

MEMORIA DE FINALIZACIÓN

DE PRÁCTICAS

UNIVERSIDAD NEBRIJA GRADO EN

INGENIERÍA INFORMÁTICA MEMORIA

PRÁCTICAS EN EMPRESA



MEMORIA DE FINALIZACIÓN

DE PRÁCTICAS

UNIVERSIDAD NEBRIJA GRADO EN

INGENIERÍA INFORMÁTICA MEMORIA

PRÁCTICAS EN EMPRESA



Derechos

D. Guillermo Martínez de Hurtado Aricha autoriza a que el presente trabajo se guarde y custodie en los repositorios de la Universidad Nebrija y además NO autoriza a su disposición en abierto.

**Contenido**

[Derechos 3](#_kv31ibqc9k1v)

[Índice de ilustraciones 5](#_o4hof8vabfx1)

[Glosario de términos 6](#_hi6kpuu1rup7)

[1. RESUMEN 9](#_2mpw2ar1by77)

[2. PRÁCTICAS 9](#_b3had5coe0d4)

[2.1. Antecedentes 9](#_ybsxjj6h4yn0)

[2.2. Objetivos de las prácticas 14](#_fzqtg4dq65c5)

[2.3. Onboarding 15](#_61we4bwrshk4)

[2.4. Funciones y tareas en las prácticas. (evidencias) 18](#_bknplxr2d2yg)

[2.5. Relaciones de problemas planteados y procedimientos para su resolución 26](#_zh40x4b0udoa)

[2.6. Aprendizajes y desarrollo profesional (habilidades adquiridas) 29](#_stndacwlhgco)

[2.7. Metodologías utilizadas 30](#_a3hhv8xts8b0)

[2.8. Herramientas utilizadas en las prácticas (evidencias) 30](#_kf6g6hyg10dq)

[2.9. Logros, resultados y discusión. 34](#_520ycz8h5suw)

[3. DESARROLLO 35](#_g6yg5y2cbju8)

[3.1. Introducción 35](#_tyjl3ogcjd1d)

[3.2. Marco técnico 38](#_on7fqt1182bv)

[3.3. Equipo de trabajo y metodología 38](#_g67y4jutogex)

[3.4. Proyecto 39](#_pj06ixetwek9)

[3.5. Resultados y discusión 41](#_2cfu7r54vk45)

[3.6. Conclusiones 41](#_8rltzaq9b32p)

[3.7. Líneas futuras 42](#_a7g5q4qmo7ac)

[4. BIBLIOGRAFÍA. 42](#_gar9lhqbii5d)

[5.](#_zeyz1zlqyv7b) ANEXOS 43

[Anexo 1: Código de python 43](#_sp74sh6ktmc7)

[Anexo 2: PBI 48](#_c2wq3zj1cg1y)

[Modelo de datos del PBI: 48](#_stobonnf7bly)

Índice de ilustraciones

[Ilustración 1: Logotipo de Serveo. (Serveo, n.d.) 10](#_7jd2mqj9rhoi)

[Ilustración 2: Horario de onboarding en Serveo. (Elaboración propia) 16](#_h8x9ogg11rr8)

[Ilustración 3: Jornada I, Eficiencia Operativa. (Elaboración propia) 17](#_4bhxsp8spfn0)

[Ilustración 4: Jornada II, Eficiencia Operativa. (Elaboración propia) 17](#_4kqjxwr8bszo)

[Ilustración 5: Aplicación OCR python en local, pruebas y errores. (Elaboración propia) 19](#_bbui4kwgdpk7)

[Ilustración 6: Power Automate, Ejemplo de flujo. (Elaboración propia) 20](#_kjwrtm18jcwn)

[Ilustración 7: Power Apps. (Elaboración propia) 22](#_2ve7dzgy7wc)

[Ilustración 8: Listas Sharepoint. (Elaboración propia) 22](#_k546fbjxltxy)

[Ilustración 9: PDF Estándar Serveo. (Elaboración propia) 23](#_u07h9wov7lt7)

[Ilustración 10: Interfaz Microsoft Fabric. (Elaboración propia) 24](#_xhlbwv81lpik)

[Ilustración 11: Herramientas en Microsoft Fabric. (Elaboración propia) 25](#_nsah6lnah7zc)

[Ilustración 12: Lagos de Datos, Microsoft Fabric. (Elaboración propia) 25](#_jiqx1mkwx48m)

[Ilustración 13: Flujos de datos, Microsoft Fabric. (Elaboración propia) 26](#_khon1sl8btcq)

[Ilustración 14: Librerías Python. (Elaboración propia) 32](#_572k6yt0q1er)

[Ilustración 15: XML. (Gobierno de España, 2025) 36](#_ykmgao11m734)

[Ilustración 16: Resultado terminal script. (Elaboración propia). 40](#_89vkmieqdsxz)

[Ilustración 17: Power Query. (Elaboración propia) 40](#_aw5lxru2wbr5)

Glosario de términos

**API (Interfaz de Programación de Aplicaciones):**

Conjunto de protocolos y herramientas que permiten la comunicación entre distintos sistemas. En este trabajo se utiliza para acceder a datos públicos, como los precios de carburantes ofrecidos por el gobierno.

**CSV (Comma-Separated Values):**

Formato de archivo que almacena datos en forma de texto plano, donde cada línea representa un registro y los campos están separados por comas (u otros delimitadores). Se emplea para almacenar y manipular datos extraídos de archivos XML.

**Data Lake (Lago de Datos):**

Repositorio centralizado que permite almacenar datos en su formato original, sean estructurados o no estructurados. En el proyecto con Microsoft Fabric se crean lagos de datos para integrar y transformar la información proveniente de diversas fuentes.

**Delta (Formato Delta):**

Formato de almacenamiento de datos optimizado para trabajar con lagos de datos, permitiendo actualizaciones y transacciones de manera eficiente.

**Eficiencia Operativa:**

Conjunto de prácticas y procesos enfocados en optimizar el rendimiento y la productividad dentro de una organización, a través de la automatización, mejora de procesos y uso de tecnología.

**Excel:**

Herramienta de Microsoft Office ampliamente utilizada para el manejo, análisis y visualización de datos en formato de hojas de cálculo.

**IA (Inteligencia Artificial):**

Campo de estudio de la informática que busca desarrollar sistemas capaces de aprender, razonar y tomar decisiones de manera autónoma. En el contexto del proyecto se emplea para entrenar modelos que permitan extraer información de documentos repetitivos.

**Log (Registro de Eventos):**

Archivo que almacena información y eventos ocurridos durante la ejecución de un programa. Se utiliza para el seguimiento y depuración de procesos en los scripts desarrollados.

**OCR (Reconocimiento Óptico de Caracteres):**

Tecnología que permite convertir diferentes tipos de documentos, como archivos PDF o imágenes, en datos editables y procesables. Se ha utilizado para extraer IBANs de documentos PDF.

**Planner:**

Herramienta de Microsoft 365 que facilita la gestión de tareas y proyectos, permitiendo asignar actividades, establecer fechas límite y coordinar el trabajo en equipo.

**Power Automate:**

Plataforma de Microsoft que permite la automatización de flujos de trabajo sin necesidad de escribir código (no-code). Se utiliza para tareas repetitivas, como la automatización del procesamiento de correos electrónicos.

**Power BI:**

Herramienta de visualización y análisis de datos de Microsoft que permite transformar datos en informes y dashboards interactivos, facilitando la interpretación de la información.

**Power Query:**

Componente de Microsoft que facilita la transformación, limpieza y preparación de datos, utilizando tanto una interfaz visual como el lenguaje de programación M para personalizaciones avanzadas.

**PowerApps:**

Plataforma de Microsoft que permite crear aplicaciones empresariales de forma rápida, con muy poco o sin necesidad de código. Se ha empleado para desarrollar aplicaciones internas, como el registro de EPIs.

**Pipeline (Flujo de Datos):**

Secuencia de procesos automatizados que permiten la ingesta, transformación y almacenamiento de datos. En Microsoft Fabric se utilizan pipelines para gestionar la actualización de datos.

**Python:**

Lenguaje de programación de alto nivel, muy utilizado en análisis y automatización de datos por su versatilidad y amplia disponibilidad de librerías.

**PySpark:**

API de Python para trabajar con Apache Spark, utilizada para el procesamiento y análisis de grandes volúmenes de datos de forma distribuida.

**SQL (Structured Query Language):**

Lenguaje de programación especializado en la gestión y manipulación de bases de datos relacionales. Se utiliza para realizar consultas y transformaciones en las bases de datos del proyecto.

**SharePoint:**

Plataforma de Microsoft para la gestión y almacenamiento de documentos y datos, que permite la colaboración y el acceso compartido a la información dentro de la empresa.

**XML (Extensible Markup Language):**

Lenguaje de marcado utilizado para almacenar y transportar datos de forma estructurada. En el proyecto se extraen datos en formato XML para luego transformarlos a CSV.

1. RESUMEN

Este trabajo documenta y analiza el recorrido y las experiencias acumuladas durante mis prácticas en el Departamento de Eficiencia Operativa y Análisis de Datos. En él se recoge el conjunto de proyectos que he llevado a cabo hasta la fecha, abarcando tanto la parte de eficiencia operativa como la de análisis de datos. Durante este periodo, he tenido la oportunidad de trabajar en la automatización de tareas, la transformación de datos y el desarrollo de aplicaciones que optimizan procesos internos. Entre los proyectos más destacados se encuentran

La segunda parte de este trabajo se dedicará en profundidad

2. PRÁCTICAS

2.1. Antecedentes

2.1.1. ¿Qué entrevistas he hecho?

Para la asignatura de Evaluación del desarrollo de capacidades en la empresa II he continuado desarrollando mis prácticas como becario en Serveo, misma empresa que para la asignatura anterior.

A la hora de firmar las prácticas para el primer cuatrimestre, me ofrecieron hacer también tanto las horas del segundo cuatrimestre como 300 horas más extracurriculares que acabarán en junio.

2.1.2. Datos de la empresa



*Ilustración 1: Logotipo de Serveo. (Serveo, n.d.)*

**Nombre y ubicación**

Continuo mis prácticas en Serveo, compañía cuyas oficinas centrales están en Las Tablas, Madrid, aunque al tratarse de una compañía que se dedica principalmente al Facility Management industrial es común visitar algún contrato y trabajar desde las oficinas del contrato al que se va a dar apoyo.

**Fechas de realización**

Para la segunda asignatura de evaluación de capacidades en la empresa mi periodo de realización de prácticas es desde enero hasta abril, y como he mencionado antes haré horas extracurriculares hasta junio.

**Descripción de la actividad de la empresa**

Serveo es una empresa del sector de servicios especializada en el mantenimiento y la gestión integral de infraestructuras, facility management y servicios relacionados. Su propuesta de valor se basa en ser un **integrador transversal de servicios de facility management**, lo que le permite centralizar la interlocución con el cliente y optimizar recursos para impulsar la eficiencia operativa y los objetivos de sostenibilidad (Net Zero). A través de un equipo multidisciplinar de más de 47 000 profesionales, Serveo cubre desde el mantenimiento técnico de instalaciones (Hard Services) hasta servicios de limpieza, jardinería, seguridad, teleasistencia y atención a personas vulnerables (Soft Services), además de servicios auxiliares como gestión de residuos y prevención de riesgos laborales.

La compañía opera en cinco grandes sectores:

* **Industria**: Incluye mantenimiento eléctrico, mecánico, climatización, automatización de líneas de producción y otros servicios técnicos especializados.
* **Salud**: Comprende servicios en hospitales, centros de salud y residencias, tales como mantenimiento de equipamiento médico, limpieza especializada, servicios sociales y atención a pacientes.
* **Corporate**: Engloba a empresas y sedes corporativas donde ofrece servicios integrales que incluyen mantenimiento de edificios, gestión de espacios, limpieza de oficinas, jardinería y conserjería.
* **Government (Administraciones Públicas)**: Presta servicios a entidades gubernamentales en edificios administrativos, organismos públicos y centros educativos, con énfasis en la eficiencia energética y el cumplimiento normativo.
* **Infraestructuras**: Atiende aeropuertos, carreteras, estaciones y otras infraestructuras críticas, ofreciendo servicios a bordo (en aeropuertos), mantenimiento de carreteras (Skyway) y conservación de infraestructuras viarias.

Serveo se autodefine como **impulsor del cambio y del progreso sostenible**, comprometiéndose con la innovación, el desarrollo de las personas y la mejora continua mediante procesos basados en indicadores de gestión y tecnología avanzada para garantizar la fiabilidad de sus servicios.

**Descripción de la estructura organizativa y puesto ocupado en ella**

Internamente, Serveo organiza su modelo de negocio en cinco áreas principales para facilitar la comprensión por parte de clientes y socios externos. Sin embargo, a nivel interno, la empresa se estructura en distintas delegaciones según la región o el sector: Delegación Norte, Delegación Sur, Delegación Centro, Delegación Industria y Delegación Servicios.

Durante mis prácticas como Serveo sigue creciendo y escalando puestos a nivel empresa en España y con una estrategia de internacionalización en sus planes, mi puesto se ha visto afectado pasando de ser soporte a todos los departamentos de estructura(Recursos Humanos, Dpto. Financiero, …) a dar soporte a la parte de negocio, es decir, a todos los contratos como Airbus, Iberia, hospitales públicos,....

**Tecnologías y metodologías utilizadas**

Serveo apuesta por la innovación y la optimización de procesos mediante herramientas y enfoques que facilitan el trabajo diario y mejoran la colaboración. Entre las principales tecnologías y metodologías que emplea se encuentran:

* **Power BI, Excel y Power Query** para la visualización, análisis y transformación de datos, permitiendo generar informes dinámicos y dashboards que ayudan a la toma de decisiones.
* **Power Automate y PowerApps** para automatizar flujos de trabajo rutinarios y crear aplicaciones de bajo código que simplifican tareas operativas sin necesidad de programación compleja.
* **Python** para desarrollar aplicaciones personalizadas que facilitan el día a día de los usuarios, desde pequeños scripts de automatización hasta soluciones más completas que integran bases de datos y servicios web.
* **Planner y SharePoint** como herramientas colaborativas, que permiten organizar tareas, gestionar proyectos en equipo y compartir documentación de forma centralizada y ordenada.
* **SQL** para facilitar el manejo de bases de datos y ser capaz de transformar grandes cantidades de datos.

**Descripción del puesto**

Para este segundo cuatrimestre me he centrado sobretodo en 3 de las herramientas que he mencionado en el apartado anterior aunque la finalidad de mi puesto sigue siendo la misma, capacidad de transformar los datos para que todas las personas puedan entenderlos y crear herramientas que agilicen el día a día de todo el personal de la empresa, las herramientas con las que más he trabajado son:

* **Python,** durante este segundo periodo me he visto en la constante necesidad de crear scripts y ejecutables con python usando librerías distintas y variadas de las que hablaré más adelante en otro de los apartados.
* **Power BI** con el fin de procesar los datos de la compañía y tener una forma interactiva y dinámica de visualizarlos, power BI es la mejor herramienta sobretodo para la parte de visualizarlos, aunque es cierto que para la parte de transformar datos he usado otras herramientas más potentes como Python, mencionada antes o **Big Query**.
* Desarrollo de aplicaciones con **Power apps**, he continuado con el desarrollo de una aplicación que mencionaba el cuatrimestre pasado, para facilitar el registro de la entrega de EPIs, a la vez que lo he usado para crear nuevas aplicaciones y con la ayuda de **Sharepoint** y **Power Automate** he conseguido desarrollar flujo de trabajo que facilitan el día a día al personal de la empresa.

**Descripción del Sector. Competidores, evolución, tendencias y oportunidades**

El entorno en el que se mueve Serveo es uno de los sectores de servicios más dinámicos y exigentes, marcado por la rápida incorporación de nuevas tecnologías. La adopción de la digitalización y la automatización se ha convertido en un factor decisivo, obligando a las empresas a reinventar sus procesos para no quedarse rezagadas. En este escenario, compiten grandes firmas de facility management y proveedores de servicios integrales, todos ellos en la carrera por desarrollar propuestas cada vez más eficaces e innovadoras. Esta transformación constante abre la puerta a la adopción de herramientas avanzadas de análisis y gestión de datos, lo que mejora notablemente la eficiencia operativa y la capacidad de respuesta ante las demandas del mercado. Consciente de este contexto, Serveo ha puesto en marcha un ambicioso plan de expansión hacia Europa, respaldado en su know‑how y en su modelo integral, para llevar su oferta de servicios a nuevos territorios.

**Posicionamiento de la empresa en el sector. Puntos fuertes y débiles. Aspectos mejorables**

Serveo ocupa un lugar destacado entre las principales firmas de facility management en España, sustentando su liderazgo en una estrategia de crecimiento constante y diversificación en sectores como Industria, Salud y Administración Pública. Su capacidad para gestionar infraestructuras de alto nivel se refleja en la reciente obtención de la certificación ISO 41001, que avala la excelencia en la gestión de inmuebles y servicios de soporte, así como en su rápida integración de equipos y proyectos tras la adquisición de Sacyr Facilities. Además, la incorporación del área de mantenimiento industrial de Dominion ha reforzado su posicionamiento técnico y ha incrementado su presencia en el sector manufacturero, aportando recursos especializados y mayores oportunidades de negocio. Entre sus puntos fuertes se encuentran la solidez de sus procesos internos, la implementación de tecnologías avanzadas para el análisis y seguimiento de KPI, y una oferta integral que abarca tanto servicios “hard” como “soft”. Sin embargo, para seguir elevando su competitividad, Serveo debería profundizar en la integración total de sus sistemas de automatización y optimizar la comunicación interna entre delegaciones, asegurando así una coordinación aún más ágil y homogénea en todas sus áreas de actuación.

2.1.3. Información previa

Durante el primer cuatrimestre ya estuve colaborando en este equipo en proyectos de análisis de datos y automatización, lo que me permitió continuar desarrollando proyectos con ellos y asignando proyectos nuevos. Debido a esa experiencia previa, al continuar con las prácticas ya tenía varias iniciativas asignadas: el equipo, consciente de mis habilidades, me confió tareas alineadas con mis fortalezas. De este modo, me encargaron proyectos de generación de informes avanzados en Power BI, desarrollo de scripts en Python para optimizar procesos operativos y creación de flujos de trabajo mediante las herramientas low‑code de Microsoft. Gracias a esa continuidad, no tuve que empezar desde cero; pude centrarme en profundizar en las soluciones concretas que se estaban desarrollando en el departamento.

En mi día a día, me dedico principalmente al análisis de datos con Power BI, construyendo dashboards interactivos que facilitan la toma de decisiones, y a la mejora de la eficiencia operativa mediante Python, desarrollando aplicaciones y automatizaciones que reducen tiempos y errores en tareas rutinarias. Además, gestiono flujos de trabajo utilizando Power Automate y Power Apps, integrando diversas fuentes de información para crear procesos robustos y escalables. Esta combinación de herramientas me permite ofrecer soluciones integrales: desde la extracción y transformación de datos hasta la generación de alertas automatizadas y la publicación de informes dinámicos.

En cuanto a mis estudios universitarios, varias asignaturas me han proporcionado una base técnica sólida que me ha sido de gran utilidad durante mis prácticas. Una de las más relevantes ha sido la asignatura de Inteligencia Artificial, donde se puso un gran énfasis en la limpieza y el preprocesamiento de datos. Durante el curso, trabajamos intensamente en proyectos que requerían la preparación de grandes volúmenes de datos, algo que resulta fundamental en mi rol actual. Además, fue una de las pocas asignaturas en las que trabajamos con Python, lo que me ha permitido aplicar algunas de las técnicas de programación aprendidas en proyectos más complejos en el entorno laboral. La limpieza de datos es un paso crucial en el análisis de datos y haber practicado tanto esta habilidad en la universidad me ha facilitado muchísimo adaptarme a los procesos de análisis de datos en Serveo.

Otra asignatura que ha sido muy valiosa en mi formación es Programación Avanzada, donde aprendí a estructurar de manera correcta el código y a utilizar buenas prácticas en el desarrollo de software. Esta asignatura me enseñó la importancia de escribir código claro y bien documentado, algo esencial no solo para crear programas funcionales, sino también para trabajar de manera colaborativa en entornos empresariales. Durante esta asignatura, también aprendimos a diseñar diagramas de flujo de código y arquitectura de software, herramientas que, aunque en ese momento parecían algo teóricas, he descubierto que tienen una aplicación muy práctica en mi trabajo diario. Gracias a esos conocimientos, he podido familiarizarme rápidamente con herramientas como Power Automate y Power Apps, dos aplicaciones de Microsoft que se utilizan en Serveo para automatizar procesos y desarrollar aplicaciones internas de una manera eficiente.

Por último, la experiencia práctica adquirida en la universidad, tanto en las asignaturas como en los trabajos de clase, ha sido clave para afrontar los retos técnicos que se presentan en el día a día de mis prácticas. Haber trabajado en proyectos que simulan entornos reales me ha permitido desarrollar una capacidad de adaptación rápida a las herramientas y procesos dentro de la empresa. Además, el conocimiento adquirido en los cursos adicionales que he realizado de manera autónoma me ha dado una ventaja a la hora de integrar diferentes tecnologías en mi flujo de trabajo.

2.1.4. Motivos para elegir estas prácticas

Desde que entré en informática siempre he tenido mucho interés por todo lo que es la ciencia del dato y el manejo de grandes cantidades de datos, entonces cuando estaba buscando prácticas pensé que mejor forma de empezar en el mundo de la informática que en unas prácticas de análisis de datos y se presentó esta oportunidad, así que no dude ni un solo momento en decir que si.

Otro motivo muy importante es que tal y como avanza la inteligencia artificial todo el mundo de los datos va a estar más cotizado a la vez que por estos mismos avances de la IA trabajos como los analistas de datos y otros van a ir desapareciendo o siendo cada vez más difícil entrar en este mundo, por eso aprovechando ahora las prácticas tengo un trasfondo en mi curriculum y con suerte puedo haber entrado en este mundillo de los datos para quedarme mucho tiempo, ya que de momento me está gustando bastante todo lo que hago y me veo en un futuro siguiendo en este mundillo.

2.2. Objetivos de las prácticas

PROYECTO FORMATIVO, actividades a desarrollar por el alumno:

1. Ser partícipe en los proyectos de optimización y mejora de procesos en los que trabaja el departamento de Eficiencia Operativa, desarrollando e implantando mejoras operativas con los diferentes interlocutores (interno y externo).
2. Realizar cuadros de mando y análisis de datos, con la finalidad de obtener conclusiones y establecer recomendaciones que optimicen las operaciones.
3. Elaborar informes y reportes sobre el resultado de los análisis realizados con le objetivo de facilitar la toma de las decisiones a la dirección.
4. Dar soporte en la identificación e implantación de acciones de mejora en los diferentes servicios de la compañía.
5. Colaborar en la implantación y desarrollo de proyectos de implantación de metodología Lean /Six Sigma.
6. Trabajar realizando consultas y análisis en bases de datos en entornos SQL y Google Cloud Platform.

Ahora que ya he finalizado las prácticas puedo describir cuánto he trabajado en cada uno de los puntos que se exigían de mí en el contrato(Los citados arriba). La gran mayoría de mi trabajo se corresponde con los 3 primeros puntos, diría que alrededor de un 90% entre esos 3 puntos con el uso de Power BI(30%), Python(30%), Power Apps(15%) y Power Automate(15%), los puntos 4 y 5 diría que corresponde alrededor de un 8% de mi trabajo por la parte de implantación de los proyectos que iba desarrollando para los distintos departamentos o contratos, y por último las consultas con SQL corresponden con el 2% restante ya que solo se utiliza para la obtención y guardado de datos y no tanto para el manejo y transformación de estos.

2.3. Onboarding

Mi departamento se divide en dos áreas principales: eficiencia operativa y datos. Yo formo parte del equipo de datos, que está compuesto por tres personas: Daniel, nuestro líder (Head) y responsable de la línea estratégica; Roberto, con quien colaboro directamente la mayor parte del tiempo; y yo, encargándome de gran parte del tratamiento y la explotación de la información. Entre los tres gestionamos una parte sustancial de los datos de la empresa. A continuación, explico con más detalle las reuniones y la organización interna.

Todos los días, o prácticamente todos, nos reunimos a las 10:00 de la mañana en las llamadas Daily. Estas reuniones pueden ser presenciales o a través de videollamada si alguien está teletrabajando tengo 5 días al mes de teletrabajo que puedo gestionar como quiera aunque lo más común es hacer 1 por semana. Su duración es de unos 10 o 15 minutos, y sirven para comentar los proyectos en los que vamos a trabajar durante la jornada, repasar los problemas surgidos el día anterior y compartir las soluciones implementadas, ya que esas experiencias pueden resultar útiles para el resto del equipo. En estas sesiones también se discuten los nuevos proyectos que llegan desde otros departamentos: la mayoría son canalizados primero a Daniel, quien luego los distribuye entre Roberto y yo, asignándolos mediante Planner según la naturaleza y la complejidad de cada tarea. Por ejemplo, si se trata de un proyecto de datos de gran envergadura—como aquellos que se reportan directamente al CEO o dependen de información financiera—suele quedar en manos de Daniel. Los trabajos más rutinarios, como informes en Power BI o generación de tablas en Excel, suelen asignárselos a Roberto. A mí, por mi parte, me confían proyectos variados que implican código (especialmente Python), desarrollo en Power Apps o la creación de flujos complejos en Power Automate. Aunque al principio tuve ciertas dificultades con Power Apps y Power Automate, las fui comentando en las reuniones diarias y recibí apoyo de mis compañeros; progresivamente, he alcanzado un nivel de manejo en estas herramientas incluso superior al suyo.

Cuando tengo que abordar un proyecto para otro departamento, organizo una reunión con la persona responsable para definir requisitos y pautas. Así, coordinamos desde el alcance del alcance de la entrega hasta los criterios de validación. Esto suele estar reflejado en el calendario interno de la empresa mediante invitaciones y documentos compartidos.

Además de las reuniones diarias, he asistido a varias jornadas y eventos relevantes. Por ejemplo, participé en una jornada de becarios en el Hospital Zendal para colaborar con pacientes de Esclerosis Lateral Amiotrófica (ELA) que se encontraban en el centro de día, aportando apoyo logístico y humano a los usuarios del servicio. También tuve la oportunidad de asistir al Google Cloud Summit Madrid 2025 en IFEMA, donde pude conocer de primera mano los proyectos más recientes de inteligencia artificial presentados por Google, como demos de soluciones de IA generativa para el sector público y privado, modelos de análisis de datos en la nube y casos de éxito de migración de Big Data. Estos eventos me han permitido ampliar mi visión sobre las últimas tendencias tecnológicas y las mejores prácticas de automatización y análisis de datos, conocimientos que luego traslado a mis actividades diarias en Serveo.

2.4. Funciones y tareas en las prácticas. (evidencias)

Mi puesto en las prácticas es el de becario en eficiencia operativa y análisis de datos. Al comenzar, entramos dos becarios para el mismo puesto, o al menos con el mismo nombre. Ambos nos incorporamos al Departamento de Eficiencia Operativa como becarios en eficiencia operativa y analistas de datos. Dado que el otro becario tenía un perfil más enfocado a la ingeniería y yo venía de ingeniería informática, me asignaron principalmente a la parte de análisis de datos, mientras que a él lo orientaron más hacia la eficiencia operativa. Sin embargo, dado que vengo de ingeniería informática y mejorar el rendimiento en eficiencia operativa a menudo implica desarrollar aplicaciones, código o flujos, también me encargaron proyectos relacionados tanto con eficiencia operativa como con análisis de datos.

Para aclarar mejor esta parte, explicaré las tareas que he realizado durante las prácticas, primero enfocándome en la eficiencia operativa y luego en la parte de datos:

Eficiencia Operativa: Proyectos

Análisis de Datos: Proyectos

2.5. Relaciones de problemas planteados y procedimientos para su resolución

2.6. Aprendizajes y desarrollo profesional (habilidades adquiridas)

Durante estas prácticas he aprendido a gestionar equipos y a organizar mejor mi tiempo, lo que me ha permitido enfrentar de manera más efectiva los distintos proyectos. Además, he profundizado en el uso de metodologías para la transformación de datos, lo que me ha ayudado a estructurar y procesar grandes volúmenes de información de forma ordenada. Sin embargo, lo que más me ilusiona es la parte del proyecto de Microsoft Fabric, donde combino bases de datos SQL con transformaciones de datos en SQL y PySpark, y también todo el trabajo relacionado con el análisis de datos en tiempo real. Esta área me ha permitido explorar nuevas técnicas y herramientas, lo que no solo amplía mis conocimientos técnicos, sino que también refuerza mi pasión por el mundo del dato y la innovación.

2.7. Metodologías utilizadas

En cuanto a la organización de los proyectos, en el departamento usamos principalmente el Planner de Microsoft 365. Cada vez que llega un nuevo proyecto, se asigna directamente a la persona correspondiente del equipo a través del Planner. Esto nos permite mantener un seguimiento claro de todas las tareas, con fechas de entrega y prioridades bien definidas, y facilita la coordinación y el trabajo en equipo. Con esta metodología, todo el mundo sabe qué tiene que hacer y cuándo, lo que ayuda a que el flujo de trabajo sea más ordenado y eficiente.

2.8. Herramientas utilizadas en las prácticas (evidencias)

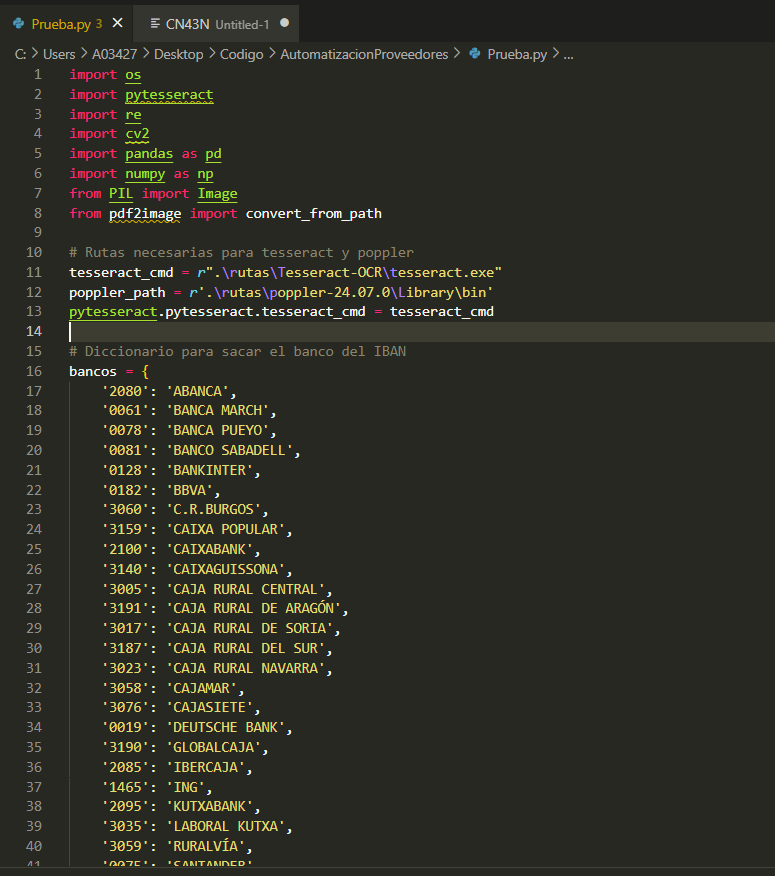
Para tratar este tema de herramientas utilizadas voy a hacer un orden de más conocidas a menos conocidas o como pienso yo, que se conoce más a las que menos se conocen.

La primera herramienta de la que voy a hablar, y que prácticamente todo el mundo en el entorno de oficina sabe utilizar, es Excel. En mi oficina, todos tienen conocimientos de Excel, ya sea a un nivel más básico o avanzado. Principalmente, Excel se utiliza para almacenar y organizar datos, facilitando la creación de tablas, gráficos y análisis sencillos. Además, es habitual que estos archivos se suban a carpetas en SharePoint para centralizar la información. Una vez allí, se pueden crear informes, reportes o incluso analizar el dato en bruto directamente desde los archivos, usando funciones avanzadas que permiten hacer cálculos complejos y automatizar tareas mediante macros.

Esto nos lleva a la segunda herramienta, Power BI, que es esencial para la creación de informes visuales y dinámicos. Aunque no es una herramienta que se use a diario en todos los puestos, un gran número de personas la conoce y reconocen su importancia para generar informes atractivos y comprensibles. Power BI permite transformar datos en dashboards interactivos, facilitando la toma de decisiones a partir de análisis en tiempo real. En mi oficina, utilizamos Power BI para reportar y analizar información de forma detallada, tanto para clientes internos como externos.

Habiendo hablado ya de Excel y Power BI, quiero explicar una herramienta que, aunque no es tan conocida, es esencial en mi puesto de trabajo: Power Query. Power Query es un entorno de trabajo basado en el lenguaje de programación M de Microsoft, diseñado específicamente para la transformación, limpieza y preparación de datos. Gracias a Power Query, es posible automatizar procesos de extracción y transformación, lo que permite integrar información de múltiples fuentes y asegurar que los datos estén en el formato adecuado antes de su análisis. Esto no solo optimiza el tiempo invertido en la preparación de los datos, sino que también ayuda a hacer mejores Power BIs.

La siguiente herramienta que he utilizado bastante ha sido Python, principalmente en Visual Studio. Esta herramienta me ha sido de gran ayuda, sobre todo para crear programas y scripts que faciliten las tareas repetitivas del día a día, haciendo el trabajo mucho más eficiente. Por ejemplo, en el proyecto de OCR, donde había 500 archivos que necesitaban ser abiertos manualmente, creé una aplicación que, con solo un clic, leía los archivos y extraía los IBANs de manera automática. Otro caso fue con la herramienta iLovePDF, que se utilizaba mucho en la empresa para juntar varios archivos PDF. Vi que esto generaba dependencias externas, por lo que decidí crear una pequeña aplicación en Python que realizaba la misma tarea sin tener que depender de aplicaciones o páginas web externas. Esto no solo optimizó el proceso, sino que también permitió tener todo bajo control dentro de la infraestructura interna de la empresa. Para estos proyectos, he trabajado con diferentes librerías en Python, tanto para tareas básicas como para tareas más complejas. Por ejemplo, utilicé pandas y numpy para la manipulación de datos, tesseract para la parte de OCR (reconocimiento óptico de caracteres) y tensorflow para desarrollar modelos de aprendizaje automático. Actualmente, estamos explorando proyectos de reconocimiento de lenguaje natural (NLP), para los cuales se están utilizando nuevas librerías, pero todavía no he profundizado lo suficiente en estos proyectos. Sin embargo, ya incluiré más detalles sobre ellos en la memoria del segundo cuatrimestre, ya que de momento no he tenido mucha interacción con estas tecnologías. (Ordenador de la empresa para el proyecto de los bancos:)



*Ilustración 14: Librerías Python. (Elaboración propia)*

Voy a hablar de varias de las herramientas de Microsoft que he utilizado a lo largo de las prácticas, excepto Power BI, Excel y Power Query, que ya mencioné antes. Empezaré de lo más sencillo a lo más complejo.

Primero, las listas de SharePoint. Estas listas son muy sencillas de usar: simplemente vas creando columnas y eliges el tipo de dato que quieres guardar, ya sea un número entero, un string, un objeto, una imagen, etc. Es bastante práctico para almacenar datos de forma organizada y fácil de integrar con otras aplicaciones.

La siguiente herramienta de la que voy a hablar es Visio. Visio sirve para crear flujos y diagramas de funcionamiento, lo que facilita ver de forma clara cómo funciona un código o un proceso. Es muy útil para representar procesos complejos de almacenamiento o transformación de datos.

También usamos la herramienta Planner de Microsoft 365. Con Planner, básicamente asignamos las tareas que le corresponden a cada uno del equipo, ponemos fechas de entrega, establecemos prioridades y todo lo que hace falta para organizar el trabajo. Es muy práctico para tener un seguimiento de lo que cada persona está haciendo.

Me voy a centrar en las dos herramientas principales que más he utilizado: Power Automate y PowerApps.

Power Automate es una versión no-code para crear flujos de datos y automatizar procesos repetitivos del día a día. La he utilizado, como ya he comentado, para la automatización de los correos diarios de la gestora de contrato de Iberia. Y también, lo bueno que tiene es que te incluye una especie de modelo de IA en el que, por ejemplo, si le pasas distintos PDF que tienen el mismo formato, lo puedes entrenar con 5 documentos más o menos y te puedes traer los datos si son todos bastante parecidos.

Por último, PowerApps sirve para crear aplicaciones de forma rápida, con muy poco código. En mi caso, el único código que suelo usar es HTML, que utilizo para crear documentos de salida, los cuales luego se guardan como PDF o en otros formatos.

Estas herramientas de Microsoft me han permitido agilizar muchos procesos en el departamento y facilitar la integración de datos en nuestras tareas diarias.

Ya por último, la última de las herramientas que he utilizado hasta el momento es Microsoft Fabric, que sigo utilizando en este momento. Es el proyecto más grande en el que estoy metido; cada día descubro una novedad sobre esta aplicación. Es muy interesante, ya que te permite desde almacenar el dato en bruto hasta mostrarlo en un informe, en un Excel o reportarlo directamente desde Fabric. El funcionamiento es básicamente: ingestas de datos, ya sea mediante un pipeline, un flujo de datos o una consulta SQL; esos datos, una vez almacenados en las tablas de Fabric, se pueden transformar, ya sea con query, con notebooks de PySpark, SQL, R u otros lenguajes de programación. Una vez has hecho todas las transformaciones, de momento estamos probando a generar únicamente informes, o incluso a exportarlos como Excel, por si algún departamento quiere utilizar el dato tal y como sale de Fabric. Dentro de Fabric se incluyen un montón de herramientas que todavía no he llegado a probar; por ejemplo, hay un área de análisis en tiempo real, un área de análisis de datos y un área de ingeniería de datos que, de momento, no he tocado. Yo, sobre todo, estoy con la parte de ingeniería de datos y Power BI, que es la visualización de datos, pero estoy deseando poder llegar a todo lo demás, especialmente al análisis de datos en tiempo real, que es lo que más me llama la atención.

2.9. Logros, resultados y discusión.

Uno de los mayores logros para mí ha sido confirmar mi pasión por el mundo del dato, algo que siempre me ha fascinado desde que empecé en informática. Poder estar en esta beca como analista de datos ya es, en sí mismo, un gran logro personal, y cada día me siento más convencido de que estoy donde quiero estar.

Dentro del departamento, he notado que mi trabajo se valora y eso se refleja en la confianza que han depositado en mí, por ejemplo, dejándome a mí solo montar un BBDD que se pretende que use toda la compañía. Cumplo con las fechas de entrega e incluso a veces se sorprenden con la velocidad que termino alguna tarea. Me encanta ver cómo, desde la automatización de procesos con Power Automate hasta la creación de informes complejos, puedo aportar mi granito de arena para mejorar la eficiencia de la empresa.

Algo que me ha sorprendido mucho es observar la diversidad en la forma de trabajar con los datos. Aunque todos estamos en el mismo equipo, se nota que cada persona tiene su propio estilo y sus propios métodos para trabajar y transformar los datos o los flujos de trabajo. Algunas personas son súper meticulosas y otras más espontáneas, lo que me ha enseñado a valorar la importancia de convertir esos datos en algo fiable y legible para todos. Este contraste me inspira a seguir perfeccionando mis técnicas, porque sé que de ahí nace el verdadero valor de nuestro trabajo.

Mirando hacia el futuro, me gustaría la idea de profundizar aún más en áreas innovadoras como el análisis en tiempo real. Buscando implementar nuevas tecnologías para mejorar el rendimiento del equipo, como por ejemplo he notado mucho la falta de IA en la empresa y ahorraría tiempo de tareas redundantes que se hacen. Personalmente, estoy ansioso por aprender y experimentar, y creo firmemente que, a medida que sigamos explorando estas herramientas, podremos transformar por completo la manera en la que trabajamos con la información.

Creo que con la gente que me rodea en este equipo voy a mejorar mucho, ya que son gente que apoya todas las ideas, mientras no sean ideas locas, y dejan crecer a la gente mucho estando con ellos, aparte de grandes profesionales son buena gente y eso hace que haya algo más de motivación por querer hacer cosas, porque nadie quiere ver a gente buena haciendo de más simplemente porque haya gente que no quiere trabajar.

3. DESARROLLO

3.1. Introducción

Mi proyecto se estructura en tres partes que, en conjunto, permiten obtener, transformar y visualizar datos de forma integral. La primera parte consiste en el desarrollo de un script en Python. Con este script, extraigo datos en formato XML desde una fuente pública, en este caso, la página de carburantes del gobierno español (<https://datos.gob.es/en/catalogo/e05068001-precio-de-carburantes-en-las-gasolineras-espanolas>). La motivación de utilizar Python radica en su versatilidad para automatizar tareas repetitivas y en su capacidad para manejar diferentes formatos de datos.

La segunda parte del proyecto se centra en la transformación de esos datos obtenidos. Con el script, convierto el XML en un archivo CSV personalizado. Este proceso de transformación no solo optimiza la consulta reduciendo, por ejemplo, el peso de la query en comparación con la extracción directa del XML, sino que también permite generar un histórico de datos que se adapta a las necesidades específicas del análisis. De este modo, puedo disponer de una base de datos ordenada y actualizada que recoge la evolución de los precios de los carburantes a lo largo del tiempo.

La tercera parte consiste en la creación de un informe en Power BI, donde se representan visualmente los datos procesados. Mediante diversas visualizaciones, el informe permite interpretar de manera clara y dinámica la información contenida en el CSV, facilitando la identificación de tendencias y la toma de decisiones. Este enfoque integral –desde la extracción y transformación hasta la visualización– proporciona una solución completa para el análisis de datos en un entorno real.

Esto es un xml:



*Ilustración 15: XML. (Gobierno de España, 2025)*

3.1.1. Motivación

La motivación principal detrás de este proyecto es la necesidad de optimizar el manejo de grandes volúmenes de datos que, en su formato original, no resultan fácilmente aprovechables. La posibilidad de automatizar la conversión de un XML a un CSV, junto con la generación de informes visuales en Power BI, abre la puerta a un análisis más eficiente y profundo de la información disponible. Esto no solo aporta valor a nivel técnico, sino que también contribuye a la toma de decisiones estratégicas al ofrecer datos procesados y fácilmente interpretables.

3.1.2. Análisis de mercado y necesidades

En la actualidad, la demanda de soluciones que permitan procesar y visualizar datos de manera rápida y eficaz está en aumento. Las organizaciones necesitan herramientas que conviertan datos brutos en información útil para la toma de decisiones. En este sentido, el proyecto responde a una necesidad real: transformar datos de una fuente pública en información histórica y visualmente accesible. Este enfoque permite a las empresas optimizar el consumo de recursos, reducir tiempos de respuesta y mejorar la calidad de sus análisis.

3.1.3. Objetivos

El objetivo principal es desarrollar una aplicación integral que combine tres componentes esenciales:

* **Extracción de datos:** Mediante un script en Python se extrae información en formato XML desde la web.
* **Transformación de datos:** Se convierte el XML en un CSV personalizado que facilite la creación de históricos y la manipulación de la información.
* **Visualización de datos:** Se construye un informe interactivo en Power BI que permita analizar y presentar la información de manera clara y dinámica.

3.1.4. Requisitos técnicos

Para llevar a cabo el proyecto se utilizan diversas herramientas y tecnologías:

* **Python:** Para el desarrollo del script de extracción y transformación de datos.
* **XML y CSV:** Como formatos de entrada y salida, respectivamente, facilitando la manipulación y el almacenamiento de la información.
* **Power BI:** Para la creación de informes visuales e interactivos.
* Conexión a la fuente de datos pública de la página de carburantes del gobierno, asegurando la actualización y veracidad de la información.

3.1.5. Análisis de mercado

El análisis de mercado muestra una creciente necesidad de soluciones automatizadas para el procesamiento de datos. La transformación de información bruta en datos estructurados y visualmente interpretables es fundamental para la toma de decisiones en entornos empresariales. Este proyecto se alinea con la tendencia actual, demostrando que mediante la integración de Python y Power BI es posible crear soluciones personalizadas que optimicen la extracción, transformación y visualización de datos, aportando un valor añadido a la organización.

3.2. Marco técnico

El marco técnico del proyecto se apoya en tres pilares clave: el lenguaje de programación Python, la API proporcionada por el gobierno español y las herramientas de visualización de datos, como Power BI.

* **Python:** Utilizado para el desarrollo de un script que automatiza la extracción, transformación y almacenamiento de datos. Python es una herramienta versátil y potente para el procesamiento de grandes volúmenes de datos, y en este proyecto aprovecho librerías como requests para realizar solicitudes HTTP, csv para la escritura de archivos y logging para la captura de eventos importantes durante la ejecución del script.
* **API gubernamental:** Los datos de los precios de carburantes provienen de la API pública del gobierno español (Gobierno de España, 2025). Esta API ofrece datos en formato JSON que contienen información de las estaciones de servicio y sus precios actualizados.
* **CSV (Comma-Separated Values):** Los datos extraídos de la API en formato JSON se transforman en archivos CSV, un formato accesible y ampliamente compatible para su posterior análisis.
* **Power BI:** Una vez que los datos están almacenados en CSV, se utilizan herramientas de visualización como Power BI para crear informes visuales, interactivos y dinámicos, que permiten interpretar fácilmente los datos transformados.

La arquitectura técnica sigue una lógica de pipeline, donde los datos se extraen mediante el script en Python, se procesan y almacenan en CSV, y finalmente se presentan en Power BI.

3.3. Equipo de trabajo y metodología

El proyecto fue desarrollado principalmente de forma individual, aunque siguiendo un poco las recomendaciones de Adrián de donde mejorar el trabajo y que se puede optimizar más. También he recibido un poco de ayuda de mi equipo de prácticas en algún ajuste del Power BI sobre todo para la parte de los clusters.

En cuanto a la metodología, he utilizado un enfoque **ágil**, dividiendo el proyecto en pequeñas tareas que he ido completando de forma incremental. Cada fase del desarrollo (extracción de datos, transformación, almacenamiento, visualización) la he hecho meticulosamente y revisando todos los posibles casos de error que se me iban ocurriendo, asegurando así el mejor resultado final de proyecto.

3.4. Proyecto

3.4.1. Resumen de contribuciones y productos desarrollados

Este proyecto ha resultado en el desarrollo de una solución automatizada para la extracción, transformación y análisis de datos públicos sobre los precios de carburantes en España. Las principales contribuciones incluyen:

* **Script en Python**: El script automatiza la descarga de datos en formato JSON, su transformación a CSV y la generación de un registro histórico.
* **Almacenamiento de datos**: Los datos son almacenados en dos archivos CSV, uno con un registro histórico de todas las extracciones realizadas (con una columna adicional de FechaRegistro) y otro con los datos más actuales.
* **Transformaciones Power Query**: Para transformar los datos y hacerlos más fácil de usar en el informe interactivo.
* **Informe en Power BI**: Creación de un informe interactivo en Power BI que muestra de manera clara y visual la evolución de los precios de carburantes.

3.4.2. Planificación temporal

La planificación del proyecto ha seguido el siguiente esquema:

1. **Primeros pasos:** Analizar los archivos XML y ver que es lo más optimo, si transformarlos con Power Query o transformar el archivo a uno más sencillo
2. **Segundos pasos**: Desarrollo del script de extracción de datos y conversión a CSV.

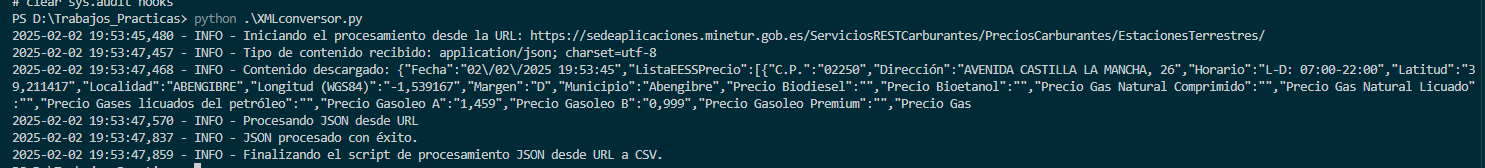
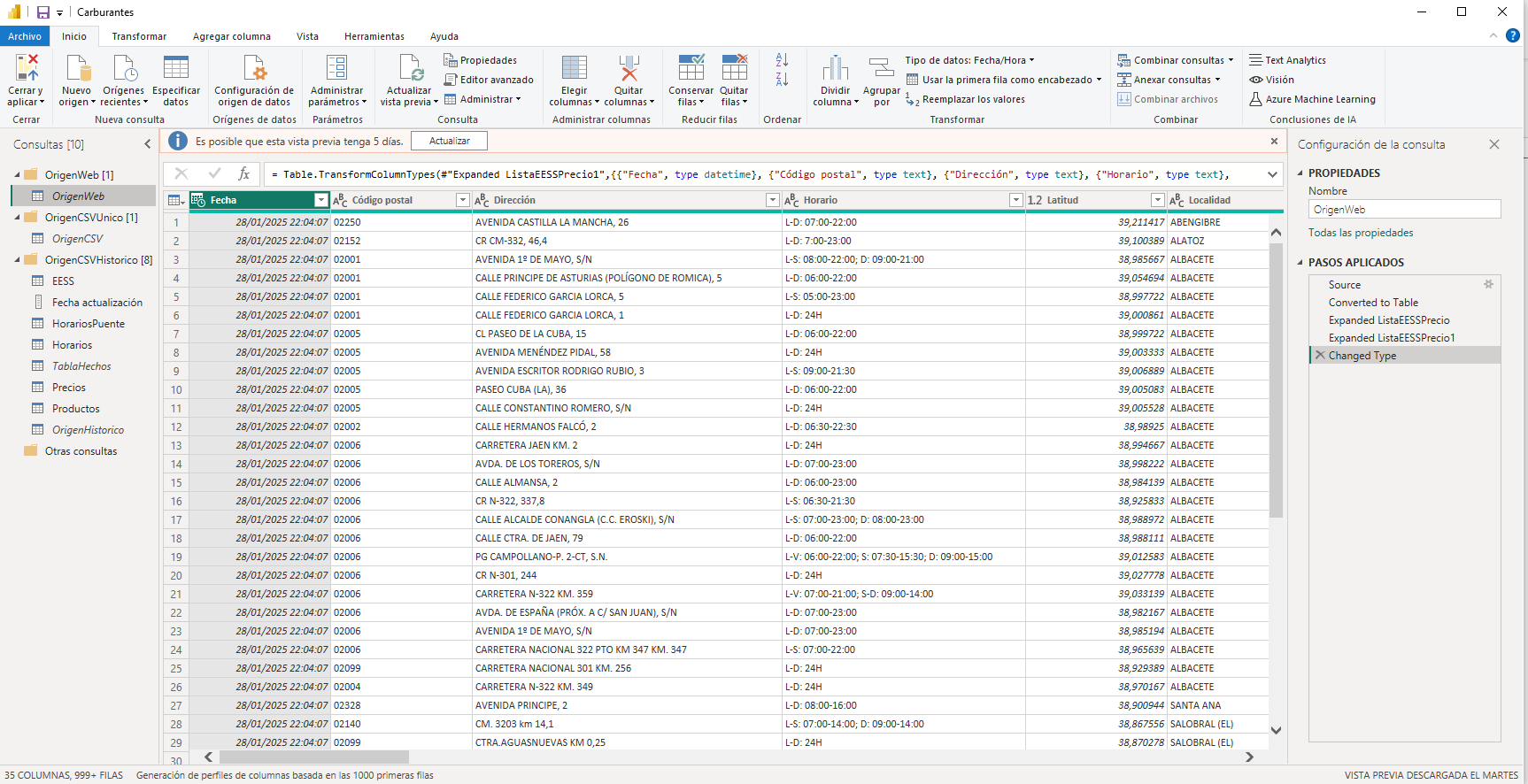
* 

Ilustración 16: Resultado terminal script. (Elaboración propia).

1. **Terceros pasos**: Optimización del script y creación de logs para el seguimiento del proceso.

* (Adjunto el código en la parte de abajo en el anexo)

1. **Cuartos pasos**: Hacer las transformaciones necesarias en Power Query demostrando la comparativa de porque desde web es menos optimo que transformando el archivo.



*Ilustración 17: Power Query. (Elaboración propia)*

1. **Quintos pasos**:Integración de Power BI para la visualización de los datos almacenados.

* (Adjunto capturas de todas las visualizaciones abajo en el anexo)

3.4.3. Recursos empleados

* **Python 3.12.6**: Para el desarrollo del script.
* **Librerías Python**: requests, csv, datetime, logging, tqdm (para el seguimiento del progreso).
* **Power BI**: Para la creación del informe visual.
* **API de precios de carburantes**: Fuente de datos en formato JSON.
* **Git**: Para el control de versiones del código.

3.4.4. Trabajo desarrollado

El trabajo principal ha sido el desarrollo del script en Python, el cual está diseñado para ejecutar automáticamente la extracción de datos desde la API del gobierno y generar archivos CSV que se pueden analizar posteriormente en Power BI.

El flujo de trabajo incluye las siguientes tareas:

1. **Descarga de datos**: El script hace una solicitud HTTP a la API, obtiene los datos en formato JSON y verifica que el contenido sea correcto.
2. **Transformación**: El JSON descargado es transformado en CSV. Se guardan dos versiones: una para el registro histórico, con la fecha de extracción, y otra para los datos más recientes.
3. **Almacenamiento**: Los datos se almacenan en archivos CSV y se generan logs para monitorear cualquier error o archivo fallido durante el proceso.
4. **Visualización**: Los CSV se importan a Power BI para la creación de informes visuales que permiten analizar los precios de carburantes en función de distintos parámetros.

3.5. Resultados y discusión

Hasta ahora, los resultados obtenidos han sido muy positivos. El script en Python funciona correctamente y logra extraer y almacenar los datos sin mayores complicaciones. La automatización de este proceso ha permitido ahorrar una cantidad significativa de tiempo, eliminando la necesidad de realizar estas tareas manualmente. Además, los informes en Power BI permiten analizar los precios de carburantes de manera visual, facilitando la interpretación de los datos para la toma de decisiones.

Una de las principales ventajas del proyecto es su escalabilidad: el script puede seguir ejecutándose periódicamente para actualizar los datos, creando un registro histórico que puede ser consultado en cualquier momento. Esto ofrece una visión en el tiempo de la evolución de los precios de carburantes, algo que no sería posible si solo se analizara la información actual.

Y como se esperaba alimentando el Power BI únicamente con el archivo histórico resultado de la ejecución del script se logra que tarde menos en actualizar los datos.

3.6. Conclusiones

Este proyecto ha demostrado que es posible integrar diferentes herramientas tecnológicas (Python, CSV, Power BI) para desarrollar soluciones completas de análisis de datos. La automatización del proceso de extracción y transformación de datos desde una API pública ha sido clave para ahorrar tiempo y mejorar la eficiencia en la gestión de grandes volúmenes de información. El uso de Power BI como herramienta de visualización ha permitido dar valor agregado al proyecto, ofreciendo una forma clara y comprensible de interpretar los datos obtenidos.

3.7. Líneas futuras

De cara al futuro, una posible mejora del proyecto sería la incorporación de análisis en tiempo real, para que los datos de carburantes sean actualizados de manera más frecuente y las visualizaciones en Power BI se sincronicen automáticamente con los datos más recientes. Además, podría explorarse el uso de técnicas de machine learning para prever la evolución de los precios de carburantes en base a históricos, lo que agregaría un componente predictivo a la solución. Este proyecto también se puede escalar para predecir precios de otro países e intentar hacerlo un proyecto internacional.

4. BIBLIOGRAFÍA.

Precio de carburantes en las gasolineras españolas

Gobierno de España. (2025).

<https://datos.gob.es/en/catalogo/e05068001-precio-de-carburantes-en-las-gasolineras-espanolas>

Sobre Serveo

Serveo

<https://serveo.com/quienes-somos/sobre-serveo>

1. ANEXOS

Anexo 1: Código de python

import tkinter as tk

from tkinter import messagebox, Toplevel, scrolledtext

import subprocess

import whisper

import text2emotion as te

from pytube import YouTube

from youtube\_transcript\_api import YouTubeTranscriptApi

import threading

import os

import sys

import matplotlib.pyplot as plt

from matplotlib.backends.backend\_tkagg import FigureCanvasTkAgg

from transformers import pipeline, AutoTokenizer, AutoModelForSequenceClassification

import nltk

from nltk.sentiment.vader import SentimentIntensityAnalyzer

from nltk.corpus import stopwords

from nltk.tokenize import word\_tokenize

import traceback

# Descargar recursos NLTK

try:

nltk.download('vader\_lexicon', *quiet*=True)

nltk.download('punkt', *quiet*=True)

nltk.download('stopwords', *quiet*=True)

analyzer\_vader = SentimentIntensityAnalyzer()

STOPWORDS = *set*(stopwords.words('english'))

except *Exception*:

print("Error al inicializar NLTK:")

traceback.print\_exc()

STOPWORDS = *set*()

# Cargar modelo RoBERTa para sentimiento y extraer etiquetas dinámicas

try:

model\_name = "cardiffnlp/twitter-roberta-base-sentiment"

tokenizer\_roberta = AutoTokenizer.from\_pretrained(model\_name)

model\_roberta = AutoModelForSequenceClassification.from\_pretrained(model\_name)

classifier\_roberta = pipeline('sentiment-analysis', *model*=model\_roberta, *tokenizer*=tokenizer\_roberta)

labels\_roberta = model\_roberta.config.id2label

except *Exception*:

print("Error al cargar modelo RoBERTa:")

traceback.print\_exc()

labels\_roberta = {0: 'NEG', 1: 'NEU', 2: 'POS'}

# === TRANSCRIPCIÓN AUTOMÁTICA YOUTUBE via API ===

*def* obtener\_transcripcion\_youtube\_api(*url*):

try:

if 'youtu.be' in url:

video\_id = url.split('/')[-1]

else:

video\_id = url.split('v=')[-1].split('&')[0]

transcript\_list = YouTubeTranscriptApi.get\_transcript(video\_id)

texto = '\n'.join(seg['text'] for seg in transcript\_list)

return texto

except *Exception*:

print("Error al obtener transcripción automática de YouTube:")

traceback.print\_exc()

raise *RuntimeError*('No existe transcripción automática para este video.')

# === DESCARGA Y TRANSCRIPCIÓN CON WHISPER ===

*def* descargar\_audio\_youtube(*url*, *nombre\_archivo*):

try:

yt = YouTube(url)

audio\_stream = yt.streams.filter(*only\_audio*=True).first()

audio\_stream.download(*filename*=nombre\_archivo)

return nombre\_archivo

except *Exception*:

print("Error al descargar audio de YouTube:")

traceback.print\_exc()

raise

*def* descargar\_audio\_twitch(*url*, *nombre\_archivo*):

try:

comando = *f*"streamlink {url} best -o {nombre\_archivo}"

subprocess.call(comando, *shell*=True)

return nombre\_archivo

except *Exception*:

print("Error al descargar audio de Twitch:")

traceback.print\_exc()

raise

*def* transcribir\_audio\_whisper(*nombre\_archivo*):

try:

modelo = whisper.load\_model('base')

resultado = modelo.transcribe(nombre\_archivo)

return resultado['text']

except *Exception*:

print("Error al transcribir audio con Whisper:")

traceback.print\_exc()

raise

# === PREPROCESAMIENTO: REMOVER STOPWORDS ===

*def* remover\_stopwords(*texto*):

try:

tokens = word\_tokenize(texto)

filtered = [t for t in tokens if t.lower() not in STOPWORDS]

return " ".join(filtered)

except *Exception*:

return texto

# === ANÁLISIS DE EMOCIONES con Text2Emotion ===

*def* analizar\_emociones\_text2emotion(*texto*):

try:

texto\_filtro = remover\_stopwords(texto)

return te.get\_emotion(texto\_filtro)

except *Exception*:

print("Error al analizar emociones con Text2Emotion:")

traceback.print\_exc()

raise

# === ANÁLISIS DE SENTIMIENTO con RoBERTa (fragmentando texto largo) ===

*def* analizar\_sentimiento\_roberta(*texto*, *max\_words\_per\_chunk*=200):

try:

texto\_filtro = remover\_stopwords(texto)

words = texto\_filtro.split()

fragmentos = [" ".join(words[i:i+max\_words\_per\_chunk]) for i in range(0, len(words), max\_words\_per\_chunk)]

resultados = []

for frag in fragmentos:

res = classifier\_roberta(frag)

for r in res:

label\_id = *int*(r['label'].split('\_')[-1])

r['label'] = labels\_roberta.get(label\_id, r['label'])

resultados.append(r)

return resultados

except *Exception*:

print("Error al analizar sentimiento con RoBERTa:")

traceback.print\_exc()

raise

# === ANÁLISIS DE SENTIMIENTO con VADER ===

*def* analizar\_sentimiento\_vader(*texto*):

try:

texto\_filtro = remover\_stopwords(texto)

return analyzer\_vader.polarity\_scores(texto\_filtro)

except *Exception*:

print("Error al analizar sentimiento con VADER:")

traceback.print\_exc()

raise

# === PROCESAMIENTO GENERAL ===

*def* procesar\_url(*url*, *usar\_youtube*, *data\_container*):

nombre\_archivo = 'audio\_temp.mp4'

try:

# 1. Obtener transcripción (API YouTube si está seleccionado y hay subtítulos disponibles)

if usar\_youtube and ('youtube.com' in url or 'youtu.be' in url):

texto = obtener\_transcripcion\_youtube\_api(url)

else:

# Descargar audio y transcribir con Whisper

if 'youtube.com' in url or 'youtu.be' in url:

descargar\_audio\_youtube(url, nombre\_archivo)

elif 'twitch.tv' in url:

descargar\_audio\_twitch(url, nombre\_archivo)

else:

raise *ValueError*("URL no válida. Solo YouTube y Twitch.")

texto = transcribir\_audio\_whisper(nombre\_archivo)

os.remove(nombre\_archivo)

# Guardar transcripción en el contenedor

data\_container['texto\_transcripcion'] = texto

# 2. Análisis Text2Emotion

data\_container['emociones\_text2emotion'] = analizar\_emociones\_text2emotion(texto)

# 3. Análisis RoBERTa (fragmentando texto largo)

data\_container['sentimiento\_roberta'] = analizar\_sentimiento\_roberta(texto)

# 4. Análisis VADER

data\_container['sentimiento\_vader'] = analizar\_sentimiento\_vader(texto)

except *Exception* as e:

print("Error en procesar\_url:")

traceback.print\_exc()

raise e

# === FUNCIONES AUXILIARES para mostrar gráficos en popups ===

*def* crear\_popup\_grafico(*figura*):

popup = Toplevel()

popup.title("Gráfico")

canvas = FigureCanvasTkAgg(figura, *master*=popup)

canvas.get\_tk\_widget().pack(*fill*=tk.BOTH, *expand*=True)

canvas.draw()

# === FUNCIONES DE INTERFAZ para mostrar resultados ===

*def* mostrar\_text2emotion(*data\_container*):

try:

if 'emociones\_text2emotion' not in data\_container:

messagebox.showinfo("Info", "Ejecuta primero el análisis.")

return

emociones = data\_container['emociones\_text2emotion']

fig, ax = plt.subplots(*figsize*=(4,2))

ax.bar(*list*(emociones.keys()), *list*(emociones.values()), *color*='skyblue')

ax.set\_title("Text2Emotion")

ax.set\_ylabel("Puntuación")

crear\_popup\_grafico(fig)

except *Exception*:

print("Error en mostrar\_text2emotion:")

traceback.print\_exc()

messagebox.showerror("Error", "Falló al mostrar resultados Text2Emotion.")

*def* mostrar\_roberta(*data\_container*):

try:

if 'sentimiento\_roberta' not in data\_container:

messagebox.showinfo("Info", "Ejecuta primero el análisis.")

return

resultados = data\_container['sentimiento\_roberta']

agr = {}

for r in resultados:

label = r['label']

agr[label] = agr.get(label, 0) + r['score']

fig, ax = plt.subplots(*figsize*=(4,2))

ax.bar(*list*(agr.keys()), *list*(agr.values()), *color*='lightgreen')

ax.set\_title("RoBERTa")

ax.set\_ylabel("Score acumulado")

crear\_popup\_grafico(fig)

except *Exception*:

print("Error en mostrar\_roberta:")

traceback.print\_exc()

messagebox.showerror("Error", "Falló al mostrar resultados RoBERTa.")

*def* mostrar\_vader(*data\_container*):

try:

if 'sentimiento\_vader' not in data\_container:

messagebox.showinfo("Info", "Ejecuta primero el análisis.")

return

vader\_score = data\_container['sentimiento\_vader']

fig, ax = plt.subplots(*figsize*=(4,2))

ax.bar(*list*(vader\_score.keys()), *list*(vader\_score.values()), *color*='salmon')

ax.set\_title("VADER")

ax.set\_ylabel("Score")

crear\_popup\_grafico(fig)

except *Exception*:

print("Error en mostrar\_vader:")

traceback.print\_exc()

messagebox.showerror("Error", "Falló al mostrar resultados VADER.")

*def* mostrar\_comparativa(*data\_container*):

try:

if not all(k in data\_container for k in ['emociones\_text2emotion','sentimiento\_roberta','sentimiento\_vader']):

messagebox.showinfo("Info", "Ejecuta primero el análisis completo.")

return

emo = data\_container['emociones\_text2emotion']

rob = data\_container['sentimiento\_roberta']

vader = data\_container['sentimiento\_vader']

# Agrupar RoBERTa

agr\_rob = {}

for r in rob:

label = r['label']

agr\_rob[label] = agr\_rob.get(label, 0) + r['score']

# Convertir Text2Emotion a Pos/Neu/Neg

pos\_t2e = emo.get('Happy',0) + emo.get('Surprise',0)

neg\_t2e = emo.get('Sad',0) + emo.get('Angry',0) + emo.get('Fear',0)

neu\_t2e = max(0, 1 - pos\_t2e - neg\_t2e)

fig, ax = plt.subplots(*figsize*=(8,4))

labels = ['Positivo','Neutro','Negativo']

t2e\_vals = [pos\_t2e, neu\_t2e, neg\_t2e]

rob\_vals = [agr\_rob.get('POS',0), agr\_rob.get('NEU',0), agr\_rob.get('NEG',0)]

vader\_vals = [vader.get('pos',0), vader.get('neu',0), vader.get('neg',0)]

x = range(len(labels))

ax.bar([p - 0.2 for p in x], t2e\_vals, *width*=0.2, *label*='Text2Emotion', *color*='skyblue')

ax.bar(x, rob\_vals, *width*=0.2, *label*='RoBERTa', *color*='lightgreen')

ax.bar([p + 0.2 for p in x], vader\_vals, *width*=0.2, *label*='VADER', *color*='salmon')

ax.set\_xticks(x)

ax.set\_xticklabels(labels)

ax.set\_ylabel('Score/ Proporción')

ax.set\_title('Comparativa de Métodos de Análisis')

ax.legend()

crear\_popup\_grafico(fig)

except *Exception*:

print("Error en mostrar\_comparativa:")

traceback.print\_exc()

messagebox.showerror("Error", "Falló al mostrar gráfica comparativa.")

# === FUNCIONES ASÍNCRONAS E INTERFAZ ===

*def* iniciar\_proceso(*entry\_url*, *var\_yt*, *lbl\_status*, *data\_container*, *cuadro\_transcripcion*):

url = entry\_url.get().strip()

if not url:

messagebox.showwarning("Atención", "Introduce una URL.")

return

lbl\_status.config(*text*="Procesando, por favor espere...")

threading.Thread(

*target*=*lambda*: run\_background(url, var\_yt.get(), lbl\_status, data\_container, cuadro\_transcripcion),

*daemon*=True

).start()

*def* run\_background(*url*, *usar\_youtube*, *lbl\_status*, *data\_container*, *cuadro\_transcripcion*):

try:

procesar\_url(url, usar\_youtube, data\_container)

# Mostrar la transcripción en el cuadro principal

texto = data\_container.get('texto\_transcripcion', '')

cuadro\_transcripcion.delete('1.0', tk.END)

cuadro\_transcripcion.insert(tk.END, texto)

lbl\_status.config(*text*="Procesamiento completado.")

except *Exception* as e:

lbl\_status.config(*text*="Error en procesamiento.")

print("Error en run\_background:")

traceback.print\_exc()

messagebox.showerror("Error", *str*(e))

*def* interfaz():

try:

ventana = tk.Tk()

ventana.title('Comparativo Sentimiento: Text2Emotion / RoBERTa / VADER')

ventana.geometry('900x850')

ventana.configure(*bg*='#f0f0f0')

# Asegurar que cerrar la ventana termine el script

ventana.protocol("WM\_DELETE\_WINDOW", *lambda*: (ventana.destroy(), sys.exit()))

tk.Label(ventana, *text*='URL YouTube o Twitch:', *bg*='#f0f0f0', *font*=('Arial',12)).pack(*pady*=5)

entry\_url = tk.Entry(ventana, *width*=90, *font*=('Arial',11))

entry\_url.pack(*pady*=3)

var\_yt\_check = tk.IntVar()

tk.Checkbutton(

ventana,

*text*='Usar subtítulos YouTube (si existen)',

*variable*=var\_yt\_check,

*bg*='#f0f0f0',

*font*=('Arial',10)

).pack()

lbl\_status = tk.Label(ventana, *text*='Esperando URL...', *bg*='#f0f0f0', *fg*='blue', *font*=('Arial',10))

lbl\_status.pack(*pady*=5)

tk.Button(

ventana,

*text*='Iniciar Análisis',

*bg*='#2196F3',

*fg*='white',

*font*=('Arial',12),

*command*=*lambda*: iniciar\_proceso(entry\_url, var\_yt\_check, lbl\_status, data\_container, cuadro\_transcripcion)

).pack(*pady*=8)

tk.Label(ventana, *text*='Transcripción Completa:', *bg*='#f0f0f0', *font*=('Arial',12,'underline')).pack(*pady*=5)

global cuadro\_transcripcion

cuadro\_transcripcion = scrolledtext.ScrolledText(ventana, *height*=10, *width*=100, *font*=('Courier',10))

cuadro\_transcripcion.pack(*pady*=3)

tk.Label(ventana, *text*='Resultados por Gráficos:', *bg*='#f0f0f0', *font*=('Arial',12,'underline')).pack(*pady*=5)

frame\_botones = tk.Frame(ventana, *bg*='#f0f0f0')

frame\_botones.pack(*pady*=5)

tk.Button(

frame\_botones,

*text*='Text2Emotion',

*bg*='#03A9F4',

*fg*='white',

*font*=('Arial',10),

*command*=*lambda*: mostrar\_text2emotion(data\_container)

).grid(*row*=0, *column*=0, *padx*=5)

tk.Button(

frame\_botones,

*text*='RoBERTa',

*bg*='#8BC34A',

*fg*='white',

*font*=('Arial',10),

*command*=*lambda*: mostrar\_roberta(data\_container)

).grid(*row*=0, *column*=1, *padx*=5)

tk.Button(

frame\_botones,

*text*='VADER',

*bg*='#FF5722',

*fg*='white',

*font*=('Arial',10),

*command*=*lambda*: mostrar\_vader(data\_container)

).grid(*row*=0, *column*=2, *padx*=5)

tk.Button(

frame\_botones,

*text*='Comparar',

*bg*='#9C27B0',

*fg*='white',

*font*=('Arial',10),

*command*=*lambda*: mostrar\_comparativa(data\_container)

).grid(*row*=0, *column*=3, *padx*=5)

ventana.mainloop()

except *Exception*:

print("Error en interfaz principal:")

traceback.print\_exc()

# Diccionario global para almacenar resultados

data\_container = {}

if \_\_name\_\_ == '\_\_main\_\_':

interfaz()