



Universidad Politécnica
de Madrid



**Escuela Técnica Superior de
Ingenieros Informáticos**

Grado en Ingeniería Informática

Trabajo Fin de Grado

**Desarrollo de una Plataforma de
Monitorización y Control en Tiempo
Real para Sistemas de Almacenamiento
y Distribución de Hidrógeno en
Empresas**

Autor: Arabela Cárceles Carrillo

Tutor: Jorge Pablo Diaz Velilla

Cotutor: Sergio José Ríos Aguilar

Madrid, Junio 2024

A mi familia, a mis descubrimientos en Madrid, en especial a Lucía ,
y a mi grupo de trabajos por excelencia de la facultad

.

Este Trabajo Fin de Grado se ha depositado en la ETSI Informáticos de la Universidad Politécnica de Madrid para su defensa.

Trabajo Fin de Grado

Grado en Ingeniería Informática

Título: Desarrollo de una Plataforma de Monitorización y Control en Tiempo Real para Sistemas de Almacenamiento y Distribución de Hidrógeno en Empresas

Junio 2024

Autor: Arabela Cárceles Carrillo

Tutor:

Jorge Pablo Diaz Velilla

Departamento de Ingeniería de Organización, Administración de Empresas y Estadística

ETSI Informáticos

Universidad Politécnica de Madrid

Cotutor:

Sergio José Ríos Aguilar

Departamento de Ingeniería de Organización, Administración de Empresas y Estadística

ETSI Informáticos

Universidad Politécnica de Madrid

Resumen

Este trabajo se ha centrado en el desarrollo de una plataforma avanzada para la monitorización y control en tiempo real de los costes asociados con el uso del hidrógeno en el contexto empresarial. El proyecto surge como respuesta a la evolución global hacia fuentes más limpias y renovables, y la necesidad por parte de las empresas de evaluar esta transición en términos de costes.

El enfoque es la creación de una herramienta dinámica y escalable que cumpla con las necesidades actuales del mercado y que también se adapte a los cambios futuros y la expansión de la tecnología de hidrógeno. Para el desarrollo, se contemplaron las limitaciones actuales del mercado, como la inmadurez de la infraestructura de hidrógeno y la volatilidad en la disponibilidad de datos. Estas condiciones guiaron el diseño de la herramienta, asegurando que sea efectiva en todo momento sin necesidad de una revisión completa.

En base a las limitaciones anteriores, la plataforma ofrece por un lado datos sobre los precios de los combustibles fósiles en las principales ciudades españolas, permitiendo el análisis y la contextualización de la evolución de estos para ofrecer una perspectiva profunda y estratégica para la toma de decisiones logísticas y operativas en las empresas. Además, se incluyen los últimos datos disponibles sobre la producción de hidrógeno y costes asociados. Esta inclusión establece la base para futuras expansiones de la herramienta, y sirve como primer paso hacia un enfoque más amplio centrado en el hidrógeno.

La implementación se ha realizado utilizando servicios de Microsoft Azure®, aprovechando su capacidad para la gestión y el almacenamiento de grandes volúmenes de datos dinámicos y su integración con Power BI para la visualización de los datos. La arquitectura de la herramienta está basada en la nube incluyendo Azure Data Factory para la extracción de los datos, y Azure Table Storage para el almacenamiento eficiente y escalable de estos.

El informe interactivo en Power BI se ha diseñado de tal forma que los gráficos y datos mostrados sean claros y comprensibles por el usuario final, de forma que puedan acceder fácilmente a información crítica. Esta herramienta incluye funcionalidades como la transformación de datos mediante Power Query y la realización de cálculos con DAX (Data Analysis Expressions). La interfaz de usuario es intuitiva, permitiendo una navegación fácil a través de gráficos y tablas que muestran tendencias y patrones.

El repositorio del proyecto se encuentra disponible en GitHub para consulta y uso público, proporcionando una valiosa oportunidad para otros desarrolladores interesados en la materia: <https://github.com/arabelacarceles/TFGInformatica>

Además, el siguiente código QR facilita el acceso directo a la plataforma. Aunque esta es funcional tanto en dispositivos móviles como en ordenadores, se recomienda que para una mejor visualización se acceda a ella desde los últimos a través del siguiente enlace: <https://short.upm.es/ndgmd>.



Abstract

This work has focused on the development of an advanced platform for real-time monitoring and control of the costs associated with the use of hydrogen in a business context. The project arises in response to the global shift towards cleaner and renewable sources, and the need for companies to evaluate this transition in terms of costs.

The approach is to create a dynamic and scalable tool that meets the current market needs and can also adapt to future changes and the expansion of hydrogen technology. For development, current market limitations such as the immaturity of the hydrogen infrastructure and the volatility in data availability were considered. These conditions guided the design of the tool, ensuring it is always effective without the need for a complete overhaul.

Based on the limitations, the platform offers data on fossil fuel prices in major Spanish cities, enabling analysis and contextualization of their evolution to provide a deep and strategic perspective for making logistical and operational decisions in companies. Additionally, the latest data available on hydrogen production and associated costs are included. This inclusion lays the groundwork for future expansions of the tool and serves as a first step towards a broader focus on hydrogen.

The implementation was conducted using Microsoft Azure® services, leveraging their capacity for managing and storing large volumes of dynamic data and their integration with Power BI for data visualization. The tool's architecture is cloud-based, including Azure Data Factory for data extraction, and Azure Table Storage for efficient and scalable storage.

The interactive informe in Power BI has been designed in such a way that the graphs and data displayed are clear and understandable to the end user, so that they can easily access critical information. This tool includes features such as data transformation using Power Query and calculations with DAX (Data Analysis Expressions). The user interface is intuitive, allowing easy navigation through graphs and tables that show trends and patterns.

The project repository is available on GitHub for consultation and public use, providing a valuable opportunity for other developers interested in the field: <https://github.com/arabelacarceles/TFGInformatica>

Additionally, the following QR code facilitates direct access to the platform. Although it is functional on both mobile devices and computers, for better visualisation it is recommended to access it from the latter through the following link: <https://short.upm.es/ndgmd>.



Tabla de contenidos

1	Introducción.....	1
2	Trabajos previos.....	3
3	Desarrollo	5
3.1	Microsoft Azure®.....	5
3.1.1	Microsoft Azure SDK.....	6
3.2	Entorno de desarrollo.....	7
3.3	Power BI.....	7
3.4	Arquitectura de la herramienta	8
3.5	Desarrollo del informe	11
3.5.1	Parte dinámica, trabajos previos	11
3.5.1.1	Suscripción en Azure	11
3.5.1.2	Grupo de recursos	13
3.5.1.3	Storage account	14
3.5.1.4	Table Storage	17
3.5.2	Parte dinámica, extracción y almacenamiento de datos	19
3.5.2.1	Azure Data Factory (ADF)	19
3.5.2.2	Servicios vinculados.....	21
3.5.2.3	Conjuntos de datos.....	21
3.5.2.4	Pipeline.....	24
3.5.2.5	Trigger	27
3.5.3	Parte estática, Excel.....	28
3.5.4	Integración y visualización de los datos en Power BI	29
3.5.5	Diseño de la herramienta.....	38
3.5.6	Optimización y Automatización de la Publicación del Informe	39
4	Limitaciones y desafíos.....	42
5	Conclusiones y resultados.....	44
6	Análisis de Impacto	45
7	Bibliografía	46
8	Anexos.....	49
8.1	Python	49
8.2	Power Query.....	55
8.3	Informe de originalidad	59

Tabla de Ilustraciones

Ilustración 1: Página principal del portal de Microsoft Azure	5
Ilustración 2: Instalación factoría de datos y punto autenticación [11]	7
Ilustración 3: Arquitectura general del proyecto.	8
Ilustración 4: Análisis de costes de los recursos asociados a la suscripción ..	12
Ilustración 5: Comando creación variable de entorno: id de la suscripción...	12
Ilustración 6: Obtención de la variable de entorno para su uso	13
Ilustración 7: Paquetes creación grupo de recursos en Python [21]	13
Ilustración 8: Control de acceso del grupo de recursos.....	14
Ilustración 9: Información general cuenta de almacenamiento del proyecto ..	15
Ilustración 10: Paquetes creación cuenta de almacenamiento en Python [24]	16
Ilustración 11: Servicios de almacenamiento vinculados	16
Ilustración 12: Creación variable de entorno: clave cuenta almacenamiento .	16
Ilustración 13: Tablas asociadas a la cuenta de almacenamiento	17
Ilustración 14: Paquetes creación de tablas de almacenamiento [25].....	17
Ilustración 15: Código para la creación de tablas de almacenamiento	18
Ilustración 16: Página para la iniciación de Azure Data Factory Studio	19
Ilustración 17: Página inicial de Azure Data Factory Studio	20
Ilustración 18: Servicios de la factoría de datos.....	20
Ilustración 19: DataSetRestBarcelona en el Data Factory Studio	22
Ilustración 20: Creación de data sets vinculados Servicio Rest.....	23
Ilustración 21: DataSetTableBarcelona en el Data Factory Studio	24
Ilustración 22: Código para la creación de las actividades de copia	26
Ilustración 23: Código para la creación del pipeline	27
Ilustración 24: Programación de las propiedades del trigger	28
Ilustración 25: Ejecuciones Trigger desde el 5 de abril hasta el 10 de abril ...	28
Ilustración 26: Importación de los datos a Power BI	30
Ilustración 27: Pantalla "Transformar datos" de Power BI	31
Ilustración 28: Pantalla final análisis diario combustibles fósiles	32
Ilustración 29: Pantalla final evolución precios combustibles fósiles	32
Ilustración 30: Transformación de los datos de la tabla "datosdeBarcelona"..	33
Ilustración 31: Obtención con DAX: precio máximo última fecha registrada..	34
Ilustración 32: Medida para el gráfico de análisis diario	35
Ilustración 33: Pantalla sobre el hidrógeno	36
Ilustración 34: Tablas finales en Power BI sobre el hidrógeno	37
Ilustración 35: Medida en DAX Produccion costes.....	38

Ilustración 36: Medida en DAX Produccion Total	38
Ilustración 37: Pantalla de inicio de la herramienta	39
Ilustración 38: Componentes tras la publicación del Informe	39
Ilustración 39: Creación enlace web.....	40
Ilustración 40: Enlace acortado herramienta.....	40
Ilustración 41: Conexiones y credenciales de los datos.....	41
Ilustración 42: Creación grupo de recursos “TFGInformatica”	49
Ilustración 43: Creación cuenta de almacenamiento "datoscombustibles"	49
Ilustración 44: Creación tablas de almacenamiento	50
Ilustración 45: Creación factoría de datos "fabricadatostfg"	50
Ilustración 46: Creación servicio Rest "ServicioRestAPIPrecioil"	51
Ilustración 47: Creación servicio "ServicioTablaAlmacenamiento"	51
Ilustración 48: Creación Datasets servicio Rest.....	52
Ilustración 49: Creación datasets servicio tablas de almacenamiento.....	53
Ilustración 50: Creación del pipeline	54
Ilustración 51: Creación del trigger	54
Ilustración 52: Transformación y filtrado de los datos importados de Azure ..	55
Ilustración 53: Limpieza y transformación datos sobre la producción [18].....	56
Ilustración 54: Obtención precio medio combustibles	56
Ilustración 55: Limpieza y transformación datos sobre los costes [17].....	57
Ilustración 56: Limpieza y transformación datos consumo	58
Ilustración 57: Informe de originalidad	59

Palabras clave

Hidrógeno

Combustibles fósiles

Energías renovables

Monitorización

Microsoft Azure

Microsoft Azure SDK

Power BI

GitHub

API

Python

Power Query

DAX

Automatización

Grupo de recursos

Cuenta de almacenamiento

Tablas de almacenamiento de Azure

Factoría de Datos de Azure

Pipeline

Trigger

Servicio REST

Excel

Informe

Interacción

Toma de decisiones

Análisis y visualización de datos

Escalabilidad

Innovación tecnológica

1 Introducción

En los últimos años, se está desarrollando por parte de la sociedad un interés en abordar los desafíos medioambientales. Esto se ha originado por la creciente preocupación a nivel mundial ante los efectos devastadores que se están observando como consecuencia del efecto invernadero. El despertar de la conciencia medioambiental ha impulsado la toma de acciones a nivel gubernamental, empresarial e individual; cuyo objetivo es la promoción del desarrollo sostenible y la reducción de los gases de efecto invernadero, en concreto aquellos derivados del uso de combustibles fósiles [1].

Estas acciones, incluyen la exploración de nuevas alternativas para que, gradualmente, las fuentes de energía limpia y renovable reemplacen a los combustibles fósiles. En este escenario, el hidrógeno se presenta como una alternativa para mitigar los impactos negativos asociados con los combustibles fósiles. Una de las principales ventajas del elemento es que es identificado como vector energético, es decir, tiene una gran capacidad para almacenar y transportar energía. Esto es fundamental para poder evolucionar hacia un sistema energético más sostenible, que aproveche fuentes de energía renovables que frecuentemente no generan energía de forma continuada.

El hidrógeno puede tener aplicación en diversos sectores industriales, pudiéndose utilizar en la industria manufacturera como materia prima para ciertos productos o en la industria del automóvil en vehículos impulsados por pilas de combustible. Sin embargo, para poder explotar el potencial que ofrece es necesario afrontar una serie de desafíos. El principal de ellos es la gran inversión que se requiere para la construcción de infraestructuras que permitan la producción, el almacenamiento y la producción de hidrógeno a gran escala. Además, hay que tener en cuenta que existe un gran coste asociado a su producción a partir de fuentes renovables de energía en comparación con los combustibles fósiles.

En función del tipo de fuente de energía que se utilice para su obtención, el hidrógeno es de un tipo u otro. Específicamente, las instituciones cuando hablan de impulsar las tecnologías de hidrógeno se refieren a hidrógeno verde; el cual se fabrica a partir de fuentes de energía renovables. Los organismos públicos subrayan la urgencia de que las empresas no solo reduzcan su huella de carbono, sino que también orienten sus decisiones estratégicas hacia la adopción de prácticas sostenibles en sus operaciones diarias, en este punto el hidrógeno verde se posiciona como una opción atractiva para las empresas [1].

Más allá de contribuir a la preservación del medio ambiente o a la reducción de las emisiones de efecto invernadero, la sustitución progresiva de combustibles fósiles por hidrógeno verde permitiría a las empresas reducir su exposición a riesgos asociados con la volatilidad de los precios del mercado energético, y estabilizar los costes. Además, la inversión en proyectos de hidrógeno impulsa la imagen corporativa de la empresa y fortalece su posición en el mercado generando también nuevas oportunidades comerciales. En un mundo cada vez más consciente del impacto ambiental, adoptar prácticas sostenibles e innovadores permite a las empresas diferenciarse [1].

En este contexto, lo que se pretende realizar es una herramienta centrada en la monitorización de los combustibles fósiles diariamente, ofreciendo una perspectiva actualizada de estos, que suponen un coste crítico en la toma de decisiones logísticas y operativas en las empresas. Además, la plataforma está adaptada para recopilar y presentar los datos más recientes disponibles sobre los costes del hidrógeno. Para su desarrollo se ha ideado un sistema de servicios alojados en la nube que ayudará con las tareas de recolección y almacenamiento de los datos. Este sistema se va a implementar utilizando la plataforma Azure de Microsoft®, que proporcionará el soporte que necesitamos para el desarrollo de los servicios necesarios. Para la visualización de los datos se utilizará Power BI, una herramienta desarrollada también por Microsoft®. A través de ella se creará el informe que permitirá a los usuarios visualizar los datos recogidos, facilitando la comprensión y el análisis.

2 Trabajos previos

Durante mi Trabajo de Fin de Grado (TFG), he elegido explorar las energías renovables, un área que me apasiona y considero crucial en el mundo actual. Mi interés en este campo se ha desarrollado a lo largo de mis estudios en administración de empresas, en los cuales tuve la oportunidad de tener mi primer contacto con el mundo de las energías renovables y algunos de los principales avances que se estaban realizando.

Además, este trabajo podría considerarse la continuación del TFG que he realizado para el Grado en Administración de Empresas. La investigación de mercado que realicé se centró en explorar las oportunidades y los desafíos que presentan la adopción de las energías renovables, concretamente el hidrógeno; como alternativa a los combustibles tradicionales.

Durante el análisis, se destaca que el hidrógeno es una pieza clave en la transición energética, pero que los altos costes de producción y la falta de infraestructura para su almacenamiento y distribución frenan su desarrollo [1]. Se ha observado una tendencia creciente en la inversión en tecnologías basadas en hidrógeno, impulsada significativamente por diversas políticas gubernamentales orientadas hacia la adopción de energías renovables. Estas políticas están acelerando el desarrollo y la adopción de soluciones de hidrógeno en el ámbito empresarial. En este contexto, el hidrógeno verde emerge como alternativa para las empresas que buscan cumplir con los estándares ambientales y posicionarse como pioneras en la transición energética [1]. El liderazgo en la implementación del hidrógeno no solo va a permitir que las empresas alcancen cierta ventaja competitiva sino también van a fortalecer su imagen y posición en el mercado [1].

A partir de todos los datos recopilados y el análisis realizado sobre las tendencias actuales y futuras en el sector de las energías renovables, concretamente en el sector del hidrógeno; decidí implementar una herramienta orientada a las empresas. El objetivo es claro, ayudar en la toma de decisiones logísticas relacionadas con el contexto, plasmando el análisis realizado junto con datos relevantes del escenario actual de precios de combustibles en España.

El enfoque inicial era la elaboración de una herramienta que permitiera a las empresas españolas monitorear y controlar sistemas de almacenamiento y distribución de hidrógeno en términos de costes. Sin embargo, debido a la inmadurez del mercado y la falta de información, no siempre es posible obtener datos actualizados y precisos sobre los costes asociados al almacenamiento y distribución del hidrógeno. Esta limitación ha guiado la evolución del proyecto hacia un enfoque ligeramente modificado que va a permitir generar un producto que aporte valor con la poca información que se tiene actualmente.

El nuevo enfoque se centra en el desarrollo de una herramienta no solo dinámica, sino escalable de forma que se pueda adaptar y expandir a medida que el mercado del hidrógeno madure y se publiquen nuevos datos. El diseño de la plataforma permite integrar nueva información a medida que esté disponible, asegurando que en sus próximas versiones pueda cumplir con el objetivo original de proporcionar un sistema de monitorización y control de los costes en los sistemas de hidrógeno.

Durante mis investigaciones previas, observé la existencia de herramientas disponibles para el público que ofrecen información actualizada diariamente sobre los precios de los combustibles en las gasolineras a lo largo del país. Aunque estos recursos son útiles, la finalidad era ir más allá de una mera difusión de datos, la idea era el desarrollo de una plataforma que presentara los datos permitiendo el análisis y la contextualización de la evolución de los precios de los combustibles para ofrecer una perspectiva más profunda y estratégica. Este enfoque pretende ayudar a las empresas a identificar patrones y prever futuros escenarios que puedan impactar en sus operaciones diarias.

Además, la plataforma integra los últimos datos disponibles sobre la producción de hidrógeno y costes asociados en España. Esta funcionalidad no proporciona actualmente el mismo tipo capacidades analíticas que el módulo dedicado al análisis de tendencias de combustibles fósiles, pero en posteriores versiones el objetivo es que sí que pueda servir para que las empresas empiecen a desarrollar estrategias de integración de energías de hidrógeno en sus operaciones cotidianas.

3 Desarrollo

En este capítulo se explicarán y detallarán todas las tareas que se han llevado a cabo para completar cada una de las fases de desarrollo del proyecto. Conforme se vaya avanzando en la explicación de las etapas, se revisará la tecnología utilizada para cada paso.

3.1 Microsoft Azure®

El proyecto está enfocado en el desarrollo de una serie de servicios que faciliten la obtención de datos dinámicos y su almacenamiento. Es importante tener en cuenta que el volumen de información a almacenar aumenta diariamente y que esta tiene una estructura concreta.

El punto de partida para empezar con el proyecto fue elegir los servicios en la nube ofrecidos por Azure. La plataforma desarrollada por Microsoft ofrece más de 200 servicios en la nube, permitiendo a los usuarios desarrollar, implementar y administrar soluciones que permitan resolver dificultades [2]. El término “nube” hace alusión al ecosistema de servidores remotos unidos a nivel mundial diseñado con el fin de almacenar y administrar datos, ejecutar aplicaciones y otra serie de servicios. Esta tecnología permite al usuario acceder a la información donde y cuando quiera siempre que tenga acceso a internet [3].

Por un lado, Azure al ser una herramienta de Microsoft facilita la integración con la herramienta escogida para la visualización de datos, Power BI; y por otro proporciona una IDE (Interfaz de Desarrollo Integrado) que permite ir depurando, comprobando en cada paso que todos los servicios programados se han creado y funcionan correctamente. La Ilustración 1, muestra como es la página principal del portal con los servicios principales que se están utilizando.

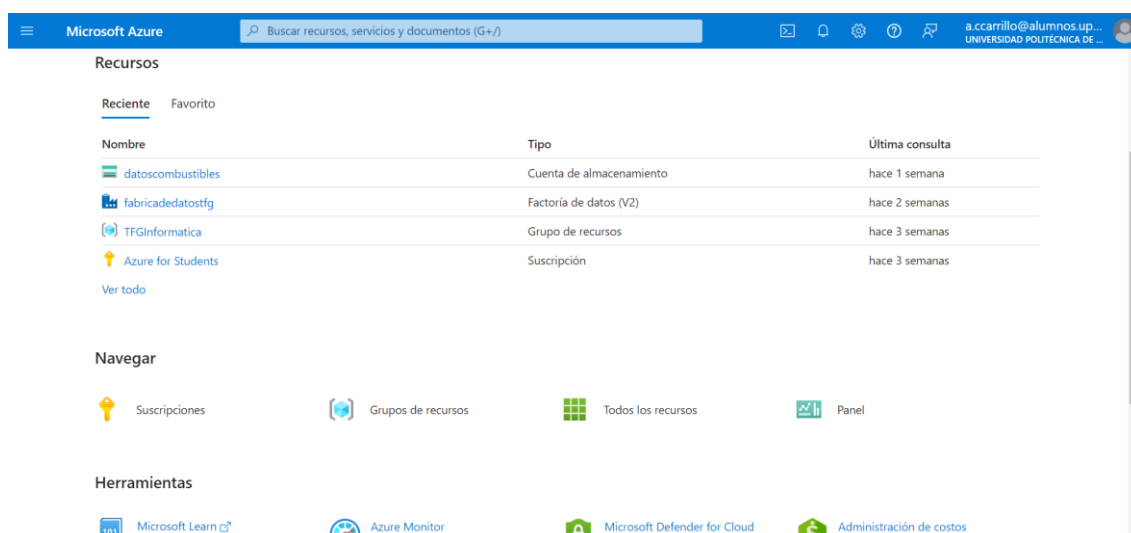


Ilustración 1: Página principal del portal de Microsoft Azure

Microsoft Azure ofrece ciertas propiedades que están alineadas con los requisitos de desarrollo:

- **Escalabilidad:** esto es crucial para manejar y controlar el crecimiento diario de los datos. La herramienta permite un ajuste sencillo de los servicios para hacer frente al aumento de demanda, asegurando almacenamiento y procesamiento eficiente de los datos [4].
- **Servicios:** como se ha mencionado la oferta de servicios es sumamente grande, pudiendo elegir aquellos que más se adapten a las necesidades del usuario [2]. En este proyecto no se exploran todas las opciones disponibles y solamente se van a explicar los servicios que hayan sido utilizados.
- **Gestión de datos:** los datos que se van a procesar tienen una estructura concreta que se pueden gestionar con varias herramientas como Azure SQL Database o Azure Table Storage. En este caso se ha elegido la segunda herramienta para la gestión de la información. Azure Table Storage forma parte de las prestaciones de Azure Storage y proporciona una solución de almacenamiento de datos NoSQL altamente disponible [5].
- **Fiabilidad y disponibilidad:** Azure ofrece unas guías de confiabilidad para que los usuarios tengan certeza de que su trabajo es resistente ante posibles malos comportamientos y que en caso de error o fallo este vuelva a funcionar [6].
- **Integración y automatización:** muchas de las prestaciones de Azure están relacionadas con los flujos de datos, obtención de datos y almacenamiento de esos datos. Además, se asegura que los datos estén almacenados eficientemente, y sean actualizados y relevantes.

Asimismo, existen otra serie de servicios que no han sido utilizados pero que hacen de Azure un instrumento muy completo. Por ejemplo, ofrece productos que usan inteligencia artificial como Azure OpenAI Service que permite crear aplicaciones de inteligencia artificial generativa [7]. Azure IoT (Internet of Things) ofrece un conjunto de servicios y herramientas que permiten recopilar, almacenar, analizar y tomar decisiones en base a grandes volúmenes de datos generados por dispositivos que se encuentran conectados entre sí [8].

3.1.1 Microsoft Azure SDK

Microsoft Azure SDK hace referencia al conjunto de herramientas que permiten la interacción con los distintos servicios de Azure. Los SDK reúnen una serie de bibliotecas que facilitan la implementación de los servicios de Azure en cualquier lenguaje de programación. Para cada uno de los lenguajes de programación disponibles, Microsoft proporciona documentación [9] que incluyen tutoriales, ejemplos de uso de código y pequeñas guías para iniciarse, facilitando al desarrollador la comprensión de la herramienta. La mayoría de los SDK son código abierto y están disponibles para todos los usuarios en repositorios de GitHub, creando una comunidad de desarrolladores muy amplia. Además, estas herramientas se acoplan perfectamente a los principales entornos de desarrollo y son mantenidos y actualizados constantemente por Microsoft.

3.2 Entorno de desarrollo

Para hacer uso de los servicios de Azure, se han utilizado las bibliotecas de los SDK en lenguaje Python [10]. Se utiliza Visual Studio Code como entorno de desarrollo porque se integra bien con las bibliotecas. Para utilizarlas es necesario por un lado iniciar sesión en Azure con la cuenta de Microsoft que se vaya a utilizar, y por otro lado se han de instalar ciertas extensiones que nos permiten acceder y administrar los servicios de Azure, y que queden reflejados estos cambios en la IDE. Es necesario el inicio de sesión pues a través de la clase *DefaultAzureCredential* nos autenticaremos en Microsoft Azure.

Para mostrar todo lo anterior, en la Ilustración 2 hay dos líneas de código. La primera muestra la instalación de los paquetes que permiten crear y administrar instancias de una factoría de datos, recurso que se va a utilizar en este proyecto. La segunda muestra cómo se obtienen las credenciales necesarias para la conexión con Azure (esto se incluye en cada una de las clases que crean un recurso), normalmente esta va seguida de la creación de un cliente específico para el servicio que se va a crear.

```
pip install azure-mgmt-datafactory

#Punto de entrada para realizar operaciones
credential = DefaultAzureCredential()
```

Ilustración 2: Instalación factoría de datos y punto autenticación [11]

Visual Studio Code ofrece una integración con Git Hub, permitiendo la colaboración en equipo y la gestión de cambios del código fuente.

3.3 Power BI

Power BI es una colección de herramientas que a partir de diferentes fuentes de datos sin relación alguna entre sí crea información coherente, interactiva y atractiva visualmente [12]. Para la visualización de los datos se eligió Power BI ya que, además de lo mencionado; ofrece numerosas herramientas para análisis y representación gráfica, permitiendo la identificación de tendencias, anomalías o patrones en los datos. Además, permite a los equipos compartir información valiosa y tomar decisiones basadas en datos de forma colaborativa. Otra prestación de Power BI, es que permite la conexión con múltiples orígenes de datos, y ofrece dos componentes esenciales para la manipulación y modelado de la información:

- **Power Query:** permite importar, limpiar, transformar y cargar datos en Power BI. Tiene su propio lenguaje para que el usuario pueda realizar consultas complejas, aunque también cuenta con una interfaz que permite tareas de manipulación sencillas sin la necesidad de escribir código [13].

El lenguaje que utiliza Power Query se denomina lenguaje M, caracterizado por hacer las operaciones paso a paso. Para la estructuración de las consultas se utilizan las palabras clave “let” e “in” para definir y controlar el flujo de ejecución [13].

- **Data Analysis Expressions (DAX):** es un lenguaje basado en fórmulas para la realización de cálculos avanzados. Es similar a las tradicionales fórmulas de Excel, pero está diseñado para trabajar específicamente con modelos de datos [14].

3.4 Arquitectura de la herramienta

En la Ilustración 3, se puede observar cual es la arquitectura general del proyecto para la recopilación de datos diarios, su transformación, y procesamiento; y su puesta en disposición para el usuario final. Los nombres usados en el diagrama son los nombres que se le ha dado a los elementos en su desarrollo.

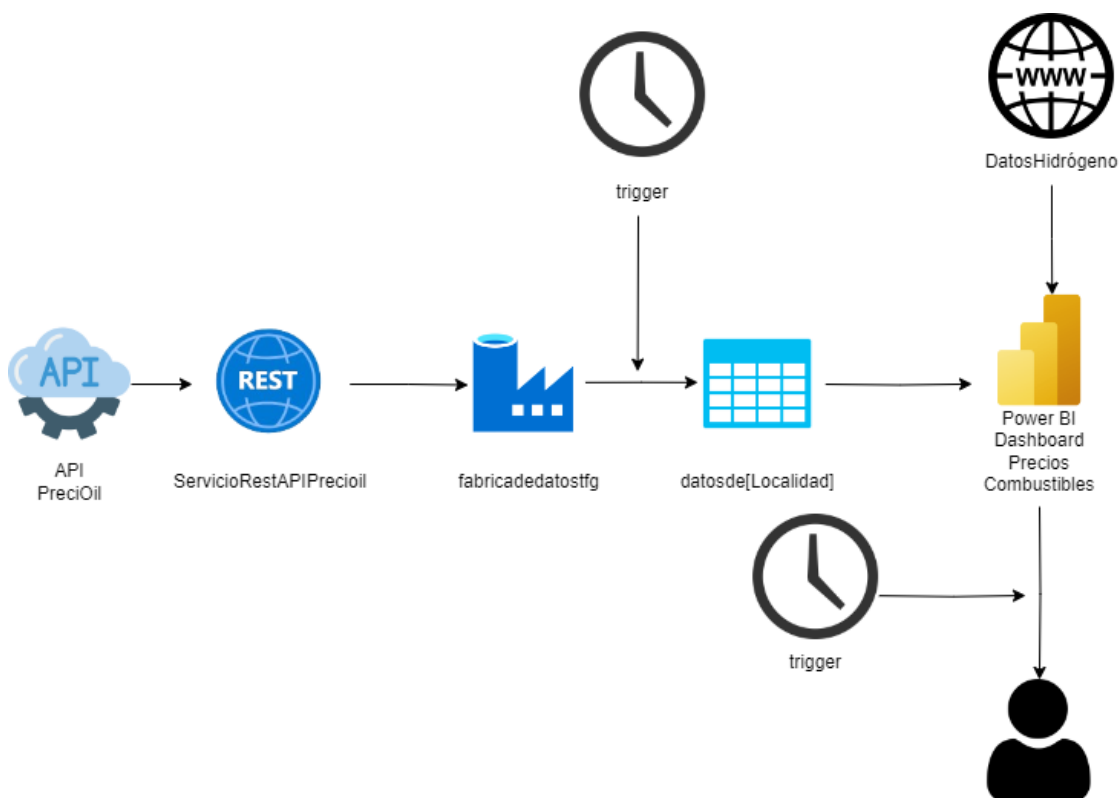


Ilustración 3: Arquitectura general del proyecto.

Todo el proceso hasta que los datos se ponen en manos del usuario final se puede resumir en diferentes etapas. Cada una de estas fases viene representada en la Ilustración 3 por iconos que caracterizan la tecnología utilizada en cada una de ellas. En secciones posteriores del capítulo se ampliará la información del proceso de desarrollo, esto es simplemente un primer vistazo del proceso de desarrollo.

1. Obtención de los datos: el informe que se ofrece al usuario presenta tanto datos dinámicos como estáticos. El origen de estos es:

- **API pública (api.precioil.es):** es el lugar del que se extraen los datos dinámicos de la herramienta. Una interfaz de programación de aplicaciones, más conocida por sus siglas en inglés como API; es un mecanismo que permite a dos componentes software comunicarse entre sí mediante un conjunto de definiciones y protocolos [15]. La API que se usa en el proyecto, obtiene los datos directamente del Ministerio para la Transición Ecológica y la DGE, los almacena en una base de datos y los pone a disposición de los usuarios. Usamos esta API ya que por medio de peticiones http ya creadas, obtenemos los datos que nos interesan, para su posterior ajuste y análisis. En este caso se ha utilizado la siguiente:

<https://api.precioil.es/estaciones/conPrecios/localidad/{localidad}>. El inicio de la URL es el nombre de la API, y es mediante los

siguientes parámetros que obtenemos todas las estaciones con los distintos combustibles que ofrecen y sus precios para una localidad específica. El nombre de la localidad se tiene que rellenar a mano en el espacio indicado por “{localidad}”. Esta URL nos va a permitir obtener datos filtrados por localidades permitiendo al usuario hacer comparaciones entre distintas ciudades.

La interfaz elegida, además del endpoint usado, ofrece otros que van desde obtener todas las estaciones con precios de los principales combustibles por municipio o el historial de precios de una estación.

- **Excel:** recolección de los datos estáticos del informe. El uso de hidrógeno como sustituto de combustibles fósiles todavía no es muy extendido, y no existen muchos datos disponibles. La información utilizada para la parte del análisis de hidrógeno proviene de unos datasets creados por European Hydrogen Observatory [16] [17] [18].

2. Recolección y almacenamiento de los datos: para obtener los datos estáticos, simplemente se necesitan importar a Power BI los datasets de European Hydrogen Observatory, y ya tendríamos los datos disponibles para su transformación. Sin embargo, para la recolección y almacenamiento de datos dinámicos el proceso es diferente. Para ello se ha hecho uso de distintos servicios ofrecidos por Microsoft Azure®. Los principales son:

- **Servicio Rest (ServicioRestAPIPrecioil):** este servicio actúa como puente entre los datos externos obtenidos de la API y la propia aplicación. El uso de este nos permite obtener datos de forma estructurada y eficiente. Se emplean métodos RESTful para la realización de consultas específicas, para esta aplicación las consultas son genéricas y lo único que cambia es la localidad buscada.

- **Factoría de datos (fabricadedatostfg):** es la parte más importante de toda la arquitectura de desarrollo. Sirve para coordinar y automatizar las tareas de recopilación, transformación y almacenamiento de datos. Mas adelante se explicarán detalladamente todos los servicios que usa la factoría, siendo uno de ellos el *Servicio Rest* explicado anteriormente. Mediante el uso de pipelines, triggers y otras actividades que ofrece se garantiza que los datos extraídos mantienen su integridad durante todo el proceso [19].
- **Tablas de almacenamiento de Azure (datosde[Localidad]):** tal y cómo se ha mencionado, este tipo de servicios ofrecen una solución No-SQL diseñada para almacenar una gran cantidad de datos estructurados, ofreciendo un acceso rápido y escalable a ellos [5]. Estas tablas se conectan a la factoría de datos y van guardando los datos que se recopilan diariamente. Dadas las características del proyecto y teniendo en cuenta que el volumen de datos es creciente a lo largo del tiempo, el uso de tablas de almacenamiento nos permite mantener los datos de forma segura y acceder a ellos con facilidad. Además, se pueden configurar las tablas de forma que únicamente se guarden los datos necesarios y con la estructura adecuada para su posterior análisis.

3. Visualización y transformación de los datos: retomando lo explicado en secciones anteriores, Power BI es la herramienta que vamos a utilizar para la transformación y filtrado de sus datos; y donde se creará la imagen que se quiere dar al usuario. Al trabajar con servicios de Microsoft, la integración y acoplamiento de los servicios de Azure con Power BI es bastante sencillo. Para la transformación y filtrado de los datos se hace uso de Power Query y de DAX. Ambas herramientas son lenguajes característicos de Microsoft, el segundo concretamente de Power BI; que permiten la transformación de los datos para su posterior visualización. Por la estructura original de los datos, en este proyecto no se ha profundizado mucho en todas las posibilidades que ambas herramientas ofrecen.

Después de haber organizado los datos de manera adecuada, se ha desarrollado un informe interactivo que permite al usuario filtrar los datos según sus requisitos específicos y visualizarlos de forma intuitiva. Este informe permite una exploración flexible y una comprensión clara de los datos, facilitando así la toma de decisiones informadas.

4. Automatización: una de las funcionalidades que el informe ofrece es la visualización de datos tomados en tiempo real. Para asegurar que la información es dinámica, se utilizan dos mecanismos que desencadenan actualizaciones diarias. Por un lado, para la obtención de datos se ha desarrollado un desencadenador o trigger, que llama diariamente al servicio Rest de Azure. Para mantener el Power BI actualizado y que se muestren los nuevos datos diarios, se programa una actualización del informe final, media hora después de que se hayan recopilado los datos para garantizar que efectivamente se han recogido los datos del día y se han cargado y transformado correctamente.

3.5 Desarrollo del informe

En el apartado anterior se han dado las primeras pinceladas de la arquitectura del informe dinámico. En esta sección se van a explorar más a fondo cada una de las fases, añadiendo detalles de pasos intermedios no explicados anteriormente pero igual de necesarios para la creación de la plataforma.

Para un mayor entendimiento se va a separar el desarrollo de la parte dinámica de la parte estática del cuadro de mando. Además, se va a hacer una distinción dentro de todo el desarrollo de la parte dinámica entre el desarrollo de la extracción y almacenamiento de datos y los trabajos previos necesarios. Todos los nombres de los servicios siguen las normas establecidas por Azure, las cuales no son iguales para todos los tipos. Todo el código asociado con la implementación de la parte dinámica se puede encontrar en el Anexo 8.1 o en el siguiente enlace: <https://github.com/arabelacarceles/TFGInformatica>.

Por último, se explicarán todas las actividades realizadas en Power BI para obtener el informe final. Se detallarán las consultas realizadas en Power Query y en el lenguaje DAX para el filtrado y la transformación de los datos.

3.5.1 Parte dinámica, trabajos previos

En esta sección se van a explicar todos los servicios de Microsoft Azure que han sido utilizados y que son necesarios para poder comenzar a extraer los datos. El orden en el que se encuentran no es arbitrario; sino que es el orden en el que se ha desarrollado el código y se ha ejecutado para conseguir una correcta implementación de todos ellos.

3.5.1.1 Suscripción en Azure

Para la realización de la parte dinámica se van a utilizar los servicios de Azure. Es importante tener en cuenta que para poder utilizarlos es necesario activar una suscripción que se adecue a las necesidades del proyecto. En este caso, se ha utilizado la suscripción ‘Azure for Students’, vinculada al correo electrónico institucional de la universidad; que ofrece un saldo anual gratuito para adquirir recursos en la plataforma.

Una vez se ha activado, en la IDE de Azure podemos acceder a ella y consultar datos relativos a costes, velocidad y previsión de los gastos y un registro de las operaciones que se han llevado a cabo junto con otros datos relevantes. La Ilustración 4 muestra los costes que se habían incurrido en el proyecto hasta el día 3 de abril de 2024. En la imagen, se puede observar una barra lateral que muestra todas las posibilidades de gestión y administración que ofrece Azure.

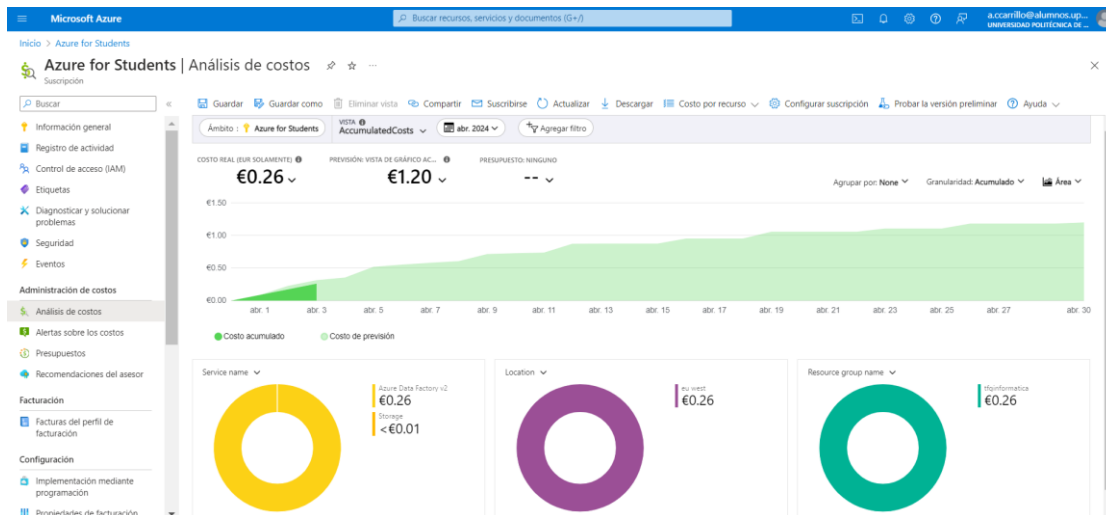


Ilustración 4: Análisis de costes de los recursos asociados a la suscripción

Antes de comenzar a crear recursos en Azure, es fundamental identificar el ID de la suscripción. No es recomendable compartir esta identificación, y ya que el código va a ser parte de un repositorio público de GitHub; se ha configurado una variable de entorno para almacenarla de forma segura y utilizarla cuando sea necesario. Para establecer el ID de la suscripción como una variable de entorno en el sistema operativo Windows, basta con ejecutar el comando de la Ilustración 5 en la línea de comandos sustituyendo “id_de_suscripcion” por el ID de la cuenta:

```
setx ID_AZURE_SUSCRPTION "id_de_suscripcion"
```

Ilustración 5: Comando creación variable de entorno: id de la suscripción

Una vez guardada cada vez que sea vaya a usar hay que tener en cuenta que para poder recuperarla es necesario importar el módulo de Python correspondiente para acceder a los métodos. Las líneas de código de la Ilustración 6 muestran cómo se ha realizado la importación del paquete y las llamadas necesarias para guardar la variable de entorno en una variable local y facilitar su uso. Además, se incluye código para el manejo de errores por si no se hubiese establecido correctamente la variable de entorno.

```
import os #para poder obtener la variable de entorno
# Autenticación para Azure, cogemos la id de la suscripción de una
variable de entorno que ha sido previamente creada
subscription_id = os.getenv('ID_AZURE_SUSCRPTION')
#Comprobamos que exista una variable de entorno con el id de la
suscripción
if not subscription_id:
    raise ValueError("Por favor, establece la variable de entorno
ID_AZURE_SUSCRPTION antes de ejecutar este script.")
```

Ilustración 6: Obtención de la variable de entorno para su uso

3.5.1.2 Grupo de recursos

Un grupo de recursos es una especie de contenedor que agrupa todos los servicios de Azure relacionados con una solución. El crearlo es una práctica esencial. Por un lado, la agrupación lógica de los recursos relacionados facilita la gestión y el seguimiento de estos. Además, todos los elementos se pueden implementar, actualizar o eliminar conjuntamente simplificando tareas de gestión del ciclo de vida. Por otro lado, existen políticas de acceso basadas en roles permitiendo asignar permisos a usuarios o grupos concretos sin tener que asignar permisos a cada uno de los recursos individualmente. Finalmente, tener todo agrupado en un mismo contenedor facilita las tareas de monitorización y diagnóstico [20].

Azure SDK proporciona una serie de bibliotecas que permiten la creación de un grupo de recursos mediante Python. Para ello, primero es necesario tener instalados tres paquetes de Python en el entorno de desarrollo por lo que se ejecuta el código de la Ilustración 7 para empezar a hacer uso de ellos [20].

```
pip install azure-identity azure-mgmt-resource azure-mgmt-storage
```

Ilustración 7: Paquetes creación grupo de recursos en Python [21]

El código de creación de este recurso está en la Ilustración 42 del Anexo 8.1. Para comprobar que se ha creado correctamente, se puede acudir nuevamente al portal de Azure y en la página principal aparece junto a la suscripción el nuevo grupo de recursos creado (ver Ilustración 1). Para este proyecto el nombre que se ha utilizado ha sido *TFGInformatica*, y si accedemos en el portal a él vamos a obtener información relevante sobre el contenedor, y sobre todos los recursos asociados a él. La Ilustración 8 presenta la pantalla para la adjudicación de roles de la que se ha hablado anteriormente. Actualmente, solo existe un usuario que tiene acceso a él, pero podrían añadirse más.

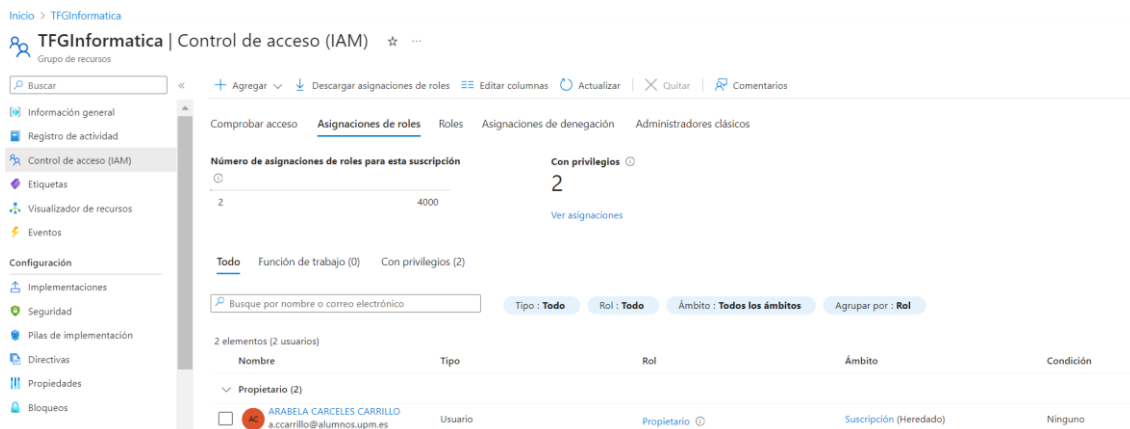


Ilustración 8: Control de acceso del grupo de recursos

3.5.1.3 Storage account

Una cuenta de almacenamiento de Azure es un servicio de almacenamiento en la nube desarrollado por Microsoft Azure que proporciona espacio para almacenar sus datos al que se puede acceder desde cualquier parte del mundo. Actúa como un contenedor para agrupar varios servicios de almacenamiento de Azure, incluidos: Azure Blobs (para almacenamiento de objetos), Azure Files (para almacenamiento de archivos compartidos), Azure Queues (para almacenamiento de mensajes en cola) y Azure Tables (para almacenamiento de datos NoSQL) [22].

Las cuentas de almacenamiento permiten acceder a sus datos desde cualquier lugar, garantizando disponibilidad y redundancia. Además, proporciona funciones de seguridad, como control de acceso basado en roles, autenticación con Azure Active Directory, cifrado de datos, y redes virtuales para un aislamiento seguro. Las cuentas de almacenamiento poseen características similares a otros servicios en cuestiones de escalabilidad y rendimiento; permitiendo el manejo de grandes cantidades de datos garantizando su consistencia y coherencia [22].

Azure ofrece distintos tipos de cuentas de almacenamiento para que el usuario escoja la que más se ajuste a sus necesidades:

- **Cuentas de uso general v2 (GPv2):** son las cuentas más conservadoras y ofrecen el mejor rendimiento y el rango más amplio. Admiten todos los servicios de almacenamiento disponibles, como Blobs, Files, Queues y Tables. Además, estas cuentas tienen funciones avanzadas como almacenamiento, replicación geográfica y acceso regular a datos [22].
- **Cuentas genéricas v1 (GPv1):** Estas son las primeras versiones de cuentas GPv2. Aunque admiten los mismos servicios de datos (Blob, Files, Queues, Tables), carecen de algunas de las nuevas características y optimizaciones de costos que se encuentran en GPv2 [22].
- **Cuentas Blob Storage:** están optimizadas para almacenar datos redundantes como blobs (objetos). Proporcionan acceso a almacenamiento de blobs de nivel activo, inactivo y de archivo, pero no admiten otros servicios como archivos, colas o tablas [22].

- **Cuentas Block Blob Storage:** estas cuentas están diseñadas para situaciones de alto rendimiento que requieren un acceso rápido y consistente a los datos. Estos solo admiten el almacenamiento de blobs y están optimizados para operaciones de blobs en bloques, pero no admiten páginas ni blobs anexados [22].
- **Cuentas de almacenamiento de archivos:** utilizadas solo con Azure Data Services, estas cuentas proporcionan funciones de almacenamiento de archivos, como IOPS más altas y mejor rendimiento en comparación con el almacenamiento estándar [22].

En la Ilustración 9 se puede observar información general de la cuenta de almacenamiento del proyecto denominada *datoscombustibles*, y se recalca con un rectángulo azul que el tipo de cuenta utilizada es de uso general v2.

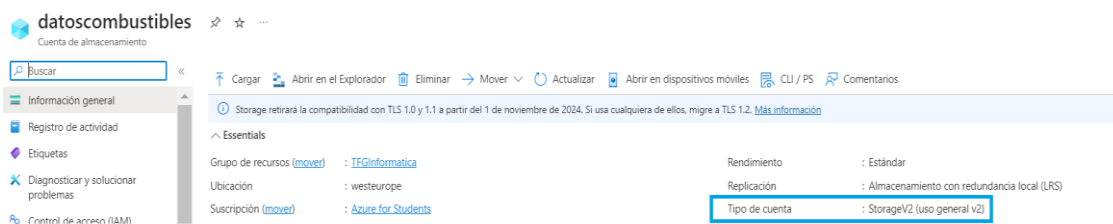


Ilustración 9: Información general cuenta de almacenamiento del proyecto

Se ha elegido este tipo de cuentas de almacenamiento por su integración con los servicios REST, pipelines y tablas que también se utilizan en el proyecto.

- Integración con **servicios REST:** GPv2 admite el acceso mediante API REST, lo que permite una comunicación eficiente y segura con servicios y aplicaciones que utilizan arquitectura REST para intercambiar datos. Como ya se había introducido, para la recopilación de datos en tiempo real de la API se utiliza este servicio que a su vez permite una interacción fluida con el servicio de almacenamiento que en este caso son tablas.
- Uso de Azure **Pipelines:** Azure Data Factory puede usar canalizaciones para hacer fluir datos desde varios orígenes a una única cuenta de almacenamiento. Las cuentas GPv2 admiten una amplia gama de servicios de Azure, lo que permite que las canalizaciones procesen y muevan datos entre múltiples plataformas y almacenamiento, incluidos blobs, archivos, columnas y tablas, lo que permite una organización de datos fácil y eficiente [23].
- Almacenamiento e implementación de **tablas:** las cuentas GPv2 permiten la integración con tablas para almacenar una gran cantidad de datos no relacionales, ideal para administrar datos recopilados por los servicios REST y procesados por las Pipelines. Además, gracias al funcionamiento de las cuentas GPv2, el acceso y gestión de esta información es fácil, permitiendo consultas eficientes y organización de información útil [5] [22].

Para poder crear la cuenta de almacenamiento mediante los recursos SDK, es necesario únicamente instalar el paquete relacionado tal y cómo se indica en la Ilustración 10, ya que el otro paquete que se usa es `azure-identity` que ya se instaló previamente para la creación del grupo de recursos.

```
pip install azure-mgmt-storage
```

Ilustración 10: Paquetes creación cuenta de almacenamiento en Python [24]

Se utiliza la IDE para comprobar que una vez se ha ejecutado el código correspondiente a este recurso (ver Ilustración 43 del Anexo), este se ha creado tal y cómo debería. Una de las prestaciones ofrecidas es la posibilidad de gestionar todos los servicios de almacenamiento de datos que están vinculados con la cuenta. En la Ilustración 11, vemos que a la cuenta `datoscombustibles` están vinculadas un total de 23 tablas.

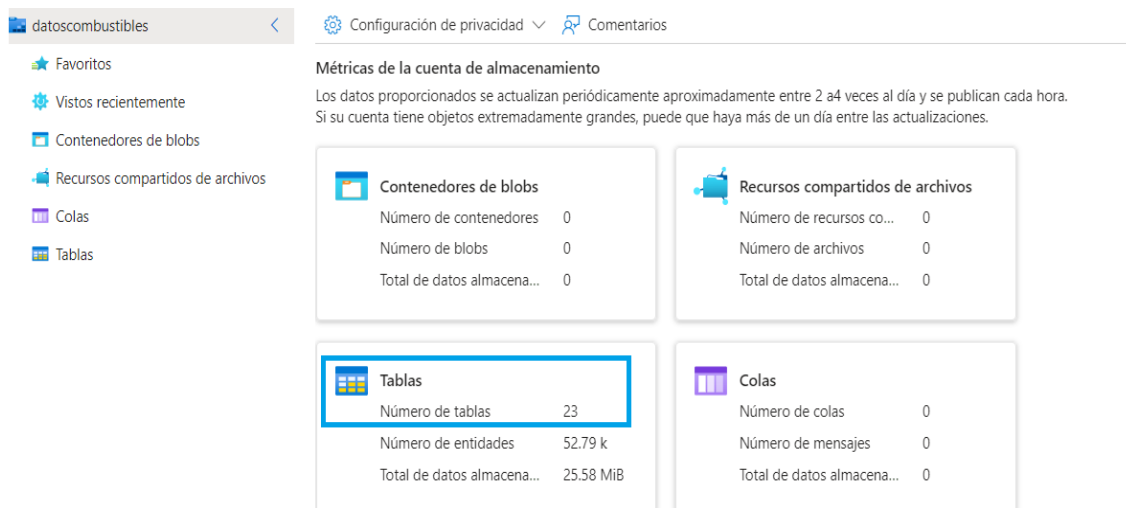


Ilustración 11: Servicios de almacenamiento vinculados

Cada cuenta de almacenamiento creada genera dos claves de acceso que son privadas y que se utilizan para la creación de recursos asociadas a la cuenta. Es por esto por lo que, al igual que se hizo con el id de la suscripción; para mantener la confidencialidad y seguridad se crea una variable de entorno con una de estas claves (Ilustración 12).

```
setx AZURE_STORAGE_ACCOUNT_KEY "clave_acceso_cuenta"
```

Ilustración 12: Creación variable de entorno: clave cuenta almacenamiento

Para acceder a la variable se ejecuta un código similar al mostrado en la Ilustración 6.

3.5.1.4 Table Storage

Por su nombre en inglés Table Storage, una tabla de almacenamiento es el recurso de almacenamiento que se ha elegido para almacenar los datos recogidos. Aunque no se vayan a utilizar hasta una vez se han recogido los datos, por su vinculación con la cuenta de almacenamiento se decidió crearlas nada más tener una cuenta a la que asociarlas.

Las tablas de almacenamiento permiten guardar una gran cantidad de datos estructurados, que no necesariamente requieren de uniones complejas, claves externas o procedimientos almacenados [5]. La principal ventaja de este tipo de servicio de almacenamiento es su fácil uso, y su compatibilidad con SDK. Asimismo, las tablas de almacenamiento aseguran de forma automática que en función de la demanda los recursos estén en todo momento disponibles y ofrece la opción de realizar consultas eficientes similares a consultas SQL sin la complejidad inherente que tienen estas últimas [5].

Cómo ya se ha comentado, hay un total de 23 tablas de almacenamiento asociadas a la cuenta. En la Ilustración 13 se puede ver que el nombre de las tablas sigue la siguiente estructura: *datosde[Localidad]*, indicando así que en cada una de ellas se almacenan los datos de la ciudad correspondiente.

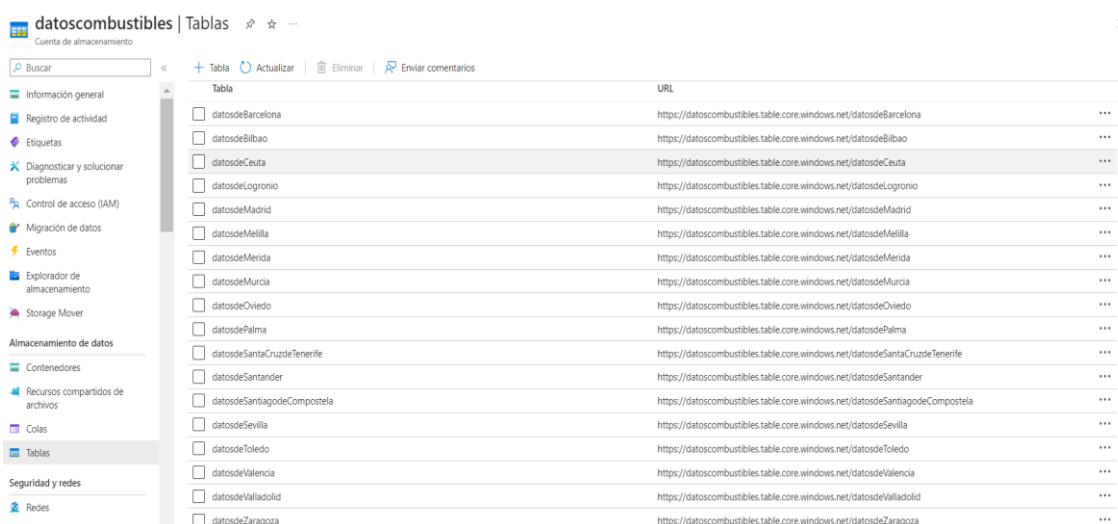


Tabla	URL
<input type="checkbox"/> datosdeBarcelona	https://datoscombustibles.table.core.windows.net/datosdeBarcelona
<input type="checkbox"/> datosdeBilbao	https://datoscombustibles.table.core.windows.net/datosdeBilbao
<input checked="" type="checkbox"/> datosdeCeuta	https://datoscombustibles.table.core.windows.net/datosdeCeuta
<input type="checkbox"/> datosdeLogronio	https://datoscombustibles.table.core.windows.net/datosdeLogronio
<input type="checkbox"/> datosdeMadrid	https://datoscombustibles.table.core.windows.net/datosdeMadrid
<input type="checkbox"/> datosdeMelilla	https://datoscombustibles.table.core.windows.net/datosdeMelilla
<input type="checkbox"/> datosdeMerida	https://datoscombustibles.table.core.windows.net/datosdeMerida
<input type="checkbox"/> datosdeMurcia	https://datoscombustibles.table.core.windows.net/datosdeMurcia
<input type="checkbox"/> datosdeOviedo	https://datoscombustibles.table.core.windows.net/datosdeOviedo
<input type="checkbox"/> datosdePalma	https://datoscombustibles.table.core.windows.net/datosdePalma
<input type="checkbox"/> datosdeSantaCruzdeTenerife	https://datoscombustibles.table.core.windows.net/datosdeSantaCruzdeTenerife
<input type="checkbox"/> datosdeSantander	https://datoscombustibles.table.core.windows.net/datosdeSantander
<input type="checkbox"/> datosdeSantiagoCompostela	https://datoscombustibles.table.core.windows.net/datosdeSantiagoCompostela
<input type="checkbox"/> datosdeSevilla	https://datoscombustibles.table.core.windows.net/datosdeSevilla
<input type="checkbox"/> datosdeToledo	https://datoscombustibles.table.core.windows.net/datosdeToledo
<input type="checkbox"/> datosdeValencia	https://datoscombustibles.table.core.windows.net/datosdeValencia
<input type="checkbox"/> datosdeValladolid	https://datoscombustibles.table.core.windows.net/datosdeValladolid
<input type="checkbox"/> datosdeZaragoza	https://datoscombustibles.table.core.windows.net/datosdeZaragoza

Ilustración 13: Tablas asociadas a la cuenta de almacenamiento

Para la creación de cada una de las tablas, primero hay que instalar el paquete correspondiente siguiendo el comando que aparece en la Ilustración 14.

```
pip install azure-data-tables
```

Ilustración 14: Paquetes creación de tablas de almacenamiento [25]

En la Ilustración 15, se ha adjuntado parte del código para la creación de tablas de almacenamiento (ver código completo en la Ilustración 44 del Anexo).

Del paquete instalado antes se hace uso de *TableServiceClient* que va a permitir interactuar con las tablas dentro de la cuenta de almacenamiento. A partir del módulo se guarda en la variable *table_service* una conexión segura con la cuenta para posteriormente crear la tabla. La cadena de conexión está formada por el tipo de protocolo de conexión utilizado para conectarse a la cuenta de almacenamiento, el nombre y clave de la cuenta, y el sufijo de la conexión, en este caso el que se utiliza de forma estandarizada para establecer la conexión [25].

Para la creación de una tabla por localidad se almacenan en una lista todos los nombres, teniendo en cuenta que no puede haber espacios ni caracteres como la ñ. Esta lista se recorre, y para cada elemento se crea una tabla diferente.

```
from azure.data.tables import TableServiceClient
connection_string=f"DefaultEndpointsProtocol=https;
    AccountName=datoscombustibles;
    AccountKey=AZURE_STORAGE_ACCOUNT_KEY;
    EndpointSuffix=core.windows.net"

table_service=
TableServiceClient.from_connection_string(conn_str=connection_string)

localidades=
['Murcia','Madrid','Sevilla','Zaragoza','Oviedo','Palma','SantaCruz
deTenerife',    'Toledo',    'Santander','Valladolid',    'Barcelona',
'Merida', 'SantiagodeCompostela', 'Logronio', 'Valencia', 'Bilbao',
'Ceuta', 'Melilla']
for i in localidades:
    #Configuracion de la tabla
    table_name = "datosde"+i
    # Creación de la tabla
    table_client=
    table_service.create_table_if_not_exists(table_name=table_name)
    #Print para asegurarnos que se ha creado la tabla
    print(f"Tabla '{table_name}' creada.")
```

Ilustración 15: Código para la creación de tablas de almacenamiento

3.5.2 Parte dinámica, extracción y almacenamiento de datos

Hasta ahora se han revisado todos los pasos necesarios para poder empezar a crear recursos y servicios que se encarguen de la extracción y almacenamiento diaria de datos. Igual que en la sección anterior, el orden en el que se van a ir explicando los servicios es crucial y relevante para una correcta implementación.

3.5.2.1 Azure Data Factory (ADF)

Las factorías de datos más comúnmente conocidas como data factories, permiten la integración de datos en la nube mediante procesos de ETL (extraer, transformar y cargar). En este proyecto la factoría supone la base donde se van a alojar todos los recursos que están encargados de la extracción dinámica de los datos. A través de canalizaciones, los datos se van a recoger de la API, se van a transformar y se van a guardar en conjuntos de datos [19].

Como ya se ha hecho anteriormente para cada uno de los servicios es necesario instalar los paquetes necesarios antes de implementar la factoría de datos en Python (ver Ilustración 2). Las factorías de datos se vinculan a una cuenta de almacenamiento, sino que están directamente relacionadas con el grupo de recursos. El código correspondiente a la creación de la factoría nombrada como *fabricadedatostfg* está contenido en la Ilustración 45 del Anexo 8.1.

Azure ofrece una IDE (Azure Data Factory Studio), independiente del portal que se ha visto en secciones anteriores para la gestión de los recursos; y se inicia directamente desde el portal de Azure (Ilustración 16).

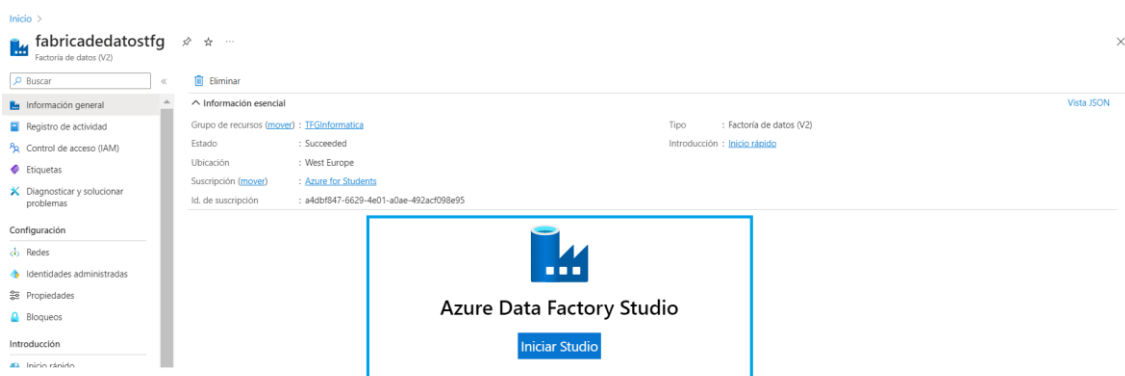


Ilustración 16: Página para la iniciación de Azure Data Factory Studio

Una vez se ha iniciado el estudio desde el panel lateral que se observa en la Ilustración 17 se van a gestionar todos los servicios asociados a la fábrica de datos y que van a permitir guardar en las tablas de almacenamiento toda la información que se busca.

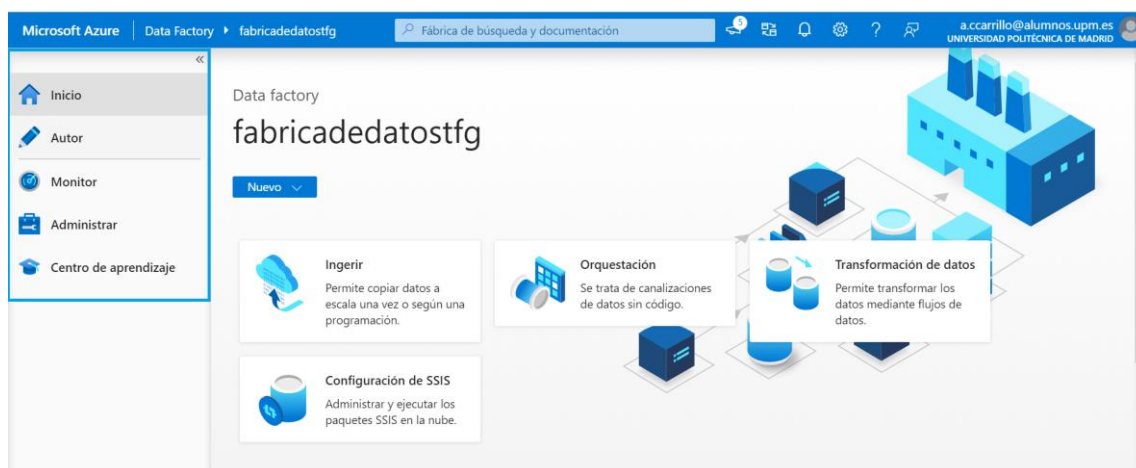


Ilustración 17: Página inicial de Azure Data Factory Studio

Antes de pasar a ver cada uno de los recursos que se van a implementar dentro de la factoría de datos, es conveniente explicar que son cada uno de ellos, las relaciones existentes y su función dentro de la factoría de datos. Para una mayor comprensión en la Ilustración 18 hay un esquema de las relaciones.

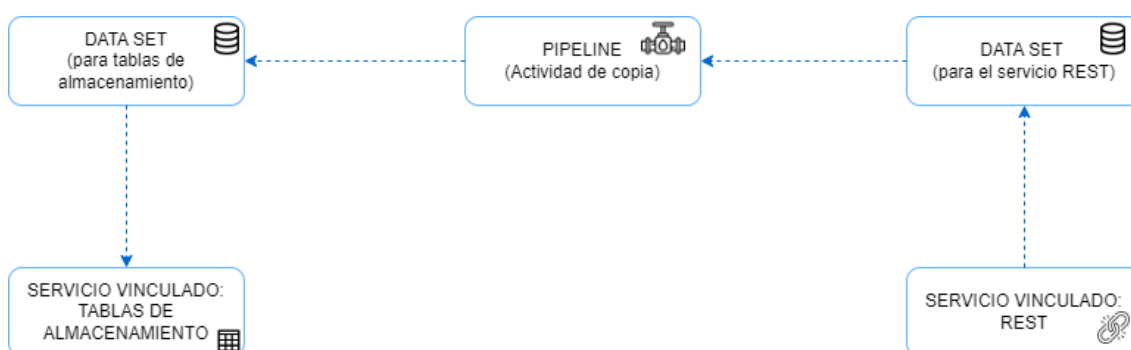


Ilustración 18: Servicios de la factoría de datos

Por un lado, se observa que existen dos servicios vinculados, cada uno de estos servicios va a servir para crear una conexión entre un recurso externo y la factoría de datos. En este caso, tenemos un servicio vinculado de un servicio Rest que sirve para conectar la factoría con la API seleccionada. Por otro lado, el servicio vinculado de tablas de almacenamiento nos permite conectar la factoría con las tablas de almacenamiento de Azure que tenemos creadas y donde se van a guardar los datos.

Los servicios vinculados están asociados a conjuntos de datos, que representan la información en la factoría. Es necesario tener un data set para definir la estructura de los datos de la API y otro para almacenar los datos extraídos. En el centro del esquema está la canalización o pipeline que representa la lógica de las actividades. La actividad principal es la de copia, que obtiene datos de la API usando el servicio vinculado Rest, los procesa, los transforma y los almacena en tablas de almacenamiento mediante el servicio vinculado correspondiente. El almacenamiento en las tablas es incremental, es decir, se mantienen todos los datos recopilados sin reemplazarlos.

3.5.2.2 Servicios vinculados

3.5.2.2.1 Servicio vinculado: Servicio Rest

Es el recurso que va a permitir a la factoría de datos conectarse y extraer datos cuya fuente es un servicio web. A partir de él, la ADF hace llamadas a la API seleccionada a través de procesos de autenticación para leer o escribir datos.

En la Ilustración 46 del Anexo se muestran el código completo para la creación del servicio Rest. Para la creación del servicio Rest únicamente necesitamos especificar la url y el tipo de autenticación. Como se trata de una API pública no es necesario credenciales para entrar por lo que el tipo es “*Anonymous*”. Por otro lado, la url es genérica para todas las localidades y únicamente cuando se vaya a acceder a la API se especificará la ciudad de la cual queremos extraer los datos. Por lo que en este caso la url simplemente sería la que está guardada en la variable *rest_service_url*. El nombre del servicio vinculado es *ServicioRestAPIPrecioil*, y se almacena en su variable correspondiente.

3.5.2.2.2 Servicio vinculado: Almacenamiento de tablas

Con este recurso todas los datos extraídos y procesados por la factoría de datos van a ser almacenados en las tablas de almacenamiento que han sido creadas anteriormente. Algunos aspectos que destacar del código de la Ilustración 47 para la creación del recurso son:

- El protocolo que se utiliza para la creación del recurso es *https*.
- Se necesita un grupo de recursos, una cuenta de almacenamiento y una factoría de datos para poder crear el servicio.
- El nombre del servicio vinculado es *ServicioTablaAlmacenamiento*

3.5.2.3 Conjuntos de datos

3.5.2.3.1 Conjunto de datos: Servicio Rest

Hace referencia a los datos que se van a usar como entrada de la actividad de copia de la canalización. Una de las principales diferencias con el servicio vinculado al que se encuentra unido, es necesario crear un conjunto de datos distinto para cada una de las localidades de las que se quiere obtener información tal y cómo se muestra en la Ilustración 19. En la Ilustración 20 se muestra como se ha manejado este escenario, para una mayor comprensión de algunas variables se recomienda ir a la Ilustración 48 del Anexo. Los nombres de los conjuntos van a tener la siguiente estructura: *DataSetRest[localidad]*.

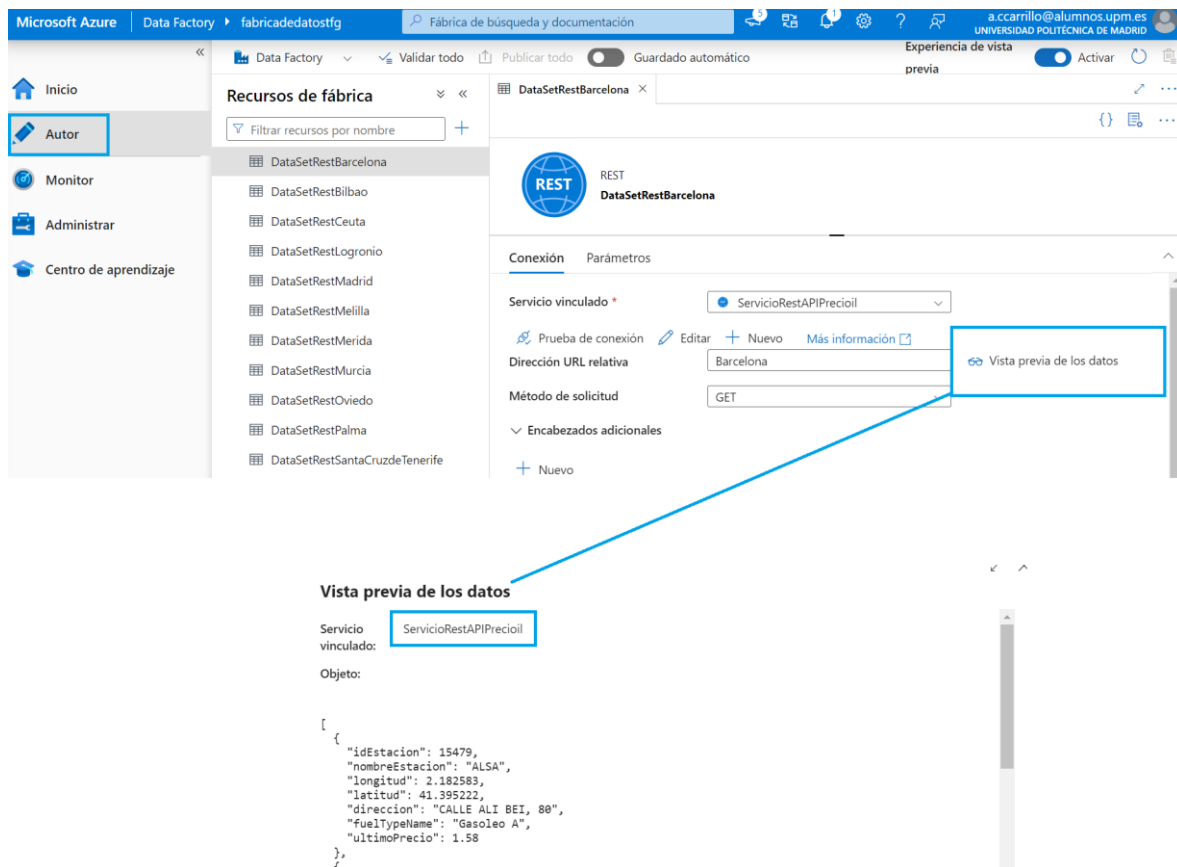


Ilustración 19: DataSetRestBarcelona en el Data Factory Studio

Para cada uno de los conjuntos de datos se parte de la url base que ya se había establecido en el Servicio Rest al que están vinculados. Pero para acceder a los datos concretos de cada una de las ciudades hay que añadir una url relativa que es el nombre de la ciudad en sí. Tal y cómo se hizo para las tablas de almacenamiento se parte de una lista con todas las localidades que se va a ir recorriendo y creando un data set por cada elemento. Según las normas de nombrado de los conjuntos de datos no pueden existir espacios en los nombres ni caracteres como la ‘ñ’, por lo que se ha tenido que hacer una gestión de nombres para que la url relativa sea la correcta y que al mismo tiempo no haya problemas con los nombres de los conjuntos de datos. Esta gestión se hace en el bloque de *ifs* anterior a la creación del dataset en sí. Los únicos nombres que resultan conflictivos son los de las localidades de Santa Cruz de Tenerife, Santiago de Compostela y Logroño.

Desde el estudio de la factoría de datos en la pestaña de autor accedemos a los recursos de la fábrica y podemos comprobar que los conjuntos de datos correctamente y a qué servicio vinculado están unidos. Además, una de las prestaciones de más utilidad de la herramienta es que proporciona una vista previa de cómo se importan los datos y con qué estructura desde la API (ver Ilustración 19).

```

localidades=['Murcia','Madrid','Sevilla','Zaragoza','Oviedo','Palma','
SantaCruzdeTenerife','Toledo',                'Santander','Valladolid',
'Barcelona','Merida','SantiagodeCompostela','Logronio',    'Valencia',
'Bilbao','Ceuta', 'Melilla']
for i in localidades:
    # URL relativa (puedes cambiar esto según sea necesario)
    if i == 'SantaCruzdeTenerife':
        relative_url='Santa Cruz de Tenerife'
    elif i == 'SantiagodeCompostela':
        relative_url='Santiago de Compostela'
    elif i == 'Logronio':
        relative_url='Logroño'
    else:
        relative_url = i
# Crear el dataset para el servicio REST
rest_dataset_name = 'DataSetRest'+i
rest_dataset = RestResourceDataset(
    linked_service_name=LinkedServiceReference(
        reference_name=linked_service_name,
        type='LinkedServiceReference'),
    relative_url=relative_url,
)
adf_client.datasets.create_or_update(
    resource_group_name=resource_group_name,
    factory_name=data_factory_name,
    dataset_name=rest_dataset_name,
    dataset=DatasetResource(properties=rest_dataset)
)

```

Ilustración 20: Creación de data sets vinculados Servicio Rest

3.5.2.3.2 Conjunto de datos: Almacenamiento de tablas

Son los datos que se producen como salida de las actividades de copia de la canalización. Al igual que en los data sets del Servicio Rest es necesario tener un data set para cada localidad. En este caso, ya que la información es salida y no está conectada directamente a la API, no hay que hacer ninguna gestión de nombres y simplemente se va recorriendo la lista de las localidades y se van creando conjuntos de datos para cada elemento. Además, cuando se crea el data set hay que indicar cual es el nombre de la tabla de almacenamiento donde se van a alojar los datos (ver Ilustración 49 del Anexo). Por este motivo la lista de localidades tiene que ser igual para todos los recursos que trabajen con ellas pues la gestión de nombres es muy importante para que todos los recursos queden vinculados correctamente. Los conjuntos de datos tienen la siguiente estructura en su nombre: *DataSetTable[localidad]*.

En el estudio al igual que antes también se puede ver que se hayan creado correctamente todos los conjuntos de datos. Como se observa en la Ilustración 21 el conjunto de datos tiene como servicio vinculado *ServicioTablaAlmacenamiento*, y la tabla a la que hace referencia es la correspondiente al dataset.

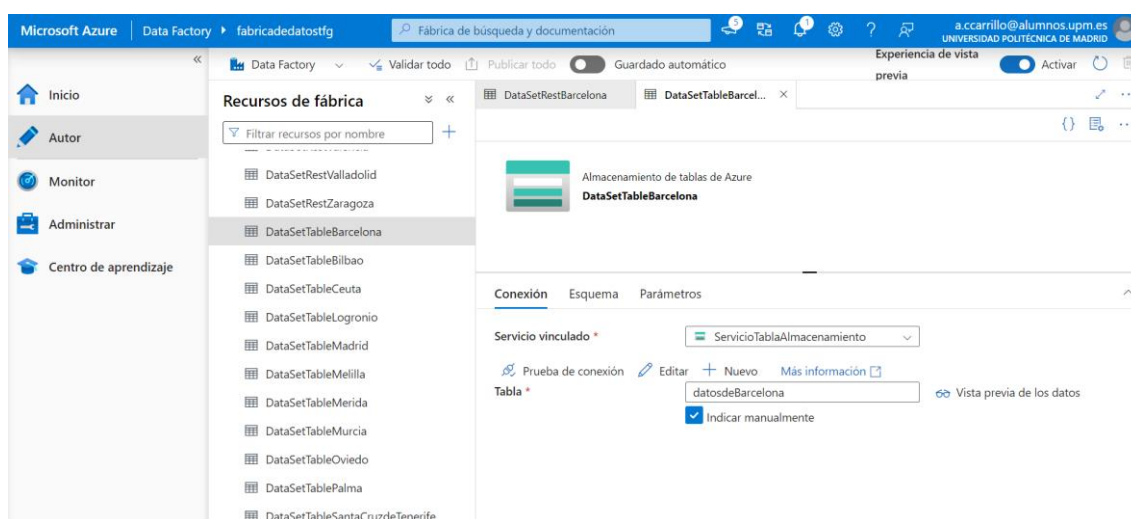


Ilustración 21:DataSetTableBarcelona en el Data Factory Studio

3.5.2.4 Pipeline

La canalización o pipeline es un conjunto de actividades que se encargan de realizar una tarea. En concreto, en este caso las actividades a realizar son de copia, las cuales transfieren datos desde una fuente (la API) hacia un destino (tablas de almacenamiento de Azure. Se habla de actividades en plural pues debe existir una actividad que transfiera los datos para cada provincia.

Siguiendo con el ejemplo de Barcelona que se ha mostrado en las Ilustraciones 19 y 21. Tenemos un *DataSetRestBarcelona* que será la fuente de la *ActividadDeCopiaBarcelona*, la cual redirigirá los datos hacia el *DataSetTableBarcelona*.

Para una mayor comprensión del código adjuntado en la Ilustración 50 del Anexo, se han separado en las Ilustraciones 22 y 23 la creación de las actividades de copia y la creación de la canalización respectivamente.

Para las actividades de copia (Ilustración 22) como en casos anteriores se parte de la lista localidades que contiene todos los nombres de las ciudades que se van a incluir en el proyecto y que son genéricos para todos los recursos creados. Además de esta lista, se crea también una lista vacía en la que se van a ir almacenando cada una de las actividades con sus propiedades respectivas para posteriormente crear una canalización para todas ellas. Como se ha mencionado antes estos procesos tienen un origen y un destino, por lo que antes de crearlos es necesarios identificarlos. El sistema de nombres que se ha implementado hace posible que para cada vuelta del bucle que va recorriendo la lista de provincias, se pueda identificar el data set del servicio Rest y el data set de las tablas de almacenamiento para cada una.

Una vez identificados, se crea la actividad en sí. Los campos dentro del método *CopyActivity* corresponden a [26]:

- Name: nombre que va a identificar la actividad. Siguiendo un poco el esquema de nombres que se viene dando hasta ahora los nombres van a tener la siguiente estructura: *ActividadDeCopia[localidad]*
- Inputs: hace referencia a los conjuntos de datos de entrada. Definen la estructura de la información.
- Outputs: conjunto de datos de salida, donde se van a alojar los datos después de la operación.
- Source: detalles para la conexión y la lectura de los datos. En este caso se pone como source también el conjunto de datos del servicio Rest pues este tiene toda la información necesaria para actuar también como origen.
- Sink: que configuración es necesaria para escribir los datos en el destino. Aquí también se establece como sumidero el conjunto de datos de tablas de almacenamiento.
- Translator: sirve para mapear y transformar los datos desde el origen hasta el destino. Si se deseara se puede especificar como se van a mapear los nombres de las columnas o se podría cambiar el tipo de los datos [27].

Tras configurar la actividad de copia, se añade a la lista de actividades mediante el método *append()*.

```

actividades_copia=[]
localidades=['Murcia','Madrid','Sevilla','Zaragoza','Oviedo','Palma',
'SantaCruzdeTenerife','Toledo','Santander','Valladolid',
'Barcelona','Merida','SantiagodeCompostela','Logronio','Valencia',
'Bilbao','Ceuta','Melilla']
for i in localidades:
    rest_dataset_name = 'DataSetRest'+i # Nombre del dataset REST
    table_storage_dataset_name = 'DataSetTable'+i # Nombre del
dataset de Table Storage
# Definición de la actividad de copia
copy_activity = CopyActivity(
    name='ActividadDeCopia'+i,

    inputs=[DatasetReference(reference_name=rest_dataset_name,
type='DatasetReference')],

    outputs=[DatasetReference(reference_name=table_storage_dat
aset_name,type='DatasetReference')],

    source=(DatasetReference(reference_name=rest_dataset_name,
type='DatasetReference')),

    sink=(DatasetReference(reference_name=table_storage_datase
t_name,type='DatasetReference')),
    translator=TabularTranslator()
)

actividades_copia.append(copy_activity)

```

Ilustración 22: Código para la creación de las actividades de copia

Para la creación de la canalización (Ilustración 23) primero establecemos en el método *PipelineResource*, que las actividades que se van a realizar son todas las que han sido creadas y almacenadas en la lista *actividades_copia*, y posteriormente se procede a la creación de la canalización en sí, con el nombre *Pipeline*.

```
# Creación del pipeline
pipeline = PipelineResource(
    activities=actividades_copia,
)

pipeline_name = 'Pipeline'
adf_client.pipelines.create_or_update(
    resource_group_name=resource_group,
    factory_name=data_factory_name,
    pipeline_name=pipeline_name,
    pipeline=pipeline
)
```

Ilustración 23: Código para la creación del pipeline

3.5.2.5 Trigger

Para automatizar la extracción diaria de datos es necesario programar un desencadenador o trigger que ejecute la actividad de copia de los datos diariamente. En la Ilustración 24 se especifican las propiedades del desencadenador, el resto del código está adjuntado en la Ilustración 51 del Anexo. Este recurso al tratarse de un trigger no basta solo con crearlo, sino que es necesario iniciarlo para que comience su actividad. El nombre de este recurso se ha establecido como *Trigger*.

```
recurrence = ScheduleTriggerRecurrence(
    frequency='Hour',
    interval=24, # Ejecutar una vez al día
    start_time='2024-03-01T08:00:00Z', # Hora y fecha de inicio en
    formato UTC
    end_time='2024-07-01T09:00:00Z', #Hora y fecha de fin en formato
    UTC
    time_zone='Romance Standard Time' #Hora de Madrid
)
```

Ilustración 24: Programación de las propiedades del trigger

En el portal de Azure donde se gestiona la factoría de datos sin necesidad de iniciar el estudio podemos ver las ejecuciones del *Trigger* y si estas han sido satisfactorias o no. En la Ilustración 25 se ilustran algunas ejecuciones del desencadenador.

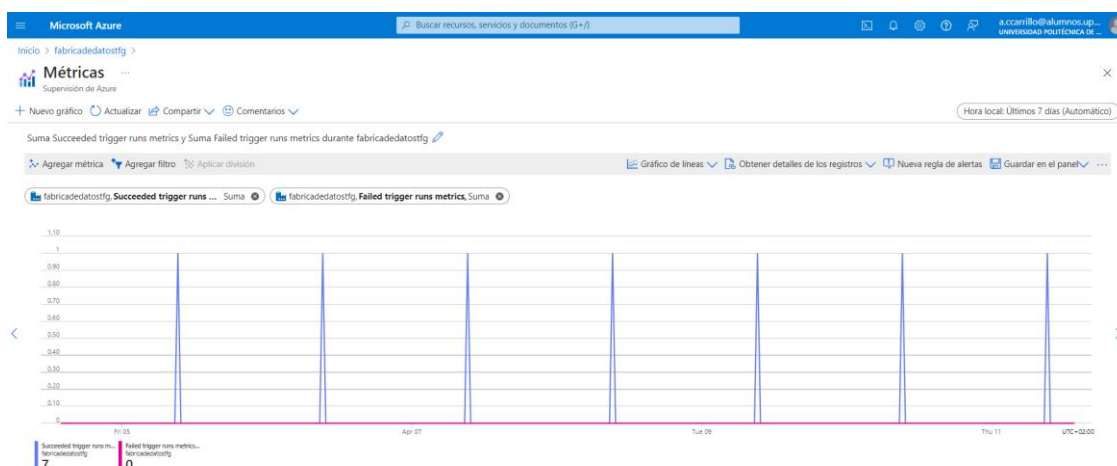


Ilustración 25: Ejecuciones Trigger desde el 5 de abril hasta el 10 de abril

3.5.3 Parte estática, Excel

La parte estática del trabajo corresponde a la recopilación de datos sobre el hidrógeno. Como ya se ha comentado antes, debido a la inmadurez del mercado existe una escasez de datos que impiden hacer un informe dinámico y detallado sobre este. Ante esto se ha recurrido a la información proporcionada por la institución European Hydrogen Observatory. En 2023 publicó datos relacionados con la producción, costes y demanda de hidrógeno en Europa correspondientes al año 2022 [17] [16] [18]. Estos datos presentan una fuente crítica de información que permite una primera aproximación al estado actual del mercado de hidrógeno en España.

Aunque actualmente la información es principalmente estática ya que no se actualiza periódicamente, el objetivo a largo plazo es transformar esta funcionalidad de la plataforma hacia un modelo más dinámico que pueda actualizar y expandir su alcance a medida que se vayan publicando más datos y el mercado del hidrógeno madure. El objetivo es continuar con la monitorización para mejorar la capacidad de las entidades en el desarrollo de estrategias entorno al uso del hidrógeno en sus operaciones diarias.

3.5.4 Integración y visualización de los datos en Power BI

En esta sección se va a explicar el proceso de transformación de los datos en brutos en información valiosa, para su visualización. Primero se explicarán los procesos de importación de datos de ambas fuentes existentes, y las técnicas de modelado empleadas. Para la definición de relaciones entre los conjuntos de datos o la creación de medidas se ha utilizado Power Query y DAX.

Uno de los factores que hacen muy atractivo el uso de Power BI es la amplia gama de opciones de conexión de datos que integra. En este proyecto se utiliza esta funcionalidad para optimizar la gestión y el análisis de datos de las dos fuentes con las que se está trabajando, Azure y Excel. En la Ilustración 26 se muestra el proceso de importación de datos.

Por un lado, tenemos el proceso de conexión al Almacenamiento de Tablas de Azure desde Power BI. A través de la interfaz de la herramienta se observan las múltiples opciones de Azure que pueden conectarse a Power BI, demostrando la compatibilidad de la plataforma con los servicios en la nube. Una vez se selecciona el tipo de recurso de Azure al que se quiere conectar, en este caso Almacenamiento de Tablas; es necesario añadir las credenciales necesarias. Cuando se establece la conexión se puede visualizar en el panel de Navegador todas las tablas existentes en él, y se seleccionan todas ellas.

Del mismo modo se conecta Power BI con la web European Hydrogen Observatory. Para cada uno de los datasets en formato Excel que se buscan importar, se obtiene su URL. Power BI reconoce las hojas de cada uno de los archivos, y se seleccionan aquellas que son de interés para el análisis.



Ilustración 26: Importación de los datos a Power BI

Una vez importados todos los datos, la función "Transformar datos" de la plataforma abrirá una nueva ventana, como se muestra en la Ilustración 27, donde todas las fuentes de datos importadas aparecerán ya cargadas y preparadas para el modelado. El proceso de "Transformar datos" permite limpiar y preparar los datos, para asegurar que el modelo final de los datos sea robusto y confiable. Los datos se importan en función de cómo se recopilan y su estructura original.

En la Ilustración 27, se muestra el estado de la información de la tabla de Azure “datosdeBarcelona” nada más cargarse a Power BI. Aparentemente no habría valor en los datos importados, por lo que es necesario transformarlos para generarlo.

Consultas [20] | Origen([Name="datosdeBarcelona"])[Data]

	PartitionKey	RowKey	Timestamp	Content
1	DefaultPartitionKey	000915c1-576f-4c4c-bd26-64536e370d77	31/03/2024 7:00:23	Record
2	DefaultPartitionKey	001c22e3-8406-4583-8ac7-4fcb88ba9484	11/05/2024 7:00:14	Record
3	DefaultPartitionKey	001ecc28-036a-4064-bc5a-b7d70539f299	31/03/2024 7:00:23	Record
4	DefaultPartitionKey	0025fa95-6a3f-469f-b1a9-78b33ec81761	05/04/2024 7:00:29	Record
5	DefaultPartitionKey	0031fe94-bc31-49b1-b9dc-258d7bbacfee	22/03/2024 9:01:09	Record
6	DefaultPartitionKey	003c5389-b333-4d11-ada5-8c78865cae5e	06/04/2024 7:00:30	Record
7	DefaultPartitionKey	003f801b-8203-4d43-9012-3ee46fcef161	14/04/2024 7:00:25	Record
8	DefaultPartitionKey	004fc3bb-a6ed-4080-9256-8868ef6e5ec0	22/03/2024 9:01:09	Record
9	DefaultPartitionKey	00506c24-71ad-413c-900d-476f5189528e	28/03/2024 7:00:22	Record
10	DefaultPartitionKey	00589a15-92a9-464d-85cb-ebbcba5f5ea9	30/03/2024 7:00:27	Record
11	DefaultPartitionKey	00610d40-8413-4f15-a83f-23fec6118fe9	28/03/2024 7:00:22	Record
12	DefaultPartitionKey	006508ab-e656-4dc2-8402-fad8f3ed0fc0	30/04/2024 7:00:50	Record
13	DefaultPartitionKey	00664d93-a956-4e2a-9d0c-a146f0523149	01/04/2024 7:00:16	Record
14	DefaultPartitionKey	006beba7-601e-46f8-a4c7-20a23c3431c7	06/05/2024 7:00:27	Record
15	DefaultPartitionKey	006cd014-718a-46cb-a00c-d573db318549	13/05/2024 7:00:21	Record
16	DefaultPartitionKey	007724f6-3a2a-4163-ae29-5f9d4b30a18b	10/04/2024 7:00:20	Record
17	DefaultPartitionKey	00779f35-026c-4bea-8c93-bc3baac728f9	24/03/2024 7:00:21	Record
18	DefaultPartitionKey	007cd843-4035-4d4a-b8ed-9ac85771d749	14/05/2024 7:00:22	Record
19	DefaultPartitionKey	00856277-afa0-4591-a7e2-bb829cde469d	05/04/2024 7:00:29	Record
20	DefaultPartitionKey	0086c997-b2a3-4f0e-9486-c89c94804ae3	27/04/2024 7:00:23	Record
21	DefaultPartitionKey	00920027-fdeb-4564-9983-4c11721ade2f	23/04/2024 7:00:25	Record

4 COLUMNAS, 999+ FILAS | Generación de perfiles de columnas basada en las 1000 primeras filas | VISTA PREVIA DESCARGADA A LAS 15:00

Ilustración 27: Pantalla "Transformar datos" de Power BI

Una de las partes esenciales es la estructuración y planificación del producto final que se desea realizar. Este paso de diseño es crucial, pues el modelado de los datos se tiene que alinear con los objetivos del análisis. Es por ello, que antes del modelado de datos es fundamental comprender no solo los datos que se tienen, sino en qué contexto se van a utilizar y cómo se puede facilitar la interpretación de ellos. Al hacerlo, se puede diseñar una estructura de modelo de datos que facilite análisis efectivos. Todo esto se traduce en una selección y organización adecuada de las tablas de datos, las relaciones entre ellas y que cálculos y agregaciones han de realizarse para que se reflejen las necesidades del usuario.

Cómo ya se ha mencionado, la plataforma se divide en dos partes principales. La primera parte abarca los combustibles fósiles, y constituye la sección dinámica de la plataforma. Con esta, se pretende que el usuario pueda observar tendencias a lo largo del tiempo y entre las diferentes regiones seleccionadas. Para una mejor visualización este análisis se divide en dos: un análisis diario y una evolución a lo largo del tiempo.

- **Análisis diario:** se muestran los últimos datos recopilados con los precios de los combustibles. Específicamente se presentan los precios medios de cada uno de los combustibles, así como el precio máximo y el precio mínimo registrado especificando en qué provincia y para qué tipo de combustible. Además, se podrá hacer una pequeña comparativa entre diferentes regiones y en función del tipo de combustible (Ilustración 28).

- **Evolución:** en esta parte se busca que el usuario pueda sacar conclusiones, extraer patrones, evaluar tendencias y tomar decisiones en función de cómo han ido variando los precios a lo largo del tiempo (Ilustración 29).

Al separar estos dos análisis, se puede por un lado entender el estado actual de los precios y evaluar tendencias y patrones, lo cual es esencial para la planificación y toma de decisiones estratégicas.

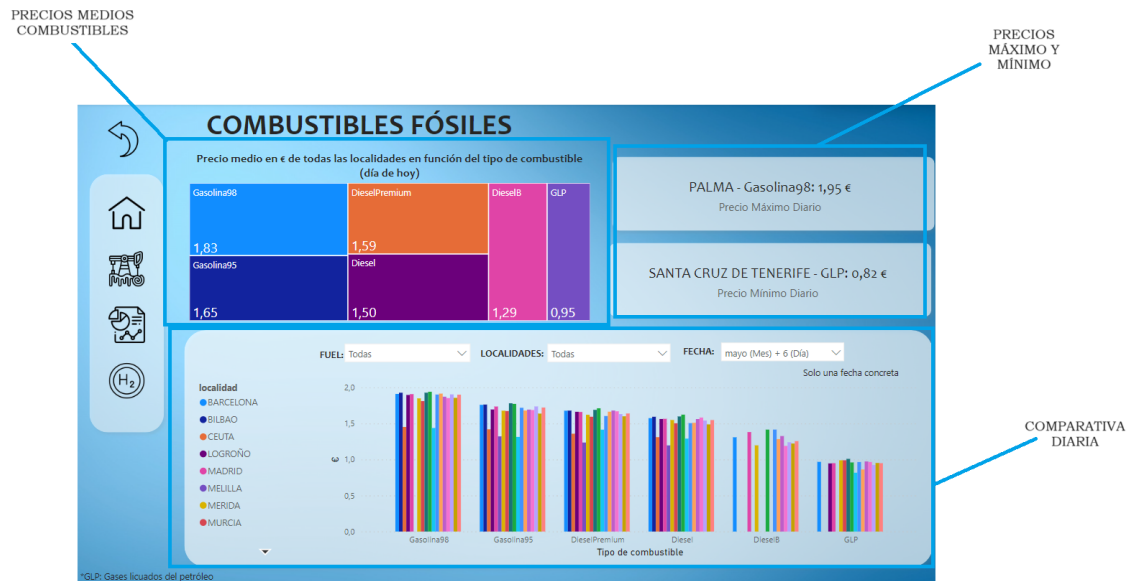


Ilustración 28: Pantalla final análisis diario combustibles fósiles

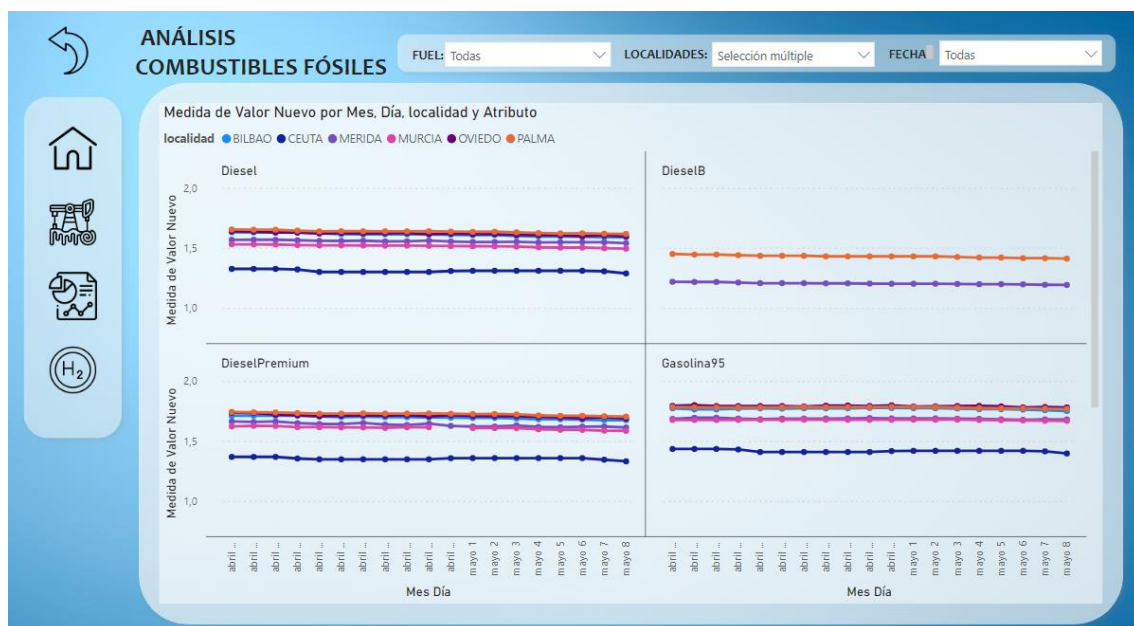



Ilustración 29: Pantalla final evolución precios combustibles fósiles

El primer paso para obtener lo descrito anteriormente es limpiar los datos, reordenarlos y filtrarlos; de forma que obtengamos la transformación mostrada en la Ilustración 28.



PK: PartitionKey	PK: RowKey	Timestamp	Content
1	DefaultPartitionKey	000915c1-576f-4c4c-bd26-64536e370d77	31/03/2024 7:00:23 Record
2	DefaultPartitionKey	001c22a3-8406-4583-8ac7-4fcb88ba9484	11/05/2024 7:00:14 Record
3	DefaultPartitionKey	001ecc28-036a-4064-bc5a-b7d70539f299	31/03/2024 7:00:23 Record
4	DefaultPartitionKey	0025fa95-6a3f-469f-b1a9-78b33ec81761	05/04/2024 7:00:29 Record
5	DefaultPartitionKey	0031fe94-bc31-4961-b9d4-25bd7bbac6ee	22/03/2024 9:01:09 Record
6	DefaultPartitionKey	003c5389-b333-4d11-ada5-8c78865cae5e	06/04/2024 7:00:30 Record
7	DefaultPartitionKey	003f801b-8203-4843-9012-3ee46f6e161	14/04/2024 7:00:25 Record
8	DefaultPartitionKey	004fc3bb-afed-4980-9256-8868ef6e5ec0	22/03/2024 9:01:09 Record
9	DefaultPartitionKey	00506c24-71ad-413c-900d-470f5189528e	28/03/2024 7:00:22 Record
10	DefaultPartitionKey	00589a15-92a9-464d-85cb-ebbcabf5feaf	30/03/2024 7:00:27 Record
11	DefaultPartitionKey	00610d80-8413-4f15-a83f-23fec6118fe9	28/03/2024 7:00:27 Record
12	DefaultPartitionKey	006508ab-ae56-4dc2-8402-fad8f13ed0f0	30/04/2024 7:00:50 Record
13	DefaultPartitionKey	00664d93-a956-4e2a-9dce-a1460523149	01/04/2024 7:00:16 Record
14	DefaultPartitionKey	0066beba-7601e-46f8-a4c7-20a23c3431c7	06/05/2024 7:00:27 Record
15	DefaultPartitionKey	0066c014-718a-46cb-a00c-d573db318549	13/05/2024 7:00:21 Record
16	DefaultPartitionKey	007724f6-3a2a-4163-ae29-5f64ab30a18b	10/04/2024 7:00:20 Record
17	DefaultPartitionKey	00779f55-026c-4bea-8c93-bc3baac728f9	24/03/2024 7:00:21 Record
18	DefaultPartitionKey	007cd843-4035-4d4a-b8ed-9ac85771d749	14/05/2024 7:00:29 Record
19	DefaultPartitionKey	00856277-af40-4591-a7e2-bb829cde469d	05/04/2024 7:00:29 Record
20	DefaultPartitionKey	0086c997-b2a3-40fe-9486-c89c948044e3	27/04/2024 7:00:23 Record
21	DefaultPartitionKey	00920027-fbe4-4564-9983-4c11721ade2f	23/04/2024 7:00:25 Record

PK: Localidad	Timestamp	PK: Atributo	1.2 Valor
BARCELONA	15/05/2024	Diesel	1,556259259
BARCELONA	15/05/2024	Gasolina95	1,737379747
BARCELONA	15/05/2024	Gasolina98	1,891166667
BARCELONA	15/05/2024	DieselPremium	1,659050847
BARCELONA	15/05/2024	DieselB	1,295333333
BARCELONA	15/05/2024	GLP	0,967230769
BARCELONA	14/05/2024	Diesel	1,556469136
BARCELONA	14/05/2024	Gasolina95	1,737949367
BARCELONA	14/05/2024	Gasolina98	1,891833333
BARCELONA	14/05/2024	DieselPremium	1,659118644
BARCELONA	14/05/2024	DieselB	1,296166667
BARCELONA	14/05/2024	GLP	0,967230769
BARCELONA	13/05/2024	Diesel	1,559851852
BARCELONA	13/05/2024	Gasolina95	1,742797468
BARCELONA	13/05/2024	Gasolina98	1,896814815
BARCELONA	13/05/2024	DieselPremium	1,661254237
BARCELONA	13/05/2024	DieselB	1,296166667
BARCELONA	13/05/2024	GLP	0,967230769
BARCELONA	12/05/2024	Diesel	1,563049383
BARCELONA	12/05/2024	Gasolina95	1,744481013
BARCELONA	12/05/2024	Gasolina98	1,897740741

Ilustración 30: Transformación de los datos de la tabla "datosdeBarcelona"

El código para la transformación de los datos está incluido en la Ilustración 52 del Anexo 8.2. Las palabras “let” e “in” como ya se introdujo sirven para estructurar las consultas. Con la primera se inicializa el bloque en el que se van a definir una serie de variables locales, cada una de estas representa una operación intermedia de manipulación de datos; con la segunda se finaliza el bloque y se especifica el resultado de la consulta.

Para todas las tablas cargadas de Azure se han seguido los mismos pasos:

1. Las dos primeras sentencias del bloque quedan registradas en el momento en el que se establece la conexión con Azure y sirven para indicar el origen de los datos.
2. Filtramos por fecha, únicamente queremos que se carguen la información recogida a partir del 20 de abril (ver apartado 4. Limitaciones).
3. Expandimos la columna Content que es donde vienen por columnas los precios de cada tipo de combustible.
4. Nos aseguramos de que los datos de las columnas, dado que son precios; estén en formato número decimal, y adaptamos el formato de la fecha a “dd/mm/yyyy”.
5. Anulamos la dinamización de las columnas, esto lo que va a suponer es que los combustibles que antes estaban cada uno en una columna pasen a ser parte de las filas. En otras palabras, en la nueva estructura de tabla cada fila representa un dato único de precio para un tipo específico de combustible en una localidad y fecha dadas; lo que hace más fácil el posterior trabajo con ellos.
6. Reordenamos las columnas, y agrupamos por fecha, localidad y tipo de combustible. De esta forma el precio que aparece para cada tipo de combustible para una fecha dada en una localidad específica es realmente una media de los precios de todas las estaciones de la región que sirven ese recurso específico, en la fecha concreta.

Una vez se han transformado los datos, es necesario realizar las consultas pertinentes para poder mostrarlos tal y cómo aparecen en la Ilustración 28. Para cada uno de los tres elementos, se han tenido que crear consultas diferentes pues el resultado varía de unos a otros.

- **Precios Medios Combustibles** (ver Ilustración 54 del Anexo 8.2): se unen todas las tablas ya transformadas, y se filtra por la última fecha registrada (esta fecha cambia diariamente cada vez que se realiza la recogida de datos de la API). En el gráfico se añaden los valores de las columnas “Atributo” (tipo de combustible) y “Valor” (precio) de la nueva tabla generada renombrada como “Promedio precios”. Una de las prestaciones de Power BI es que al añadir datos a gráficos puedes calcular automáticamente medidas con ellos, así que sin necesidad de expandir más la consulta en el gráfico se añade el “Promedio de Valor”.
- **Precios Máximo y Mínimo:** la única consulta necesaria de Power Query es la de unión de las tablas de Azure (resulta en una nueva tabla llamada “Tarjeta Max-Min”), porque para la obtención de los precios máximos y mínimos, para no tener que separarlos en dos consultas diferentes se han empleado medias escritas en lenguaje DAX. La medida para la obtención del precio máximo es la de la siguiente Ilustración. Para la obtención del precio mínimo, se realiza de la misma forma. La medida se llama “Precio_Mínimo_Ultimo_Dia”, y lo único que cambia es el cálculo.

```
Precio_Maximo_Ultima_Fecha =
VAR LatestTimestamp = MAX('Tarjeta Max-Min'[Timestamp])
VAR MaxPrecioEnUltimoTimestamp = CALCULATE(
    MAX('Tarjeta Max-Min'[Valor]),
    'Tarjeta Max-Min'[Timestamp] = LatestTimestamp
)
RETURN
    CALCULATE(
        MAXX(
            FILTER(
                'Tarjeta Max-Min',
                'Tarjeta Max-Min'[Timestamp] = LatestTimestamp && 'Tarjeta Max-
Min'[Valor] = MaxPrecioEnUltimoTimestamp
            ),
            'Tarjeta Max-Min'[localidad] & " - " & 'Tarjeta Max-Min'[Atributo] & ": "
        ) & FORMAT(MaxPrecioEnUltimoTimestamp, "Currency")
    ),
    REMOVEFILTERS()
)
```

Ilustración 31: Obtención con DAX: precio máximo última fecha registrada

La medida tiene como nombre “Precio_Maximo_Ultima_Fecha”, en ella primero se crea la variable “LatestTimestamp” para obtener la fecha más reciente disponible en los datos de la tabla “Tarjeta Max-Min”, y posteriormente en la variable “MaxPrecioEnUltimoTimestamp” a través de la función “CALCULATE” se obtiene el precio máximo garantizando que sea el más reciente.

No basta solo con calcularlo, sino que es necesario establecer un formato final de la información. Con “FILTER” se obtienen únicamente los registros que coinciden con las variables creadas anteriormente, y con “MAXX” se busca y da formato la información sobre la localidad y el tipo de combustible (almacenado en la columna “Atributo” de la tabla) asociados al precio máximo. La función “FORMAT” permite formatear el precio máximo como una cadena de texto en formato de moneda.

- **Comparativa diaria:** al igual que para los precios máximos y mínimo, se crea una nueva consulta, que genera una nueva tabla en este caso “Promedio precios-Grafico”. Una de las funcionalidades que se le ofrecen al usuario es que pueda seleccionar sobre qué tipo de combustible, que localidades y en qué fecha concreta quiere realizar el análisis. Para ello se ha hecho uso de la segmentación de datos. En concreto se han añadido tres filtros para este gráfico, por tipo de combustible, por localidad y por fecha. Las segmentaciones se pueden modificar en función de lo que se pretenda conseguir. Es por esto por lo que, en los dos primeros filtros, el usuario tiene la opción de seleccionar las opciones que desee; sin embargo, el filtro de fecha queda restringido a selección única. De esta forma se asegura que en este gráfico el análisis sea por día y se enfoque más en una comparativa entre regiones que en la evolución de los precios; funcionalidad que también se ofrece, pero en otra pantalla de la herramienta (Ilustración 29).

Para la creación del gráfico, se crea una medida (Ilustración 32) que sirva para mostrar los precios en función de la segmentación. Como se puede observar para el cálculo se ha utilizado la función “MIN” pero podría haberse usado también la función “MAX”. Esto se debe a que solo existe un único valor para cada localidad, tipo de combustible y fecha específica. En este caso el uso de estas funciones es trivial, pero en otros casos sí que podría ser significativo el uso de uno o de otro.

Medida de Valor Grafico = `MIN('Promedio precios-Grafico'[Valor])`

Ilustración 32: Medida para el gráfico de análisis diario

En cuanto a la obtención del gráfico que muestra la evolución de los precios (Ilustración 29), se utiliza la misma mecánica que en el gráfico de análisis diario con la salvedad de que la segmentación de fecha no está restringida y el usuario puede elegir según sus necesidades el periodo que desee analizar.

La segunda parte de la plataforma se corresponde con la parte estática, los datos relacionados con el hidrógeno. En la Ilustración 33 se muestra el resultado final tras la limpieza y transformación de los datos.

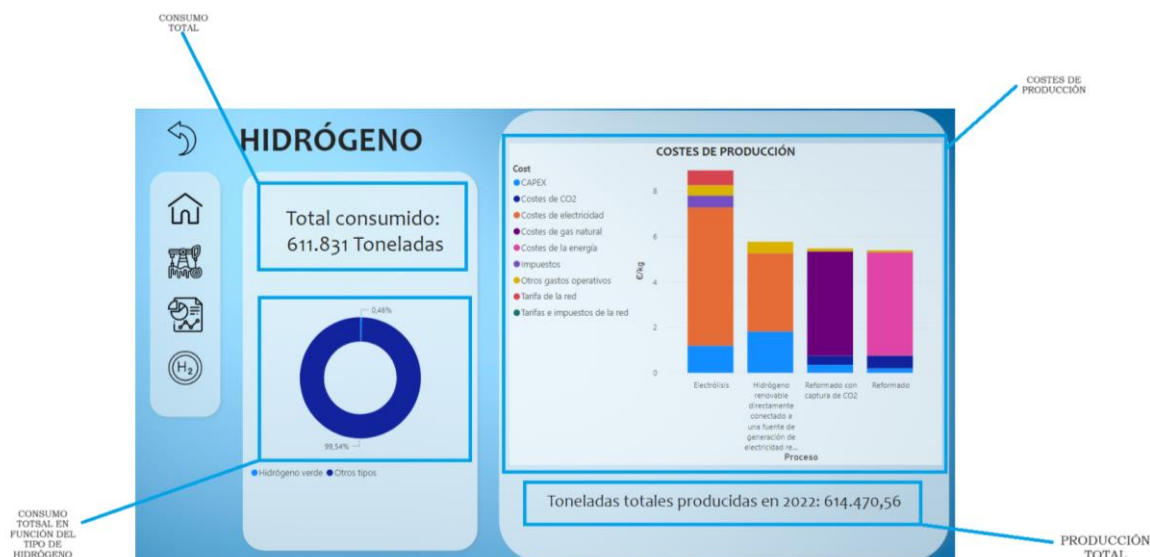


Ilustración 33: Pantalla sobre el hidrógeno

En la Ilustración 26 se puede observar la vista previa de cómo se van a importar los datos en Power BI. En concreto, la información de la imagen corresponde con la producción de hidrógeno en Europa en 2022 [18]. Para las otras dos fuentes de datos relacionadas con los costes y el consumo de hidrógeno, la estructura es similar a la de la Ilustración 26, por lo que el primer paso es limpiar y transformar los datos para obtener valor de ellos. Además, la información descargada está en inglés por lo que es necesario traducir los nombres que se vayan a mostrar en el informe final ya que el público al que va dirigido tiene como primera lengua el español. En las Ilustraciones 53,55 y 56 del Anexo 8.2 se puede examinar el código en Power Query para la limpieza y transformación de los datos de hidrógeno, para obtener las tablas mostradas en la siguiente ilustración.

Producción

	ABC 123 Country	ABC 123 Process type	ABC 123 End-use	ABC 123 Production capacity T/year	ABC 123 Output T/year
1	Spain	By-product	Unknown	1400	966
2	Spain	By-product	Other chemicals	10052	6935,88
3	Spain	By-product	Other chemicals	11529,675	9225,189362
4	Spain	By-product	Refining	13733,124	10988,22555
5	Spain	By-product	Unknown	2367,9	1846,962
6	Spain	By-product	Other chemicals	229,6945	179,16171
7	Spain	By-product	Refining	8900,1	7878,777049
8	Spain	Reforming	Ammonia	108129,4725	66298,56006
9	Spain	Reforming	Refining	468686	374076,2198
10	Spain	Reforming	Unknown	43334,49112	33800,90307
11	Spain	Reforming	Other chemicals	8863,873184	6913,821084
12	Spain	Reforming	Refining	115670,6345	92551,04822
13	Spain	Water electrolysis	Ammonia	3305,660377	2247,849057
14	Spain	Water electrolysis	Mobility	826,4150943	561,9622642

Costes Producción

	ABC Country	ABC Technology	ABC Cost	1.2 Value (€/kg)
1	Spain	Electrólisis	CAPEX	1,189625926
2	Spain	Electrólisis	Tarifa de la red	0,650126997
3	Spain	Electrólisis	Otros gastos operativos	0,457325845
4	Spain	Electrólisis	Impuestos	0,520101597
5	Spain	Electrólisis	Costes de la red eléctrica	6,091362675
6	Spain	Reformado con captura de CO2	CAPEX	0,357430417
7	Spain	Reformado con captura de CO2	Costes de CO2	0,404611323
8	Spain	Reformado con captura de CO2	Costes de Gas Natural	4,581792205
9	Spain	Reformado con captura de CO2	Otros gastos operativos	0,131201875
10	Spain	Hidrógeno renovable	CAPEX	1,818209216
11	Spain	Hidrógeno renovable	Costes de electricidad	3,448532465
12	Spain	Hidrógeno renovable	Tarifas e impuestos de la red	0
13	Spain	Hidrógeno renovable	Otros gastos operativos	0,500546857
14	Spain	Reformado	CAPEX	0,199724684
15	Spain	Reformado	Costes de CO2	0,554676485
16	Spain	Reformado	Costes de energía	4,547767561
17	Spain	Reformado	Otros gastos operativos	0,091633056

Consumo anual

	ABC Country	1.2 Clean consumption T/Y	1.2 Other consumption T/Y	1.2 Total consumption
1	Spain	0	478901,3353	478901,3353
2	Spain	0	17718,93854	17718,93854
3	Spain	0	14519,06665	14519,06665
4	Spain	2247,849057	64050,71101	66298,56006
5	Spain	38,76	0	38,76
6	Spain	523,2022642	33831,57767	34354,77994

Ilustración 34: Tablas finales en Power BI sobre el hidrógeno

Para obtener la información mostrada en la Ilustración 33 es necesario crear medidas en lenguaje DAX.

- **Costes de producción:** el gráfico muestra para cada método de producción de hidrógeno utilizado en España un desglose de los costes. Para ello, se crea la medida “Produccion costes” que suma todos los costes. Al añadirla como valores del eje Y del gráfico, se dividen en función del método de producción.

```
Produccion Costes = SUM('Costes Producción'[Value (€/kg)])
```

Ilustración 35: Medida en DAX Produccion costes

- **Producción total:** es la suma de todas las toneladas de hidrógeno producidas en España en 2022. En la medida creada, se le da el formato que aparece en la Ilustración 33.

```
Produccion Total = "Toneladas totales producidas en 2022: " &  
FORMAT(SUM('Produccion'[Output T/year]),"0,0.00")
```

Ilustración 36: Medida en DAX Produccion Total

- **Consumo total:** utilizando la columna “Total consumption” de la tabla “Consumo anual”, se hace una medida similar a la anterior.
- **Consumo total en función del tipo de hidrógeno:** en este caso a partir de la tabla “Consumo anual” es necesario hacer dos medidas que sumen el total de “Clean consumption T/y” y el total de “Other consumption T/y”. Estas dos medidas son añadidas a un gráfico circular, en el que se elige que los datos se muestren como porcentaje.

3.5.5 Diseño de la herramienta

Para el diseño de la herramienta se ha optado por una estructura que divide la información en varias páginas en vez de agrupar todos los datos en una sola pantalla. Esta decisión pretende mejorar la experiencia del usuario al facilitar la navegación y hacer que la interacción con la plataforma sea intuitiva.

Además de las pantallas que se han mostrado anteriormente donde se refleja la información, la herramienta tiene una página principal (Ilustración 37). Esta pantalla ofrece una visión de las opciones disponibles en la plataforma. Cada uno de los tres iconos, sirve como un enlace de navegación que permite al usuario acceder a las páginas específicas. Asimismo, la fecha ubicada en la esquina superior izquierda (presente en todas las páginas); funciona como un botón de retorno permitiendo a los usuarios volver a la vista anterior. Esto es una práctica común en el diseño de interfaces para mejorar la navegación.

Otra característica que destacar del diseño y que se puede también observar en las Ilustraciones 28, 29 y 33 es la barra de navegación. Esta barra incluye los mismos iconos de la página principal y un icono extra en forma de casa que hace referencia a esta. Así se proporciona una forma consistente de navegación entre secciones sin necesidad de regresar a la pantalla de inicio.



Ilustración 37: Pantalla de inicio de la herramienta

3.5.6 Optimización y Automatización de la Publicación del Informe

Una vez se ha completado el desarrollo de la herramienta, el siguiente paso es su publicación en el área de trabajo asociada a la cuenta de Microsoft donde se estaba desarrollando en local. Este proceso permite que el informe sea accesible a través de la plataforma web de Power BI, donde se puede visualizar, realizar ajustes y modificaciones adicionales y donde se va a gestionar el control de las actualizaciones de datos diarias.

La publicación genera dos componentes (Ilustración 38):

- **Modelo semántico:** actúa como núcleo gestionando el flujo de datos y las credenciales necesarias para las actualizaciones.
- **Informe:** permite la manipulación continua del contenido visual y analítico. Desde aquí se puede adaptar la herramienta a nuevas necesidades actualizando el producto final de forma automática.




	Nombre	Tipo
	Dashboard	Informe
	Dashboard	Modelo semántico

Ilustración 38: Componentes tras la publicación del Informe

La generación de un enlace web para hacer pública la herramienta es a través del “Informe”.

Una vez dentro de él, siguiendo los pasos que aparecen en la primera parte de la Ilustración 39, aparecerá la pantalla inferior en la que se genera un enlace asociado al Informe, y además permite la personalización de visualización de la herramienta.

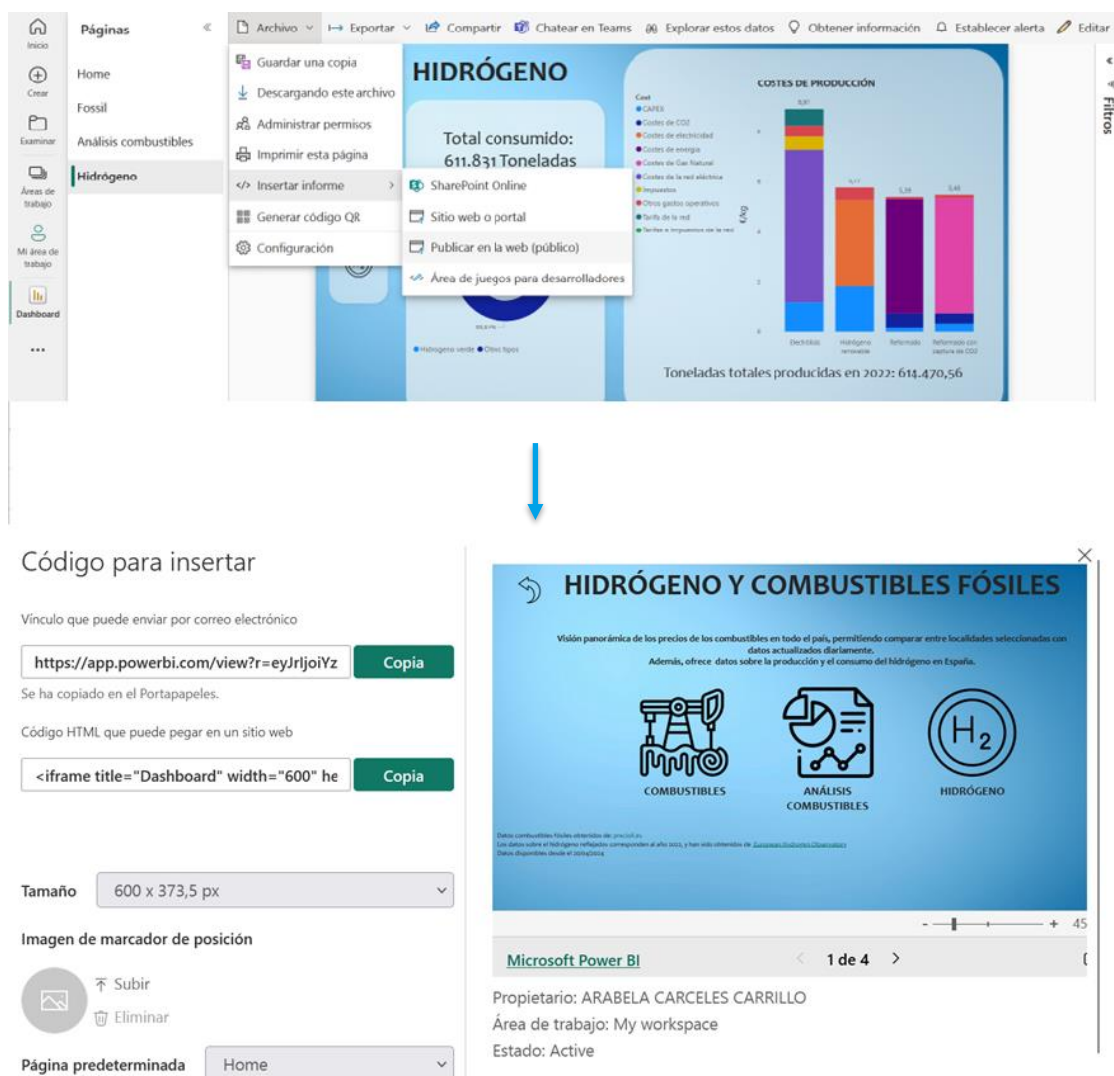


Ilustración 39: Creación enlace web

El enlace que se obtiene es relativamente largo, por lo que para este proyecto se ha hecho uso una vez mas de servicios desarrollados por la Universidad Politécnica de Madrid para acortarlo.

<https://short.upm.es/ndgmd>

Ilustración 40: Enlace acortado herramienta

El último paso es la gestión de las actualizaciones. Hasta ahora el único desencadenante de actualizaciones se habría desarrollado en Microsoft Azure para guardar diariamente en tablas los datos de los combustibles fósiles. Sin embargo, si no existe un desencadenante desde Power BI que actualice los datos de la plataforma una vez se han recogido, no se reflejarían.

La programación de las actualizaciones se realiza desde el “Modelo semántico”. Un paso muy importante es verificar que las conexiones y las credenciales al origen de los datos sean adecuadas, es decir; deberían aparecer como se muestra en la Ilustración 41. Es probable que a la hora de verificarlas, la interfaz pida que vuelvas a autenticarte si es necesaria alguna clave, como es el caso de la cuenta de almacenamiento de Azure.



Ilustración 41: Conexiones y credenciales de los datos

Una vez establecidas las conexiones, se ha programado una actualización diaria que se lleva a cabo treinta minutos después de la actualización de Azure, con el fin de asegurar que cualquier retraso no impida la correcta captura y registro de los datos.

4 Limitaciones y desafíos

Hasta ahora se ha revisado toda la estructura de la herramienta y su desarrollo. Sin embargo, es necesario mencionar ciertas limitaciones que se han de tener en cuenta si se pretende usar la herramienta para la toma de decisiones en un entorno empresarial.

En primer lugar, la API pública que se ha usado ofrece una solución para la recolección de datos muy sencilla y fácil de usar. Sin embargo, el hecho de que haya que meter el nombre de la localidad a mano, y que no exista un método para obtener todas las localidades disponibles ha supuesto que para el desarrollo de la primera versión de la herramienta solo se recojan datos de las capitales de las diferentes Comunidades Autónomas. Además, no existe posibilidad de obtener datos de Pamplona, por lo que esta región está excluida del análisis. Todo esto supone, que el usuario tenga que limitarse al ámbito de dichas localidades y no pueda explorar más datos. Como todo proyecto, el informe generado es simplemente una primera versión, y se podría ir actualizando en versiones posteriores para ir recogiendo más datos e ir expandiéndolos.

Otro factor para tener en cuenta es que la API utilizada es un servicio en pruebas. Esto ha supuesto inconvenientes durante todo el desarrollo del proyecto, pues por actividades de mantenimiento y actualización del servicio la estructura de los datos ha ido cambiando, e incluso se han eliminado algunos. Estos cambios en la estructura de la información no han supuesto problemas a la hora de la extracción, pues las tablas se adaptaban dinámicamente para alojar la nueva estructura; pero sí que en Power BI se han tenido que cambiar las consultas cada vez que lo hacían los datos. Además, era necesario controlar que días se habían producido los cambios en la estructura, para filtrar en la aplicación a partir de qué día se tenían que transformar los datos para añadirlos al informe. Es por esto por lo que sólo se muestran datos a partir del 20 de abril de 2024, cuando en realidad las tareas de recolección llevan en marcha desde el 23 de marzo de 2024.

Por otro lado, los servicios proporcionados por Azure suelen tener un coste asociado. Para el desarrollo de la plataforma, se eligió utilizar estos servicios aprovechando la licencia de Microsoft proporcionada por la Universidad Politécnica de Madrid (UPM), que incluye un saldo para fines educativos. Es importante tener en cuenta que, si la herramienta fuera a ser utilizada por una empresa sería necesario suscribirse y pagar por los recursos que recopilan y almacenan dinámicamente los datos. Dado que el saldo disponible es limitado, se ha programado la finalización de la actualización de los datos para el 1 de julio. Al igual que con Azure, algunas de las funcionalidades de Power BI utilizadas en la publicación de la plataforma no son gratuitas para los usuarios, pero al utilizar la cuenta institucional de la UPM sí que se han podido utilizar.

Un desafío que apareció en las últimas etapas del desarrollo está relacionado con la importación de los datos estáticos. Idealmente se hubiesen extraído conectando el Power BI directamente con el archivo Excel que se utilizó para el análisis del TFG del Grado de Administración de Empresas. Sin embargo, debido a la programación de actualizaciones diarias de los datos en Power BI era necesario mantener operativo un Gateway personal de datos en todo momento para poder conectar el archivo Excel local con Power BI y permitir la publicación de los datos.

Realmente las actualizaciones diarias solo afectarían a los datos extraídos de forma dinámica, pero como los datos estáticos también forman parte del modelo tienen que pasar por el proceso, aunque no tengan modificaciones.

El uso del Gateway requiere mantener una sesión iniciada constantemente para que funcione y sincronice los datos. Esto contradecía el objetivo de automatizar las actualizaciones ya que requería el inicio de sesión en el momento en el que se fuera a producir la actualización para mantener el sistema operativo. Es por esto por lo que se optó por la descarga directa de datos desde European Hydrogen Observatory. De esta forma se aseguró el acceso a los datos de manera regular y automática.

5 Conclusiones y resultados

Una vez expuesto el desarrollo como las limitaciones de la plataforma encontradas durante la implementación, es adecuado valorar el resultado final y su impacto. La idea de este TFG era el desarrollo de una plataforma de monitorización y control en tiempo real de los sistemas de almacenamiento y distribución de hidrógeno en empresas. Debido a la inmadurez del mercado y la falta de información se enfocó el desarrollo hacia un enfoque que generara valor significativo incluso con las limitaciones encontradas.

La elección de diseño de la herramienta permite que sea dinámica y escalable, permitiendo incorporar nuevas funcionalidades y datos, asegurando que sea relevante y útil en el largo plazo. La usabilidad de la plataforma se muestra a través de una interfaz muy intuitiva e interactiva, y a través de las múltiples herramientas de análisis que proporciona.

El impacto de la plataforma en las operaciones diarias empresariales puede ser profundo. Al proporcionar datos precisos y actualizados en tiempo real, el propósito es que aquellas compañías líderes en sostenibilidad que busquen transaccionar hacia tecnologías de hidrógeno puedan mejorar su capacidad de respuesta ante las fluctuaciones del mercado y reducir costes mediante una mejor planificación y gestión de los recursos.

La elección de las tecnologías de Microsoft Azure y Power BI ofrecen robustez, escalabilidad e integración de los datos. Azure permite un manejo de grandes volúmenes de información, crucial para el funcionamiento de la herramienta que requiere adaptabilidad y actualización constante. Por otro lado, Power BI ofrece capacidades avanzadas de visualización de datos y análisis, impulsando la toma de decisiones de los negocios. Además, la integración de ambas plataformas facilita el mantenimiento del flujo de datos de forma coherente.

En lugar de desarrollar una aplicación web propia para la visualización de los datos, se optó por hacer uso de las prestaciones de Power BI. Las capacidades avanzadas de creación de informes dinámicos interactivos permiten al usuario final acceder a visualizaciones comprensibles, crucial para el contexto empresarial en el que se pretende introducir la herramienta. Desde la perspectiva del desarrollador, el amplio soporte técnico y la gran comunidad activa con la que cuenta permite que cualquier desafío pueda ser rápidamente abordado. Además, Power BI ofrece muchas opciones de personalización y configuración permitiendo obtener un producto final en línea con las necesidades del usuario.

En conclusión, la plataforma desarrollada podría posicionarse como una herramienta indispensable en el panorama empresarial. Su capacidad para integrar y analizar los datos de manera eficiente y proporcionar visualizaciones relevantes de ellos, permite mejorar la gestión energética dentro de las empresas. Además, con este trabajo se refleja el potencial de las nuevas tecnologías para fomentar un futuro verde y sostenible.

6 Análisis de Impacto

A través de la evaluación de impacto se busca comprender los beneficios y posibles desafíos asociados al desarrollo de la plataforma. Personalmente enfrentarme a dos tecnologías bastante novedosas para mí ha supuesto una mejora en mis habilidades y conocimiento. Durante la implementación he adquirido una mejor soltura programando en Python, lenguaje que no había utilizado mucho hasta ahora; y he entrado en contacto con Microsoft de Azure, servicio el cual no conocía.

Para las empresas, esta herramienta pretende ser el inicio para la mejora de la eficiencia operativa y de la gestión de los costes asociados con los recursos de hidrógeno. Facilita decisiones estratégicas basadas en datos actualizados diariamente, crucial para mantener la competitividad en un mercado que está en constante evolución. Sin embargo, siempre existen riesgos asociados con la adopción de nuevas herramientas como por ejemplo el aprendizaje por parte de los empleados de su uso.

Socialmente se pretende fomentar una mayor conciencia sobre la importancia de la transición energética y de las fuentes de energía renovables. Se muestra como los precios de los combustibles cada vez van en aumento, y la necesidad de adoptar otras fuentes de energía. En relación con el encarecimiento de los recursos, desde la perspectiva de las empresas; esto es esencial para el control de costes. La plataforma pretende contribuir en el largo plazo a que las empresas estudien la viabilidad de la transición de las energías tradicionales a las energías obtenidas a partir de hidrógeno.

En cuanto a los Objetivos de Desarrollo Sostenible (ODS) de la Agenda 2030, el proyecto contribuye a:

1. ODS 7- Energía Asequible y No Contaminante: la plataforma busca que las empresas transicionen hacia tecnologías de hidrógeno verde, una fuente de energía limpia y renovable.
2. ODS 9 – Industria, Innovación e Infraestructura: la implementación de soluciones tecnológicas relacionadas con el hidrógeno es un ejemplo de cómo la innovación puede contribuir a la industrialización inclusiva y sostenible con el medio ambiente.
3. ODS 13 – Acción por el Clima: el informe facilita la adopción y el manejo eficiente del hidrógeno como fuente de energía limpia, contribuyendo de forma indirecta a la reducción de emisiones de gases de efecto invernadero
4. ODS 12 – Producción y Consumo Responsables: la eficiencia en la gestión del hidrógeno puede llegar a contribuir en la adopción de prácticas de consumo más sostenibles al optimizar el uso de recursos y reducir el desperdicio.

7 Bibliografía

- [1] A. C. Carrillo, *Estudio de Mercado Manufacturero, Logístico y Financiero para Aplicaciones de Hidrógeno: Oportunidades y Desafíos en el Contexto Empresarial*, Madrid, 2024.
- [2] Microsoft, «Azure,» 2024. [En línea]. Available: <https://azure.microsoft.com/es-es/resources/cloud-computing-dictionary/what-is-azure#:~:text=Azure%20proporciona%20soluciones%20para%20todos,con%20las%20soluciones%20de%20Azure..>
- [3] Microsoft, «¿Qué es la nube?,» 2024. [En línea]. Available: <https://azure.microsoft.com/es-es/resources/cloud-computing-dictionary/what-is-the-cloud/>.
- [4] Varios, «Best practices in cloud applications,» 16 Diciembre 2022. [En línea]. Available: <https://learn.microsoft.com/en-us/azure/architecture/best-practices/index-best-practices>.
- [5] Varios, «¿Qué es Azure Table Storage?,» 20 Abril 2023. [En línea]. Available: <https://learn.microsoft.com/es-es/azure/storage/tables/table-storage-overview>.
- [6] Varios, «Azure reliability documentation,» 24 Septiembre 2023. [En línea]. Available: <https://learn.microsoft.com/en-us/azure/reliability/overview>.
- [7] Microsoft, «Servicios de Azure AI,» 2024. [En línea]. Available: <https://azure.microsoft.com/es-es/products/ai-services>.
- [8] Microsoft, «Azure IoT,» 2024. [En línea]. Available: <https://azure.microsoft.com/es-es/solutions/iot>.
- [9] Microsoft, «Documentación SDK Python,» 2024. [En línea]. Available: <https://learn.microsoft.com/es-es/python/api/overview/azure/?view=azure-python>.
- [10] Microsoft Learn, «Azure for Python Developers,» [En línea]. Available: <https://learn.microsoft.com/en-us/azure/developer/python/?view=azure-python>.
- [11] Varios, «Bibliotecas de Azure Data Factory para Python,» 12 Junio 2018. [En línea]. Available: <https://learn.microsoft.com/es-es/python/api/overview/azure/datafactory?view=azure-python>.
- [12] Varios, «¿Qué es Power BI?,» 22 Marzo 2024. [En línea]. Available: <https://learn.microsoft.com/es-es/power-bi/fundamentals/power-bi-overview>.
- [13] Varios, «¿Qué es Power Query?,» Microsoft, 24 Enero 2024. [En línea]. Available: <https://learn.microsoft.com/es-es/power-query/power-query-what-is-power-query>.
- [14] Varios, «Información general sobre DAX,» Microsoft, 20 Octubre 2023. [En línea]. Available: <https://learn.microsoft.com/es-es/dax/dax-overview>.

- [15] «¿Qué es una interfaz de programación de aplicaciones (API)?» [En línea]. Available: <https://aws.amazon.com/es/what-is/api/>.
- [16] European Hydrogen Observatory, «Hydrogen Demand,» 2022.
- [17] European Hydrogen Observatory, «Cost of hydrogen production,» 2022.
- [18] European Hydrogen Observatory, «Hydrogen Production,» 2022.
- [19] Varios, «¿Qué es Azure Data Factory?,» 20 Julio 2023. [En línea]. Available: <https://learn.microsoft.com/es-es/azure/data-factory/introduction>.
- [20] Varios, «Administrar grupos de recursos de Azure Resource Manager mediante Python,» 27 Enero 2024. [En línea]. Available: <https://learn.microsoft.com/es-es/azure/azure-resource-manager/management/manage-resource-groups-python?tabs=windows>.
- [21] Varios, «Ejemplo: Uso de las bibliotecas de Azure para crear un grupo de recursos,» 6 Marzo 2024. [En línea]. Available: <https://learn.microsoft.com/es-es/azure/developer/python/sdk/examples/azure-sdk-example-resource-group?view=azure-python&tabs=cmd>.
- [22] Varios, «Storage account overview,» 12 Junio 2023. [En línea]. Available: <https://learn.microsoft.com/en-us/azure/storage/common/storage-account-overview>.
- [23] Varios, «¿Qué es Azure Pipelines?,» 25 Marzo 2024. [En línea]. Available: <https://learn.microsoft.com/es-es/azure/devops/pipelines/get-started/what-is-azure-pipelines?view=azure-devops>.
- [24] Varios, «Bibliotecas cliente de Azure Storage para Python,» 2 Abril 2024. [En línea]. Available: <https://learn.microsoft.com/es-es/python/api/overview/azure/storage?view=azure-python>.
- [25] Varios, «Azure Tables client library for Python - version 12.5.0,» 16 Enero 2024. [En línea]. Available: <https://learn.microsoft.com/en-us/python/api/overview/azure/data-tables-readme?view=azure-python>.
- [26] Varios, «Copy activity in Azure Data Factory and Azure Synapse Analytics,» 10 Agosto 2023. [En línea]. Available: <https://learn.microsoft.com/en-us/azure/data-factory/copy-activity-overview>.
- [27] Varios, «Schema and data type mapping in copy activity,» 20 Octubre 2023. [En línea]. Available: <https://learn.microsoft.com/en-us/azure/data-factory/copy-activity-schema-and-type-mapping#tabular-source-to-tabular-sink>.
- [28] Varios, «Copia y transformación de datos desde y hacia un punto de conexión de REST mediante Azure Data Factory,» 16 Febrero 2024. [En línea]. Available: <https://learn.microsoft.com/es-es/azure/data-factory/connector-rest?tabs=data-factory>.
- [29] Varios, «Inicio rápido: Creación de una factoría de datos y una canalización con SDK de .NET,» 15 Diciembre 2023. [En línea]. Available:

<https://learn.microsoft.com/es-es/azure/data-factory/quickstart-create-data-factory-dot-net>.

- [30] Microsoft Learn, «RestServiceLinkedService Class,» [En línea]. Available: <https://learn.microsoft.com/en-us/python/api/azure-synapse-artifacts/azure.synapse.artifacts.models.restservicelinkedservice?view=azure-python-preview>.
- [31] Varios, «Servicios vinculados en Azure Data Factory y Azure Synapse Analytics,» 20 Octubre 2023. [En línea]. Available: <https://learn.microsoft.com/es-es/azure/data-factory/concepts-linked-services?tabs=data-factory#data-store-linked-services>.

8 Anexos

8.1 Python

```
GrupoRecursos.py > ...
1 import os #para poder obtener la variable de entorno
2 from azure.identity import DefaultAzureCredential
3 from azure.mgmt.resource import ResourceManagementClient
4
5 # Autenticación para Azure, cogemos la id de la suscripcion de una variable de entorno que ha sido
6 #previamente creada
7 subscription_id = os.getenv('ID_AZURE_SUSCRPTION')
8 #Comprobamos que exista una variable de entorno con el id de la suscripcion
9 if not subscription_id:
10     raise ValueError("Por favor, establece la variable de entorno ID_AZURE_SUSCRPTION antes de ejecutar este script.")
11
12
13 # Punto de entrada para realizar operaciones
14 credential = DefaultAzureCredential()
15 resource_client = ResourceManagementClient(credential, subscription_id)
16
17 #Configuración del grupo de recursos
18 resource_group_name = 'TFGInformatica'
19 location = 'West Europe'
20
21 # Creación el grupo de recursos
22 resource_group = resource_client.resource_groups.create_or_update(
23     resource_group_name,
24     {
25         'location': location
26     }
27 )
28
29 #Print para asegurarnos que se ha creado el grupo de recursos
30 print(f"Grupo de recursos '{resource_group_name}' creado en la ubicación '{location}'".)
31
```

Ilustración 42: Creación grupo de recursos “TFGInformatica”

```
StorageAccount.py > ...
1 import os #para poder obtener la variable de entorno
2 from azure.identity import DefaultAzureCredential
3 from azure.mgmt.storage import StorageManagementClient
4 from azure.mgmt.storage.models import StorageAccountCreateParameters
5
6 # Autenticación para Azure, cogemos la id de la suscripcion de una variable de entorno que ha sido
7 #previamente creada
8 subscription_id = os.getenv('ID_AZURE_SUSCRPTION')
9 #Comprobamos que exista una variable de entorno con el id de la suscripcion
10 if not subscription_id:
11     raise ValueError("Por favor, establece la variable de entorno ID_AZURE_SUSCRPTION antes de ejecutar este script.")
12
13 # Punto de entrada para realizar operaciones
14 credential = DefaultAzureCredential()
15 storage_client = StorageManagementClient(credential, subscription_id)
16
17 # Configuración de la cuenta de almacenamiento
18 resource_group = 'TFGInformatica'
19 storage_account_name = 'datoscombustibles'
20 location = 'West Europe'
21
22
23 # Parámetros de la cuenta de almacenamiento
24 parameters = StorageAccountCreateParameters(
25     sku={'name': 'Standard_LRS'},
26     kind='StorageV2',
27     location=location
28 )
29
30 # Creacion de la cuenta de almacenamiento
31 storage_account = storage_client.storage_accounts.begin_create(
32     resource_group_name=resource_group,
33     account_name=storage_account_name,
34     parameters=parameters
35 ).result()
36
37 #Print para asegurarnos que se ha creado la cuenta de almacenamiento
38 print(f"Storage account {storage_account_name} creado con éxito.")
39
```

Ilustración 43: Creación cuenta de almacenamiento "datoscombustibles"

```

TableStorage.py > ...
1 import os #para poder obtener la variable de entorno
2 from azure.data.tables import TableServiceClient
3
4 #Autenticación de la cuenta de almacenamiento que queremos usar
5 account_name = "datoscombustibles"
6 account_key = os.getenv('AZURE_STORAGE_ACCOUNT_KEY') #ha sido creada anteriormente
7 #Comprobamos que exista una variable de entorno con la clave de la cuenta de almacenamiento
8 if not account_key:
9     raise ValueError("Por favor, establece la variable de entorno AZURE_STORAGE_ACCOUNT_KEY antes de ejecutar este script.")
10
11 # Crear una instancia del servicio TableServiceClient usando la cadena de conexión
12 connection_string = f"DefaultEndpointsProtocol=https;AccountName={account_name};AccountKey={account_key};EndpointSuffix=core.windows.net"
13 table_service = TableServiceClient.from_connection_string(conn_str=connection_string)
14
15 localidades = ['Murcia','Madrid','Sevilla','Zaragoza','Oviedo','Palma','SantaCruzdeTenerife', 'Toledo', 'Santander',
16              'Valladolid', 'Barcelona', 'Merida', 'SantiagodeCompostela', 'Logronio', 'Valencia', 'Bilbao',
17              'Ceuta', 'Melilla']
18 for i in localidades:
19
20     #Configuración de la tabla
21     table_name = "datosde"+i
22     # Creación de la tabla
23     table_client = table_service.create_table_if_not_exists(table_name=table_name)
24     #Print para asegurarnos que se ha creado la tabla
25     print(f"Tabla '{table_name}' creada.")
26
27
28
29
30
31

```

Ilustración 44: Creación tablas de almacenamiento

```

DataFactory.py > ...
1 import os #para poder obtener la variable de entorno
2 from azure.identity import DefaultAzureCredential
3 from azure.mgmt.resource import ResourceManagementClient
4 from azure.mgmt.datafactory import DataFactoryManagementClient
5 from azure.mgmt.datafactory.models import *
6
7 # Autenticación para Azure, cogemos la id de la suscripción de una variable de entorno que ha sido
8 #previamente creada
9 subscription_id = os.getenv('ID_AZURE_SUSCRPTION')
10 #Comprobamos que exista una variable de entorno con el id de la suscripción
11 if not subscription_id:
12     raise ValueError("Por favor, establece la variable de entorno ID_AZURE_SUSCRPTION antes de ejecutar este script.")
13
14 # Punto de entrada para realizar operaciones tanto del grupo de recursos como de la fabrica
15 credential = DefaultAzureCredential()
16 resource_client = ResourceManagementClient(credential, subscription_id)
17 adf_client = DataFactoryManagementClient(credential, subscription_id)
18
19 # Parámetros grupo de recursos
20 resource_group = 'TFGInformatica'
21 location = 'West Europe'
22
23 #Nos aseguramos que el grupo de recursos al que va a estar asociado la fabrica este configurado
24 #correctamente y de acuerdo a la configuración que se va a hacer de la fabrica
25 rg_params = {'location': location}
26 resource_client.resource_groups.create_or_update(resource_group, rg_params)
27
28
29 # Creación de la fabrica de datos
30 data_factory_name = 'fabricadadatostfg'
31 df_resource = Factory(location=location)
32 data_factory = adf_client.factories.create_or_update(resource_group, data_factory_name, df_resource)
33
34 #Print para asegurarnos que se ha creado la fabrica
35 print(f"Data Factory '{data_factory_name}' ha sido creada.")
36

```

Ilustración 45: Creación factoría de datos "fabricadatatostfg"

```

ServicioRestAPIPrecioil.py > ...
1 import os #para poder obtener la variable de entorno
2 from azure.identity import DefaultAzureCredential
3 from azure.mgmt.datafactory import DataFactoryManagementClient
4 from azure.mgmt.datafactory.models import LinkedServiceResource, RestServiceLinkedService
5
6 # Autenticación para Azure, cogemos la id de la suscripción de una variable de entorno que ha sido
7 #previamente creada
8 subscription_id = os.getenv('ID_AZURE_SUSCRPTION')
9 #Comprobamos que exista una variable de entorno con el id de la suscripción
10 if not subscription_id:
11     raise ValueError("Por favor, establece la variable de entorno ID_AZURE_SUSCRPTION antes de ejecutar este script.")
12
13 #Punto de entrada para realizar operaciones
14 credential = DefaultAzureCredential()
15 adf_client = DataFactoryManagementClient(credential, subscription_id)
16
17 #Configuración del servicio rest
18 resource_group = 'TFGInformatica'
19 data_factory_name = 'fabricadadatostfg'
20 rest_service_url = 'https://api.precioil.es/estaciones/conPrecios/localidad/'
21
22
23 #Creación del servicio rest
24 rest_service_linked_service = RestServiceLinkedService(
25     url=rest_service_url,
26     authentication_type='Anonymous' #No hace falta autenticarse
27 )
28
29 ls_name = 'ServicioRestAPIPrecioil'
30 adf_client.linked_services.create_or_update(
31     resource_group_name=resource_group,
32     factory_name=data_factory_name,
33     linked_service_name=ls_name,
34     linked_service=LinkedServiceResource(properties=rest_service_linked_service)
35 )
36
37
38 #Print para asegurarnos que se ha creado la fabrica
39 print(f"Linked Service {ls_name} para REST creado.")
40

```

Ilustración 46: Creación servicio Rest "ServicioRestAPIPrecioil"

```

ServicioTablaAlmacenamiento.py > ...
1 import os #para poder obtener la variable de entorno
2 from azure.identity import DefaultAzureCredential
3 from azure.mgmt.datafactory import DataFactoryManagementClient
4 from azure.mgmt.datafactory.models import LinkedServiceResource, AzureTableStorageLinkedService, DatasetResource, AzureTableDataset
5
6 # Autenticación para Azure, cogemos la id de la suscripción de una variable de entorno que ha sido
7 #previamente creada
8 subscription_id = os.getenv('ID_AZURE_SUSCRPTION')
9 #Comprobamos que exista una variable de entorno con el id de la suscripción
10 if not subscription_id:
11     raise ValueError("Por favor, establece la variable de entorno ID_AZURE_SUSCRPTION antes de ejecutar este script.")
12
13 account_key = os.getenv('AZURE_STORAGE_ACCOUNT_KEY')
14 if not account_key:
15     raise ValueError("Por favor, establece la variable de entorno AZURE_STORAGE_ACCOUNT_KEY antes de ejecutar este script.")
16
17 #Punto de entrada para realizar operaciones
18 credential = DefaultAzureCredential()
19 adf_client = DataFactoryManagementClient(credential, subscription_id)
20
21 #Configuración del servicio
22 resource_group = 'TFGInformatica'
23 data_factory_name = 'fabricadadatostfg'
24 storage_account_name = 'datoscombustibles'
25
26 # Crear Linked Service para Azure Table Storage
27 azure_table_storage_linked_service = AzureTableStorageLinkedService(
28     connection_string=f"DefaultEndpointsProtocol=https;AccountName={storage_account_name};AccountKey={account_key};EndpointSuffix=core.windows.net"
29 )
30
31 ls_name = 'ServicioTablaAlmacenamiento'
32 adf_client.linked_services.create_or_update(
33     resource_group_name=resource_group,
34     factory_name=data_factory_name,
35     linked_service_name=ls_name,
36     linked_service=LinkedServiceResource(properties=azure_table_storage_linked_service)
37 )
38
39 print(f"Linked Service {ls_name} creado.")
40
41

```

Ilustración 47: Creación servicio "ServicioTablaAlmacenamiento"

```

1 import os #para poder obtener la variable de entorno
2 from azure.identity import DefaultAzureCredential
3 from azure.mgmt.datafactory import DataFactoryManagementClient
4 from azure.mgmt.datafactory.models import DatasetResource, RestResourceDataset, LinkedServiceReference
5
6 # Autenticación para Azure, cogemos la id de la suscripción de una variable de entorno que ha sido
7 #previamente creada
8 subscription_id = os.getenv('ID_AZURE_SUSCRIPCIÓN')
9 #Comprobamos que exista una variable de entorno con el id de la suscripción
10 if not subscription_id:
11     raise ValueError("Por favor, establece la variable de entorno ID_AZURE_SUSCRIPCIÓN antes de ejecutar este script.")
12
13 #Punto de entrada para realizar operaciones
14 credential = DefaultAzureCredential()
15 adf_client = DataFactoryManagementClient(credential, subscription_id)
16
17 # Datos de configuración
18 resource_group_name = 'TFGIInformatica'
19 data_factory_name = 'fabricadadatosfmg'
20 linked_service_name = 'ServicioRestAPIPreciOil'
21
22 localidades = ['Murcia', 'Madrid', 'Sevilla', 'Zaragoza', 'Oviedo', 'Palma', 'SantaCruzdeTenerife', 'Toledo', 'Santander',
23               'Valladolid', 'Barcelona', 'Merida', 'SantiagodeCompostela', 'Logronio', 'Valencia', 'Bilbao',
24               'Ceuta', 'Melilla']
25
26 for i in localidades:
27     # URL relativa (puedes cambiar esto según sea necesario)
28     if i == 'SantaCruzdeTenerife':
29         relative_url='Santa Cruz de Tenerife'
30     elif i == 'SantiagodeCompostela':
31         relative_url='Santiago de Compostela'
32     elif i == 'Logronio':
33         relative_url='Logroño'
34     else:
35         relative_url = i
36     # Crear el dataset para el servicio REST
37     rest_dataset_name = 'DataSetRest'+i
38     rest_dataset = RestResourceDataset(
39         linked_service_name=LinkedServiceReference(reference_name=linked_service_name, type='LinkedServiceReference'),
40         relative_url=relative_url,
41     )
42
43     adf_client.datasets.create_or_update(
44         resource_group_name=resource_group_name,
45         factory_name=data_factory_name,
46         dataset_name=rest_dataset_name,
47         dataset=DatasetResource(properties=rest_dataset)
48     )
49
50     print(f"Dataset REST '{rest_dataset_name}' creado.")
51

```

Ilustración 48: Creación Datasets servicio Rest

```

DataSetAPIPreciOil.py > ...
1  import os #para poder obtener la variable de entorno
2  from azure.identity import DefaultAzureCredential
3  from azure.mgmt.datafactory import DataFactoryManagementClient
4  from azure.mgmt.datafactory.models import DatasetResource, AzureTableDataset
5
6  # Autenticación para Azure, cogemos la id de la suscripción de una variable de entorno que ha sido
7  #previamente creada
8  subscription_id = os.getenv('ID_AZURE_SUSCRPTION')
9  #Comprobamos que exista una variable de entorno con el id de la suscripción
10 if not subscription_id:
11     raise ValueError("Por favor, establece la variable de entorno ID_AZURE_SUSCRPTION antes de ejecutar este script.")
12
13 #Punto de entrada para realizar operaciones
14 credential = DefaultAzureCredential()
15 adf_client = DataFactoryManagementClient(credential, subscription_id)
16
17
18
19 resource_group = 'TFGInformatica'
20 data_factory_name = 'fabricadatedatostfg'
21 linked_service_name = 'ServicioTablaAlmacenamiento' # Actualiza esto con el nombre de tu Linked Service
22
23 localidades = ['Murcia','Madrid','Sevilla','Zaragoza','Oviedo','Palma','SantaCruzdeTenerife', 'Toledo', 'Santander',
24               'Valladolid', 'Barcelona', 'Merida', 'SantiagodeCompostela', 'Logronio', 'Valencia', 'Bilbao',
25               'Ceuta', 'Melilla']
26
27 for i in localidades:
28     table_name = 'datosde'+i
29
30     # Crear un Azure Table Storage dataset
31     azure_table_dataset = AzureTableDataset(
32         linked_service_name={
33             "referenceName": linked_service_name,
34             "type": "LinkedServiceReference"
35         },
36         table_name=table_name # El nombre de tu tabla en Azure Table Storage
37     )
38
39     dataset_name = 'DataSetTable'+i
40     adf_client.datasets.create_or_update(
41         resource_group_name=resource_group,
42         factory_name=data_factory_name,
43         dataset_name=dataset_name,
44         dataset=DatasetResource(properties=azure_table_dataset)
45     )
46
47     print(f"Dataset para Azure Table Storage {dataset_name} creado.")
48

```

Ilustración 49: Creación datasets servicio tablas de almacenamiento

```

1 Pipeline.py > -
2 import os #para poder obtener la variable de entorno
3 from azure.identity import DefaultAzureCredential
4 from azure.mgmt.datafactory import DataFactoryManagementClient
5 from azure.mgmt.datafactory.models import (
6     PipelineResource,
7     CopyActivity,
8     DatasetReference,
9     TabularTranslator,
10    CopySource,
11    CopySink
12)
13 # Autenticación para Azure, cogemos la id de la suscripción de una variable de entorno que ha sido
14 #previamente creada
15 subscription_id = os.getenv('ID_AZURE_SUSCRPTION')
16 #Comprobamos que exista una variable de entorno con el id de la suscripción
17 if not subscription_id:
18     raise ValueError("Por favor, establece la variable de entorno ID_AZURE_SUSCRPTION antes de ejecutar este script.")
19
20 #Punto de entrada para realizar operaciones
21 credential = DefaultAzureCredential()
22 adf_client = DataFactoryManagementClient(credential, subscription_id)
23
24 resource_group = 'TFGInformatica'
25 data_factory_name = 'fabricadadatosf'
26
27 actividades_copia=[]
28 localidades = ['Murcia','Madrid','Sevilla','Zaragoza','Oviedo','Palma','SantaCruzdeTenerife','Toledo','Santander',
29               'Valladolid','Barcelona','Merida','SantiagodeCompostela','Logronio','Valencia','Bilbao',
30               'Ceuta','Melilla']
31 for i in localidades:
32     rest_dataset_name = 'DataSetRest'+i # Nombre del dataset REST
33     table_storage_dataset_name = 'DataSetTable'+i # Nombre del dataset de Table Storage
34
35     # Definición de la actividad de copia
36     copy_activity = CopyActivity(
37         name='ActividadDeCopia'+i,
38         inputs=[DatasetReference(reference_name=rest_dataset_name,type='DatasetReference')],
39         outputs=[DatasetReference(reference_name=table_storage_dataset_name,type='DatasetReference')],
40         source=[DatasetReference(reference_name=rest_dataset_name,type='DatasetReference')], # Usamos un CopySource genérico
41         sink=[DatasetReference(reference_name=table_storage_dataset_name,type='DatasetReference')], # Usamos un CopySink genérico
42         translator=TabularTranslator()
43     ) # Aquí puedes agregar opciones de configuración adicionales para la actividad de copia.
44
45     actividades_copia.append(copy_activity)
46
47
48
49 # Creación del pipeline
50 pipeline = PipelineResource(
51     activities=actividades_copia,
52 )
53
54 pipeline_name = 'Pipeline'
55 adf_client.pipelines.create_or_update(
56     resource_group_name=resource_group,
57     factory_name=data_factory_name,
58     pipeline_name=pipeline_name,
59     pipeline=pipeline
60 )
61
62 print(f"Pipeline {pipeline_name} ha sido creado.")
63

```

Ilustración 50: Creación del pipeline

```

1 Trigger.py > -
2 import os #para poder obtener la variable de entorno
3 from azure.identity import DefaultAzureCredential
4 from azure.mgmt.datafactory import DataFactoryManagementClient, ScheduleTriggerRecurrence, TriggerResource, TriggerPipelineReference, PipelineReference
5
6
7
8 # Autenticación para Azure, cogemos la id de la suscripción de una variable de entorno que ha sido
9 #previamente creada
10 subscription_id = os.getenv('ID_AZURE_SUSCRPTION')
11 #Comprobamos que exista una variable de entorno con el id de la suscripción
12 if not subscription_id:
13     raise ValueError("Por favor, establece la variable de entorno ID_AZURE_SUSCRPTION antes de ejecutar este script.")
14
15 #Punto de entrada para realizar operaciones
16 credential = DefaultAzureCredential()
17 adf_client = DataFactoryManagementClient(credential, subscription_id)
18
19 resource_group = 'TFGInformatica'
20 data_factory_name = 'fabricadadatosf'
21
22 trigger_name = 'Trigger'
23 pipeline_name = 'Pipeline'
24
25 pipeline_reference = TriggerPipelineReference(
26     pipeline_reference=PipelineReference(reference_name=pipeline_name, type='PipelineReference'),
27     parameters={}
28 )
29
30 recurrence = ScheduleTriggerRecurrence(
31     frequency='Hour', # Puede ser Minute, Hour, Day, Week, Month
32     interval=24, # Ejecutar una vez al día
33     start_time='2024-03-01T08:00:00Z', # Hora de inicio en formato UTC
34     end_time='2024-07-01T09:00:00Z',
35     time_zone='Romance Standard Time'
36 )
37
38
39 trigger = ScheduleTrigger(
40     recurrence=recurrence,
41     pipelines=[pipeline_reference]
42 )
43
44 trigger_resource = TriggerResource(
45     properties=trigger
46 )
47
48 # Crear o actualizar el trigger en Data Factory
49 adf_client.triggers.create_or_update(resource_group, data_factory_name, trigger_name, trigger_resource)
50
51 response = adf_client.triggers.begin_start(resource_group, data_factory_name, trigger_name)
52 response.result() # Espera a que la operación de inicio finalice
53
54 print(f"El trigger '{trigger_name}' ha sido activado.")
55

```

Ilustración 51: Creación del trigger

8.2 Power Query

```
let
    Origen = AzureStorage.Tables("datoscombustibles"),
    datosdeBarcelona1 = Origen[Name="datosdeBarcelona"][Data],
    #"Filas filtradas por fecha"=Table.SelectRows(datosdeBarcelona1, each
[Timestamp] > #datetime(2024, 4, 20, 0, 0, 0)),
    #"Expandir columna Content" = Table.ExpandRecordColumn(#"Filas filtradas
por fecha", "Content", {"Diesel", "DieselPremium", "Gasolina95", "Gasolina98",
"localidad", "DieselB", "GLP"}, {"Diesel", "DieselPremium", "Gasolina95",
"Gasolina98", "localidad", "DieselB", "GLP"}),
    #"Fecha descendente" = Table.Sort(#"Expandir columna
Content",{{"Timestamp", Order.Descending}}),
    #"Columnas quitadas" = Table.RemoveColumns(#"Fecha
descendente",{ "PartitionKey", "RowKey"}),
    #"Columnas reordenadas" = Table.ReorderColumns(#"Columnas
quitadas",{ "Timestamp", "localidad", "Diesel", "DieselPremium", "Gasolina95",
"Gasolina98", "DieselB", "GLP"}),
    #"Mismo sistema numerico" = Table.TransformColumnTypes(#"Columnas
reordenadas", {{ "Gasolina98", type number}, {"DieselB", type number}, {"GLP",
type number},{ "Diesel", type number},{ "DieselPremium", type
number},{ "Gasolina95", type number}}, "en-US"),
    #"Cambiar formato fecha" = Table.TransformColumnTypes(#"Mismo sistema
numerico",{{"Timestamp", type date}}),
    #"Columna de anulación de dinamización" =
Table.UnpivotOtherColumns(#"Cambiar formato fecha", {"Timestamp",
"localidad"}, "Atributo", "Valor"),
    #"Columnas reordenadas1" = Table.ReorderColumns(#"Columna de anulación de
dinamización",{ "localidad", "Timestamp", "Atributo", "Valor"}),
    #"Filas agrupadas" = Table.Group(#"Columnas reordenadas1",
{"localidad","Timestamp", "Atributo"}, {{ "Valor", each List.Average([Valor]),
type number}})
in
    #"Filas agrupadas"
```

Ilustración 52: Transformación y filtrado de los datos importados de Azure

```

let
    Origen = Excel.Workbook(Web.Contents("https://observatory.clean-hydrogen.europa.eu/sites/default/files/2023-09/Hydrogen%20production.xlsx"), null, true),
    #"Aggregated data_Sheet" = Origen{[Item="Aggregated data",Kind="Sheet"]}[Data],
    #"Eliminar null" = Table.Skip(#"Aggregated data_Sheet",6),
    #"Columnas quitadas null" = Table.RemoveColumns(#"Eliminar null",{"Column1"}),
    #"Encabezados promovidos" = Table.PromoteHeaders(#"Columnas quitadas null", [PromoteAllScalars=true]),
    #"Otras columnas quitadas" = Table.SelectColumns(#"Encabezados promovidos",{"Country", "Process type", "End-use", "Production capacity T/year", "Output T/year"}),
    #"Filas filtradas" = Table.SelectRows(#"Otras columnas quitadas", each ([Country] = "Spain"))
in
    #"Filas filtradas"

```

Ilustración 53: Limpieza y transformación datos sobre la producción [18]

```

let
    Origen = Table.Combine({#"datosdeZaragoza (2)", #"datosdeMurcia (2)", #"datosdeLogronio (2)", #"datosdeCeuta (2)", #"datosdeBarcelona (2)", #"datosdeBilbao (2)", #"datosdeMelilla (2)", #"datosdeMadrid (2)", #"datosdeMerida (2)", #"datosdeOviedo (2)", #"datosdePalma (2)", #"datosdeSantaCruzdeTenerife (2)", #"datosdeSantander (2)", #"datosdeSevilla (2)", #"datosdeSantiagodeCompostela (2)", #"datosdeToledo (2)", #"datosdeValladolid (2)", #"datosdeValencia (2)"}),
    #"Formato fecha" = Table.TransformColumnTypes(Origen,{{"Timestamp", type datetime}}),
    #"Ultimo dia" = Table.SelectRows(#"Formato fecha", let latest = List.Max(#"Tipo cambiado"[Timestamp]) in each [Timestamp] = latest)
in
    #"Ultimo dia"

```

Ilustración 54: Obtención precio medio combustibles


```

let
    Origen = Excel.Workbook(Web.Contents("https://observatory.clean-hydrogen.europa.eu/sites/default/files/2023-09/Hydrogen%20production%20costs_0.xlsx"),
    null, true),
    Sheet1_Sheet = Origen{[Item="Sheet1",Kind="Sheet"]}[Data],
    #"Columnas quitadas" = Table.RemoveColumns(Sheet1_Sheet,{"Column1"}),
    #"Otras columnas quitadas" = Table.SelectColumns(#"Columnas quitadas",{"Column2",
    "Column3", "Column4", "Column5"}),
    #"Filas superiores quitadas" = Table.Skip(#"Otras columnas quitadas",6),
    #"Encabezados promovidos" = Table.PromoteHeaders(#"Filas superiores quitadas",
    [PromoteAllScalars=true]),
    #"Filas filtradas" = Table.SelectRows(#"Encabezados promovidos", each ([Country] =
    "Spain")),
    #"Valor reemplazado" = Table.ReplaceValue(#"Filas filtradas","Grid
    electrolysis","Electrólisis",Replacer.ReplaceText,{"Technology"}),
    #"Valor reemplazado1" = Table.ReplaceValue(#"Valor reemplazado","Reforming with carbon
    capture","Reformado con captura de CO2",Replacer.ReplaceText,{"Technology"}),
    #"Valor reemplazado2" = Table.ReplaceValue(#"Valor reemplazado1","Renewable hydrogen
    directly connected to a renewable electricity generation source","Hidrógeno
    renovable",Replacer.ReplaceText,{"Technology"}),
    #"Valor reemplazado3" = Table.ReplaceValue(#"Valor
    reemplazado2","SMR","Reformado",Replacer.ReplaceText,{"Technology"}),
    #"Valor reemplazado4" = Table.ReplaceValue(#"Valor reemplazado3","Grid Fees","Tarifa
    de la red",Replacer.ReplaceText,{"Cost"}),
    #"Valor reemplazado5" = Table.ReplaceValue(#"Valor reemplazado4","Other OPEX","Otros
    gastos operativos",Replacer.ReplaceText,{"Cost"}),
    #"Valor reemplazado6" = Table.ReplaceValue(#"Valor
    reemplazado5","Taxes","Impuestos",Replacer.ReplaceText,{"Cost"}),
    #"Valor reemplazado7" = Table.ReplaceValue(#"Valor reemplazado6","Wholesale
    electricity costs","Costes de la red eléctrica",Replacer.ReplaceText,{"Cost"}),
    #"Valor reemplazado8" = Table.ReplaceValue(#"Valor reemplazado7","CO2 costs","Costes
    de CO2",Replacer.ReplaceText,{"Cost"}),
    #"Valor reemplazado9" = Table.ReplaceValue(#"Valor reemplazado8","NG costs","Costes de
    Gas Natural",Replacer.ReplaceText,{"Cost"}),
    #"Valor reemplazado10" = Table.ReplaceValue(#"Valor reemplazado9","Electricity
    costs","Costes de electricidad",Replacer.ReplaceText,{"Cost"}),
    #"Valor reemplazado11" = Table.ReplaceValue(#"Valor reemplazado10","Grid fees and
    taxes","Tarifas e impuestos de la red",Replacer.ReplaceText,{"Cost"}),
    #"Valor reemplazado12" = Table.ReplaceValue(#"Valor reemplazado11","Energy
    costs","Costes de energia",Replacer.ReplaceText,{"Cost"})
in
    #"Valor reemplazado12"

```

Ilustración 55: Limpieza y transformación datos sobre los costes [17]

```

let
    Origen = Excel.Workbook(Web.Contents("https://observatory.clean-hydrogen.europa.eu/sites/default/files/2023-09/Hydrogen%20demand_0.xlsx"), null, true),

    #"Annual consumption aggregated_Sheet" = Origen{[Item="Annual consumption aggregated",Kind="Sheet"]}[Data],

    #"Tipo cambiado" = Table.TransformColumnTypes(#"Annual consumption aggregated_Sheet",{{"Column1", type any}, {"Column2", type text}, {"Column3", type text}, {"Column4", type any}, {"Column5", type any}, {"Column6", type any}, {"Column7", type any}, {"Column8", type any}, {"Column9", type any}, {"Column10", type any}, {"Column11", type any}, {"Column12", type any}, {"Column13", type any}, {"Column14", type any}, {"Column15", type any}, {"Column16", type any}, {"Column17", type any}, {"Column18", type any}, {"Column19", type any}, {"Column20", type any}, {"Column21", type any}, {"Column22", type any}, {"Column23", type any}, {"Column24", type any}, {"Column25", type any}, {"Column26", type any}, {"Column27", type any}, {"Column28", type any}, {"Column29", type any}, {"Column30", type any}, {"Column31", type any}, {"Column32", type any}, {"Column33", type any}, {"Column34", type any}, {"Column35", type any}, {"Column36", type any}, {"Column37", type any}, {"Column38", type any}, {"Column39", type any}, {"Column40", type any}}),

    #"Columnas quitadas" = Table.RemoveColumns(#"Tipo cambiado",{"Column1"}),

    #"Filas superiores quitadas" = Table.Skip(#"Columnas quitadas",6),

    #"Encabezados promovidos" = Table.PromoteHeaders(#"Filas superiores quitadas",[PromoteAllScalars=true]),

    #"Tipo cambiado1" = Table.TransformColumnTypes(#"Encabezados promovidos",{{"Country", type text}, {"End-use", type text}, {"Number of plants", Int64.Type}, {"Clean consumption T/Y", type number}, {"Other consumption T/Y", type number}, {"Total consumption", type number}, {"Column7", type any}, {"Column8", type any}, {"Column9", type any}, {"Column10", type any}, {"Column11", type any}, {"Column12", type any}, {"Column13", type any}, {"Column14", type any}, {"Column15", type any}, {"Column16", type any}, {"Column17", type any}, {"Column18", type any}, {"Column19", type any}, {"Column20", type any}, {"Column21", type any}, {"Column22", type any}, {"Column23", type any}, {"Column24", type any}, {"Column25", type any}, {"Column26", type any}, {"Column27", type any}, {"Column28", type any}, {"Column29", type any}, {"Column30", type any}, {"Column31", type any}, {"Column32", type any}, {"Column33", type any}, {"Column34", type any}, {"Column35", type any}, {"Column36", type any}, {"Column37", type any}, {"Column38", type any}, {"Column39", type any}}),

    #"Filas filtradas" = Table.SelectRows(#"Tipo cambiado1", each ([Country] = "Spain")),


    #"Otras columnas quitadas" = Table.SelectColumns(#"Filas filtradas",{"Country", "Clean consumption T/Y", "Other consumption T/Y", "Total consumption"})

in
    #"Otras columnas quitadas"

```

Ilustración 56: Limpieza y transformación datos consumo

8.3 Informe de originalidad





Recibo digital

Este recibo confirma que su trabajo ha sido recibido por **Turnitin**. A continuación podrá ver la información del recibo con respecto a su entrega.

La primera página de tus entregas se muestra abajo.

Autor de la entrega:	ARABELA CARCELES CARRILLO
Título del ejercicio:	Turnitin Memoria Final (Moodle PP)
Título de la entrega:	TFGInformatica_ArabelaCarceles.pdf
Nombre del archivo:	18523_ARABELA_CARCELES_CARRILLO_TFGInformatica_Arabe...
Tamaño del archivo:	3.13M
Total páginas:	66
Total de palabras:	16,747
Total de caracteres:	96,781
Fecha de entrega:	03-jun.-2024 11:05a. m. (UTC+0200)
Identificador de la entre...	2394464960



Universidad Politécnica de Madrid
Escuela Técnica Superior de Ingeniería Informática
Centro de Ingeniería Informática

Trabajo Final de Grado

Desarrollo de una Plataforma de Monitorización y Control en Tiempo Real para Sistemas de Almacenamiento y Distribución de Hidrógeno en Empresas

Arabela Carceles Carrillo
18523 - Matrícula de Grado de Ingeniería de Sistemas de Información
Grado de Ingeniería de Sistemas de Información

Madrid, 2 de julio 2024

Derechos de autor 2024 Turnitin. Todos los derechos reservados.

Ilustración 57: Informe de originalidad