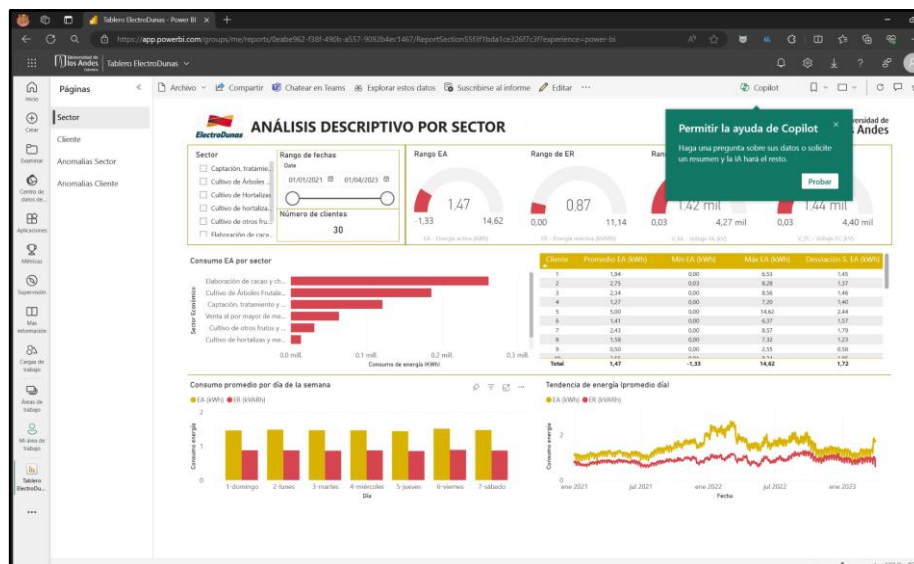


Figura 27. Artefacto desplegado en la nube

En este punto es importante aclarar que, si bien para todo el proceso anteriormente descrito se ha utilizado herramientas de uso libre, para el caso de publicar en la nube también es posible. Sin embargo, se sugiere que los datos sigan conservando el carácter de anonimización ya que no se tiene una capa de seguridad adicional como cuando se paga un servicio en la nube de Microsoft para una licencia llamada “PRO” y se publica en un espacio de trabajo específico lo cual podría tener un costo aproximado de 10 a 12 USD, lo que también permitiría publicar el tablero de manera segura y compartirlo con miembros específicos de la compañía, así que vamos a explicar este procedimiento por si es de interés para ElectroDunas:

Para compartir el Artefacto con otros miembros de la compañía se sigue el siguiente proceso:

- Hacer clic en la parte superior de la ventana en el ícono llamado “Archivo”, luego hacer clic en el desplegable “</> Insertar informe” y finalmente en el ícono de “Sitio web o portal”

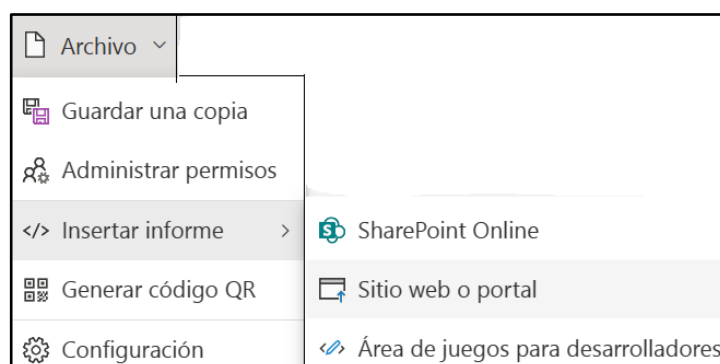


Figura 28. Compartir el Artefacto con otros miembros de la empresa

- En la nueva ventana que se abre se despliegan dos enlaces, se toma el primero y se envía a las personas a las que se desea otorgar el acceso a la herramienta.



Figura 29. Enlace para compartir la herramienta

- c. Haciendo uso del primer enlace y abriéndolo en el navegador se expone el Artefacto de la siguiente manera:



Figura 30. Artefacto abierto desde un enlace público

Demos de actualización:

Para mayor claridad del procedimiento se crearon dos videos que se pueden consultar para la actualización del artefacto, se encuentran alojados en YouTube:

- Actualización de Archivos. [enlace](#)
- Actualización del Dashboard. [enlace](#)

El video llamado “*Demo Actualización app*” muestra un video para actualizar los scripts de python y el video llamado “*Demo Actualización Power BI*” es el instructivo visual para actualizar el Dashboard.

Contacto y soporte:

Para cualquier información adicional, duda o requerimiento, por favor contactar a cualquiera de los miembros del equipo desarrollador:

- Grace González
ga.gonzalez1804@uniandes.edu.co
- Miller Puentes

m.puentesl@uniandes.edu.co

- Camilo Robayo
c.robayoa@uniandes.edu.co
- Joan Chacón
j.chaconq@uniandes.edu.co