Universidad Politécnica
de Madrid

Escuela Técnica Superior de Ingenieros Informáticos

Grado en Ingeniería Informática

Trabajo Fin de Grado

Cuadro de Mando para el Análisis del Impacto del Uso de Prácticas Éticas en los Resultados de las Empresas

Autor: Alejandra Castillo Álvarez

Tutor(a): Fernando Pérez Costoya

Madrid, Junio 2024

Agradecimientos

A mis padres, hermanos, pareja, familia y amigas, por apoyarme en cada paso que he dado.

A mi tutor por acompañarme en la realización de este proyecto y a mis compañeros de carrera por todos estos años compartiendo esta experiencia juntos.

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: Cuadro de Mando para el Análisis del Impacto del Uso de Prácticas Éticas en los Resultados de las Empresas
Febrero 2024

Autor: Alejandra Castillo Álvarez

Tutor:
Fernando Pérez Costoya
Dpto. de Arquitectura y Tecnología de Sistemas Informáticos
Universidad Politécnica de Madrid

Resumen

En el contexto actual, donde la sostenibilidad y las prácticas éticas empresariales ganan cada vez más relevancia, este trabajo presenta un análisis del impacto de estas prácticas en el rendimiento de las empresas. A través del desarrollo de un cuadro de mando integral utilizando herramientas avanzadas como Power BI y AWS (Amazon Web Services), se ofrece a los inversores una herramienta visual y analítica para evaluar si las estrategias éticas afectan los resultados financieros de las empresas cotizadas.

El proyecto se ha desarrollado en varias fases, comenzando con la creación de una aplicación en AWS para automatizar la recolección y almacenamiento de datos financieros obtenidos de Yahoo Finance. Se han seleccionado empresas españolas basadas en su cumplimiento de todos los índices ESG (FTSE4Good IBEX®, IBEX Gender Equality® y la familia IBEX ESG®), comparando empresas con altos estándares éticos contra otras que solo tienen presencia en uno de estos índices y que además pertenecen al índice del IBEX 35. Esta selección permite analizar si las prácticas éticas y responsables impactan positivamente en el rendimiento y la estabilidad en el mercado.

La metodología aplicada para llevar a cabo el Proyecto incluye el diseño y ejecución de funciones AWS Lambda para la extracción diaria de precios, el almacenamiento de datos a través de Amazon DynamoDB y S3, y la integración final con Power BI para la visualización de datos. El cuadro de mando creado permite visualizar indicadores clave como precios diarios, volatilidad y rendimiento por acción para la realización de comparaciones.

Este TFG introduce una herramienta analítica innovadora para inversores interesados en el impacto social y económico de sus inversiones. La integración de análisis ético en las decisiones de inversión podría, por lo tanto, alentar a más empresas a adoptar prácticas responsables, contribuyendo así a un futuro empresarial más sostenible y ético.

Abstract

In the current context, where sustainability and ethical business practices are gaining more and more relevance, this paper presents an analysis of the impact of these practices on company performance. Through the development of a balanced scorecard using advanced tools such as Power BI and AWS (Amazon Web Services), investors are offered a visual and analytical tool to assess whether ethical strategies affect the financial performance of listed companies.

The project has been developed in several phases, starting with the creation of an application in AWS to automate the collection and storage of financial data obtained from Yahoo Finance. Spanish companies have been selected based on their compliance with all ESG indices (FTSE4Good IBEX®, IBEX Gender Equality® and the IBEX ESG® family), comparing companies with high ethical standards against others that only have a presence in one of these indices and belong to the IBEX 35 index. This selection makes it possible to analyse whether ethical and responsible practices have a positive impact on market performance and stability.

The methodology applied to carry out the Project included the design and execution of AWS Lambda functions for daily price extraction, data storage through Amazon DynamoDB and S3, and final integration with Power BI for data visualization. The dashboard created allows the visualization of key indicators such as daily prices, volatility, and performance per share for comparison purposes.

This TFG introduces an innovative analytical tool for investors interested in the social and economic impact of their investments. The integration of ethical analysis into investment decisions could therefore encourage more companies to adopt responsible practices, thus contributing to a more sustainable and ethical business future.

Tabla de contenidos

Tabla de ilustraciones

1 Introducción

Cada vez hay mayor concienciación por la sostenibilidad. En septiembre de 2015, las Naciones Unidas crearon los Objetivos de Desarrollo Sostenible (ODS) para abordar los desafíos sociales, económicos y ambientales mundiales, promoviendo el desarrollo sostenible. Gracias al surgimiento de los ODS se impulsan las finanzas éticas, una alternativa a las convencionales, ya que dejan de centrarse en maximizar el rendimiento económico-financiero, comenzando a invertir y gestionar recursos financieros para generar beneficios ambientales y sociales, creando un impacto positivo para el bienestar global. De la misma forma también se impulsan las empresas socialmente responsables, que son aquellas que adoptan estas prácticas éticas y sostenibles en toda su cadena de producción y comercialización.

Ante el nacimiento de estas nuevas prácticas, como inversor surge una duda, ¿afectará la adopción de estas nuevas prácticas al rendimiento en el mercado de las empresas? La hipótesis de la que se parte en este trabajo es, que, si las empresas pasan a ser mejores en sus prácticas, su rendimiento en el mercado será mayor y además serán más estables en cuanto a su comportamiento.

En este trabajo se va a realizar un cuadro de mando con la herramienta PowerBi que permita al inversor resolver de manera gráfica la duda que se plantea, para ello se creará una aplicación con la herramienta Amazon Web Services (AWS) que permitirá a través de los diferentes servicios que ofrece la automatización de la extracción de los datos necesarios para el análisis, además de la transformación y almacenamiento de estos. Los datos necesarios para la realización del trabajo se obtendrán de YahooFinance, y el lenguaje a utilizar para la generación de los diferentes scripts que sean necesarios para la implementación de la aplicación será Python.

Para realizar el proyecto se ha seleccionado un conjunto de 14 empresas en función de su grado de responsabilidad social. Para ello, se considerarán como empresas socialmente responsables a las que tienen presencia en los 3 índices ESG (FTSE4Good IBEX®, IBEX Gender Equality® y la familia IBEX ESG®), que muestran el grado de cumplimiento de las compañías cotizadas españolas en materia ambiental, de sostenibilidad y gobernanza. Como empresas convencionales se considerarán a empresas que pertenecen al índice del IBEX35 y que solo aparecen en 1 de los índices ESG.

Las empresas consideradas como responsables son: Applus, Atresmedia, CIE automotive, Ebro Foods, Ence y Gestamp, y las consideradas como empresas convencionales son: Amadeus, Acelormit, Endesa, Merlín, Naturgy, Rovi, Solaria y Unicaja.

2 Desarrollo

En este capítulo se va a detallar paso a paso la creación de un cuadro de mando que plasmará las medidas necesarias para que un inversor pueda decidir si es mejor invertir en empresas socialmente responsables. Para ello se desarrollará una aplicación que recogerá, almacenará y automatizará la carga de datos que alimentarán al cuadro de mando.

Para llevar a cabo el trabajo descrito anteriormente, se ha seleccionado el servicio Amazon Web Services Academy (AWS Academy) para la implementación de la aplicación. AWS Academy [1] es una plataforma de servicios en la nube ofrecida por Amazon Web Services (AWS) que proporciona una simulación la misma. AWS que es una herramienta que proporciona servicios informáticos en la nube, como almacenamiento, computación, bases de datos, redes entre otros, permitiendo desarrollar y desplegar aplicaciones de manera escalable y rentable sin necesidad de una infraestructura física propia [1].

Para la visualización y el análisis de los datos se ha seleccionado la aplicación PowerBi, que es una plataforma de análisis de datos y visualización de estos desarrollada por Microsoft que permite múltiples opciones para la inserción de los datos a utilizar [2]. Y el lenguaje de programación Python, ya que es un lenguaje muy intuitivo y dispone de múltiples librerías como yfinance, que es una librería que proporciona un histórico de datos financieros de todas las empresas.

Ilustración 1: Logo Amazon Web Services Academy

Ilustración 2: Logo PowerBi

Ilustración 3: Logo Python

2.1 Diseño de la estructura de la aplicación y tecnologías utilizadas

En este apartado se detallará la estructura de la aplicación que se va a desarrollar y las herramientas y tecnologías utilizadas en las etapas del desarrollo.

2.1.1 Estructura de la aplicación

Para el desarrollo de la aplicación se han seguido 4 procesos, que son la extracción de los datos, el almacenamiento, la conexión entre AWS Academy y PowerBi, y la visualización y el análisis de los datos. En la siguiente imagen podemos ver la arquitectura más detallada:

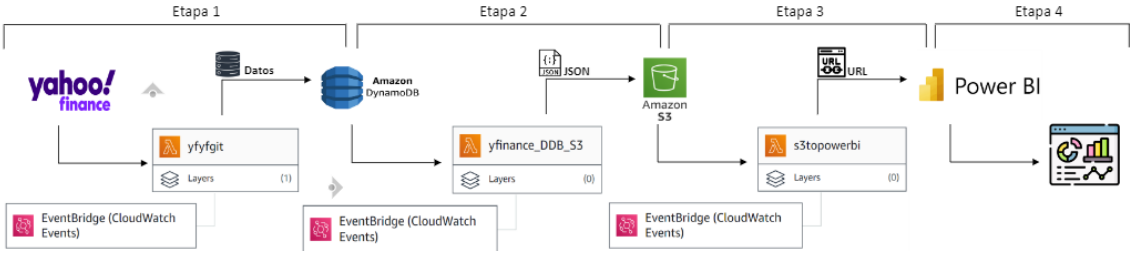


Ilustración 4: Diseño de la estructura de la aplicación. Fuente: elaboración propia

En la imagen se pueden distinguir 4 etapas, todas ellas trabajarán de manera secuencial para conseguir un proceso totalmente automatizado. La primera etapa, será la referente a la extracción de los datos necesarios. La segunda etapa se encargará de convertir los datos que se han extraído a un formato JSON sobre el que se pueda trabajar posteriormente. En la tercera etapa se creará la conexión entre AWS Academy y PowerBi a través de una URL. Finalmente, en la última etapa se creará un cuadro de mando en PowerBi a través de los datos extraídos que permitirá el análisis de datos.

El objetivo global de la arquitectura es que las etapas se ejecuten secuencialmente de manera que cada sábado a partir de las 10:30 de la mañana el usuario final con acceder al informe podrá observar el análisis que presenta el mismo actualizado en base a los datos extraídos en la semana actual.

El diseño secuencial y automatizado de la arquitectura de la aplicación garantiza una eficiencia operativa notable, minimizando la intervención manual y reduciendo el riesgo de errores. Cada etapa está planificada para asegurar que los datos recolecten y procesen correctamente, para posteriormente ser presentados de manera que faciliten la toma de decisiones. Este enfoque sistemático asegura la consistencia y la actualización regular de los datos, lo cual es crucial para mantener la relevancia y la precisión en el análisis financiero. Al ejecutarse todas las etapas automáticamente cada sábado, se proporciona a los usuarios un acceso constante y actualizado, permitiendo una interpretación dinámica y continua del mercado.

2.1.2 Herramientas utilizadas

Para el desarrollo global del proyecto, se han utilizado 3 herramientas como se ha mencionado anteriormente. Por un lado, Amazon Web Service Academy y Python y, por otro lado, PowerBi.

2.1.2.1 Amazon Web Services Academy

AWS Academy es un servicio que ofrece Amazon Web Services (AWS) a las instituciones de educación para la introducción a la computación en la nube a través de un laboratorio que simula AWS. La única diferencia de que presenta respecto a AWS es que presenta más limitaciones en cuanto a los recursos [1].

Los servicios que ofrecen tanto AWS como AWS Academy satisfacen las áreas de computación, almacenamiento, base de datos, redes y entrega de contenido, análisis, machine learning, seguridad, identidad y cumplimiento a través de diferentes servicios [1].

El motivo de la elección de esta herramienta se basa en su intuitiva interfaz y la multitud de servicios web que ofrece. La limitación que presenta es que es necesaria una cuenta de pago para su uso y que no tiene una conexión directa con PowerBi como es el caso de Microsoft Azure.

A continuación, se listan algunos de los servicios ofrecidos por AWS y AWS Academy más relevantes, entre los cuales se encuentran algunos de los utilizados para el desarrollo de este proyecto.

- ***Amazon EC2 (Elastic Compute Cloud)*:** proporciona capacidad informática escalable en la nube, permitiendo a los usuarios lanzar y gestionar servidores virtuales según sus necesidades [3].
- ***Amazon S3 (Simple Storage Service):*** es un servicio de almacenamiento en la nube diseñado para almacenar y recuperar grandes cantidades de datos de manera segura y rentable [4]. Este es uno de los servicios que se han seleccionado para la realización de la arquitectura debido a que ofrece grandes facilidades a la hora de crear otros recursos y almacenar resultados.
- ***Amazon RDS (Relational Database Service):*** ofrece bases de datos relacionales gestionadas en la nube, permitiendo a los usuarios configurar, operar y escalar bases de datos relacionales como MySQL, PostgreSQL, Oracle y SQL Server sin necesidad de gestionar la infraestructura subyacente [5].
- ***Amazon DynamoDB:*** es un servicio de base de datos NoSQL totalmente gestionado que proporciona rendimiento rápido y escalabilidad automática. Es útil para aplicaciones que requieren almacenamiento de datos de alto rendimiento y baja latencia [6]. Este servicio también es utilizado en este proyecto debido a la gran cantidad de datos que se recogen, ya que al ofrecer bases de datos NoSQL presenta una gran escalabilidad que permite un gran almacenamiento y un crecimiento incremental a medida que se ejecutan las funciones y va creciendo el conjunto de datos a lo largo del tiempo.

- ***Amazon Lambda:*** es un servicio de computación sin servidor que permite ejecutar código en respuesta a eventos y administrar automáticamente la infraestructura subyacente necesaria para ejecutar el código [7]. Las funciones Lambda y las Lambda Layer, que es la forma de importación de librerías que no son nativas en AWS [8], son otros de los recursos ofrecidos por Lambda. Permiten la creación de un cuerpo a través de un script, al que se le pueden añadir desencadenadores para programar su ejecución. Ambos servicios serán utilizados para la ejecución del proyecto.
- ***Amazon Route 53*:** es un servicio de sistema de nombres de dominio (DNS) escalable y altamente disponible que ayuda a dirigir el tráfico de internet a los recursos de AWS y a otras ubicaciones [9].
- ***Amazon CloudFront*:** es un servicio de entrega de contenido (CDN) que distribuye datos, videos, aplicaciones y APIs a través de una red global de centros de datos, mejorando la velocidad de entrega y reduciendo la latencia [10].
- ***Amazon SNS (Simple Notification Service*):** es un servicio de mensajería que permite enviar notificaciones desde la nube a dispositivos móviles, aplicaciones web y otros servicios distribuidos [11].
- ***Amazon SQS (Simple Queue Service):*** es un servicio de encolado de mensajes que permite desacoplar y escalar los componentes de una aplicación distribuida [12].
- ***Amazon ECS (Elastic Container Service):*** es un servicio altamente escalable para ejecutar, detener y administrar contenedores Docker en la nube de AWS [13].
- ***Amazon CloudWatch:*** es un servicio de supervisión y observación que proporciona datos y alertas operacionales en tiempo real para las aplicaciones y servicios de AWS [14]. Permite la creación de desencadenadores programables para las funciones Lambda, por lo que es otro de los recursos que se ha utilizado.
- ***AWS Glue:*** es un servicio de extracción, transformación y carga (ETL) que facilita la preparación y carga de datos para análisis en la nube de AWS [15].

2.1.2.2 Power Bi

Es una herramienta diseñada para business intelligence desarrollada por Microsoft cuya principal función es traducir grandes volúmenes de datos en gráficas, paneles o informes que facilitan la toma de decisiones. Las principales facilidades que proporciona PowerBi son la flexibilidad, adaptabilidad, colaboración y compartición [2].

PowerBi permite la transformación de los datos obtenidos para dar el formato deseado, y una vez el formato es el correcto proporciona herramientas que permiten realizar cálculos a partir de estos datos.

Las capacidades que ofrece son la importación de datos desde múltiples fuentes, tratamiento de esos datos y la posterior realización de gráficas a través del cruce o cálculo de los datos importados, además permite mantener los datos actualizados con la actualidad de la fuente de datos [16].

Es debido a la multitud de fuentes y a la gran cantidad de posibilidades que ofrece para el tratamiento de los datos y de visualización de estos por lo que se ha elegido PowerBi.

Generalmente, el resultado final que se obtiene con la herramienta es o un cuadro de mando o un informe que permite que el usuario final interactúe con la información de las diferentes dimensiones plasmadas en el informe.

En este trabajo se ha optado por la realización de un cuadro de mando, también conocido como dashboard. Un cuadro de mando es una única página que permite la visualización de diversas gráficas sobre un tema en concreto. Al estar formado de una sola página y el espacio es limitado, es importante decidir qué medidas son las que se quieren mostrar [17].

2.1.2.3 Python

Python es un lenguaje de programación de alto nivel utilizado para múltiples disciplinas como desarrollo software, análisis de datos o Machine Learning entre otras muchas [18].

Para la realización de este proyecto se ha elegido este lenguaje de programación debido a su simplicidad de entendimiento y su sintaxis, que al no ser tan estricta como otras, es muy fácil de aprender y de utilizar. Además, otro de los motivos de la elección ha sido la cantidad de librerías que proporciona, como yfinance, que ha sido crucial para el desarrollo del proyecto como se explicará más adelante.

2.1.3 Tecnologías utilizadas para las diferentes etapas del desarrollo de la aplicación

A continuación, se van a desglosar las diferentes tecnologías ofrecidas por las diferentes herramientas que se han seleccionado para el desarrollo de cada una de las fases del proyecto.

2.1.3.1 Etapa 1: extracción de los datos

El objetivo de esta etapa es coger los precios diarios de las 14 empresas seleccionadas sobre las que se hará el estudio. Estos datos posteriormente se insertarán en una tabla DynamoDB.

A continuación, se va a realizar una introducción a las diferentes tecnologías utilizadas para el desarrollo de esta fase.

- ***Yahoo! Finance:*** es un servicio de Yahoo! que ofrece información financiera como cotizaciones de bolsa, índices bursátiles, comunicados de prensa corporativos y financieros y herramientas para el manejo organizado de finanzas personales [19]. Python tiene una librería específica llamada "yfinance" que permite el acceso a todos los datos proporcionados por este servicio [20]. Se han obtenido de este servicio los precios diarios de las empresas seleccionadas para el informe, que se utilizarán para pintar gráficamente la evolución de estos y para el cálculo del rendimiento y la volatilidad.
- ***AWS Lambda*:** es un servicio automático proporcionado por AWS Academy que permite ejecutar código sin aprovisionar ni administrar servidores. Se encarga de realizar todas las tareas de administración de los recursos de computación, incluido el mantenimiento del servidor y del sistema operativo, el aprovisionamiento de capacidad y el escalado automático, así como las funciones de registro [21]. Dentro del servicio AWS Lambda encontramos tanto las funciones Lambda, que son las encargadas de la ejecución del código, y las Lambda Layer, que son las encargadas de alimentar a las funciones Lambda con las librerías que no están incluidas por defecto [8]. En esta fase la función Lambda que se ha implementado recoge los precios diarios y los almacena en una tabla DynamoDB. Además, para la ejecución de esta función Lambda se ha creado una Lambda Layer con la librería de Python yfinance.
- ***Amazon EventBridge:*** es un servicio proporcionado por AWS Academy que facilita la creación de arquitecturas de aplicaciones basadas en eventos, donde las aplicaciones pueden reaccionar y responder a eventos generados por una variedad de fuentes, como aplicaciones internas, servicios de terceros, plataformas SaaS (Software as a Service) [22]. El desencadenador hará que se ejecute la función lambda los sábados a las 9:30 de la mañana recolectando así los datos de mercado de toda la semana (de lunes a viernes que es cuando el mercado está abierto). El objetivo es que las 3 funciones que forman la aplicación se ejecuten sucesivamente.

- ***Amazon DynamoDB:*** es un servicio proporcionado por AWS Academy de base de datos NoSQL totalmente administrado que proporciona un rendimiento rápido y predecible con una escalabilidad perfecta. [23]. La tabla creada cuenta con 4 columnas, id de tipo String, fecha, precios y empresa, en ella se insertarán los datos obtenidos de la primera función lambda.

2.1.3.2 Etapa 2: almacenamiento de los datos

El objetivo de esta etapa consiste en almacenar los datos en un archivo JSON que permitirá posteriormente insertar los datos en PowerBi. Para esta etapa se han utilizado las siguientes tecnologías:

- ***Amazon S3 (Simple Storage Service):*** es un servicio de almacenamiento en la nube proporcionado por AWS Academy que permite almacenar y recuperar datos en cualquier momento desde cualquier ubicación a través de internet. S3 está diseñado para ser altamente escalable, duradero y seguro, y permite almacenar cualquier tipo de archivo [24]. Este se utilizará para almacenar el archivo zip con las librerías de Python necesarias y posteriormente para el almacenamiento del archivo JSON que se va a generar.
- ***JSON:*** es un formato de texto que forma parte del sistema de JavaScript y que se deriva de su sintaxis [25]. Tiene como objetivo el acceso, almacenamiento e intercambio de datos [25]. Se creará un archivo JSON en el que se almacenarán los datos de la tabla DynamoDB.
- ***AWS Lambda:*** para esta etapa se va a implementar una nueva función Lambda, que se encargará de pasar los datos ya almacenados en la tabla DynamoDB a un cubo S3 en formato JSON.
- ***Amazon EventBridge:*** el desencadenador en este caso ejecutará la función los sábados a las 9:45 de la mañana, con 15 minutos de separación con la anterior para asegurar que la primera función haya tenido tiempo de ejecutarse y los datos se hayan almacenado correctamente.

2.1.3.3 Etapa 3: conexión entre AWS Academy y PowerBi

El objetivo de esta etapa es crear el enlace entre AWS Academy y PowerBi. Para ello se creará una función Lambda que cree una URL que contenga el archivo JSON, de manera que a través de esta URL se pasarán los datos a PowerBi.

- ***AWS Lambda:*** la función que se va a implementar se encarga de realizar lo que se ha explicado previamente.
- ***Amazon EventBridge:*** el desencadenador de la función hará que se ejecute los sábados a las 10:00 de la mañana, de nuevo con 15 minutos de separación con la anterior, asegurando que la función que la precede se haya ejecutado y el archivo JSON esté listo para pasar a PowerBi.

2.1.3.4 Etapa 4: diseño del cuadro de mando y visualización de los datos

El objetivo de esta etapa es la creación del cuadro de mando que va a ver el cliente final, en el que se representarán los datos recolectados. Para esta fase se ha utilizado la herramienta PowerBi de Microsoft.

- ***PowerBi:*** como se ha mencionado anteriormente, es una plataforma de análisis de datos y visualización que facilita la toma de decisiones basada en datos [2]. Con esta herramienta se creará el cuadro de mando y además se programará su automatización para que el usuario final no tenga que encargarse de la actualización de los datos cada vez que acceda al informe.

2.2 Etapas del desarrollo de la aplicación

En este capítulo se va a desarrollar paso por paso el desarrollo de la aplicación. Para la realización del proyecto ha sido necesaria una cuenta de AWS Academy y otra de PowerBi.

En primer lugar, una vez hecho el login de la sesión en AWS Academy, esta es la visión general que se tiene.

Ilustración 5: Página de inicio AWS Academy

Desde contenidos, se clica en "Lanzamiento del Laboratorio para el alumnado de AWS Academy"

Ilustración 6: Lanzamiento del laboratorio de AWS Academy

Esta acción redirige a la pestaña que se muestra en la ilustración 4, para comenzar se debe clicar en el botón ▶ Start Lab .

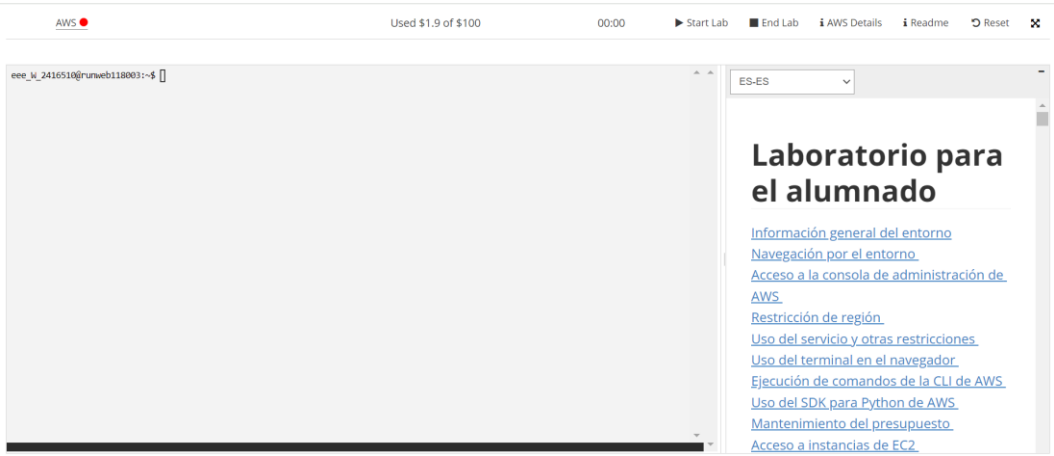


Ilustración 7: Pestaña de lanzamiento del laboratorio

Tras presionar este botón, el botón AWS, pasará de color rojo a amarillo AWS, y finalmente, cuando haya finalizado la carga del laboratorio a verde AWS. Entonces se clicará en él para acceder al área de trabajo. A partir de este momento se disponen de 4 horas para trabajar, en caso de que el tiempo se agotara no habría más que volver a pulsar el botón ▶ Start Lab.

En la página de inicio podemos ver los servicios que se han usado recientemente, además se dispone de un botón que despliega todos los servicios y una barra de búsqueda para buscar servicios específicos que no aparezcan en "visitados recientemente".

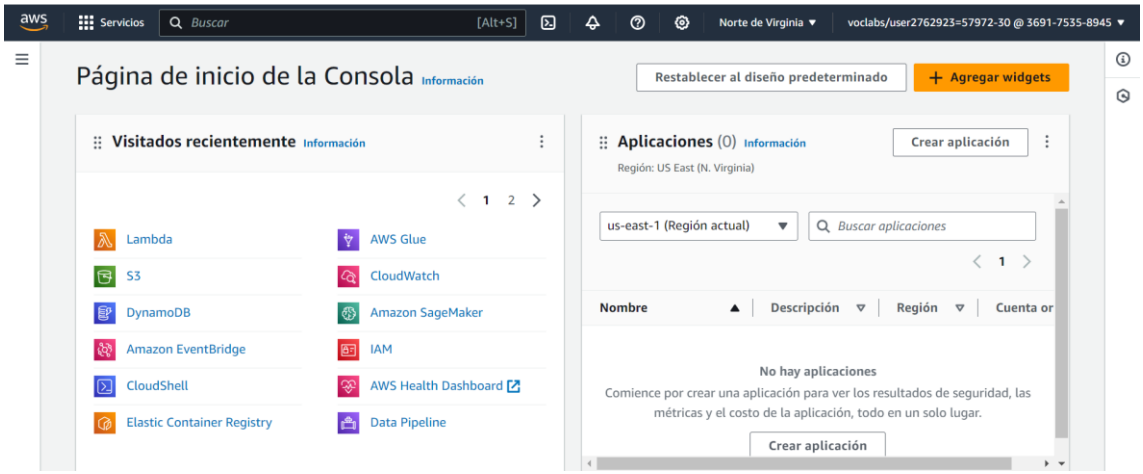


Ilustración 8: Página de inicio AWS Academy

Una vez dentro del laboratorio se puede acceder a todos los servicios ofrecidos y se procederá a la realización de la aplicación. Para ello como ya se ha explicado se crearán 3 funciones Lambda (desde el servicio Lambda) con sus respectivos desencadenadores, y para la primera función se creará también una Lambda Layer. Adicionalmente se usará para apoyar a estas funciones una tabla DynamoDB (del servicio DynamoDB) y dos cubos S3 (del servicio S3).

2.2.1 Etapa 1: extracción de los datos

La fase de extracción de datos consistirá en conseguir los datos con los que se realizará el informe final.

Para ello, se crea la función lambda “yfyfgit”. Para ello, se debe acceder al apartado funciones del servicio Lambda. Una vez dentro se clica en el botón “crear función”. Cada vez que se cree una nueva función se seguirán los pasos que se muestran a continuación.

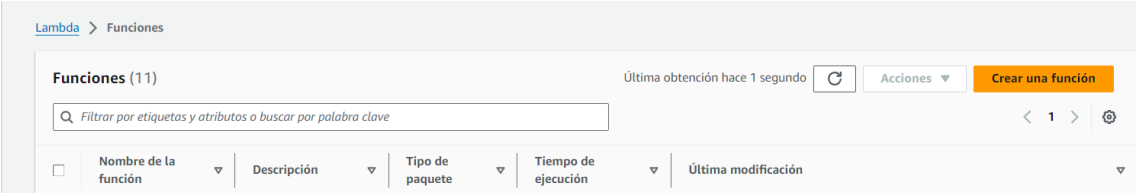


Ilustración 9: Visión del servicio Lambda

Una vez se haya abierto el panel de configuración se debe insertar el nombre de la función, el tiempo de ejecución y el rol en el que se quiere ejecutar. Al estar utilizando desde AWS Academy se debe seleccionar el rol laboratorio.

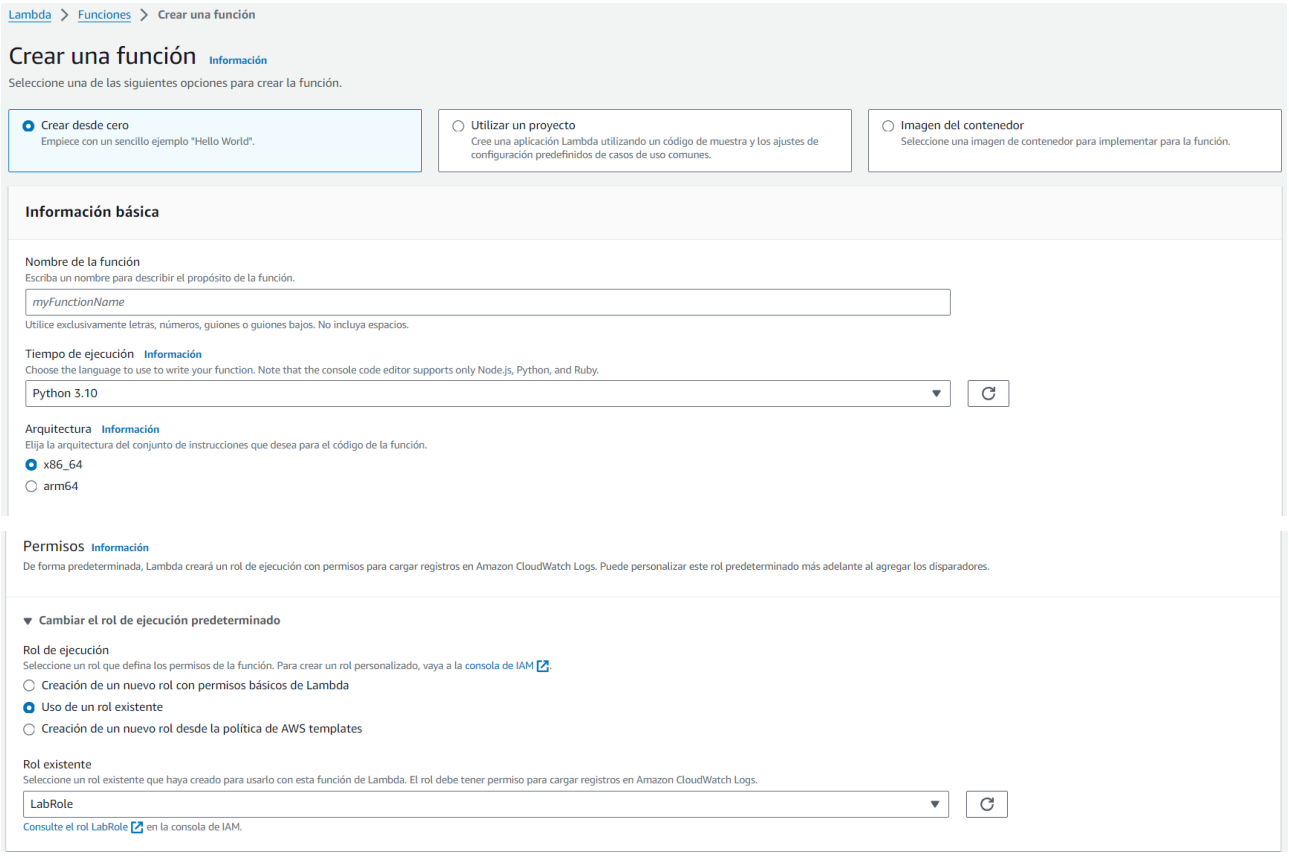


Ilustración 10: Configuración de una función Lambda

Para crear el cuerpo de la función se va a utilizar la biblioteca externa yfinance [19], esta no es una de las bibliotecas que se encuentran nativas dentro de Python en Amazon Web Service, es por eso por lo que hay que crear un LambdaLayer.

Las Lambda Layer con unos recursos de AWS Lambda que permiten compartir bibliotecas, dependencias personalizadas, y otros componentes entre múltiples funciones Lambda, sin necesidad de incluirlos en el paquete de despliegue de cada función individual. Esto ayuda a reducir el tamaño del paquete de la función Lambda y facilita la gestión del código que se usa en común entre diferentes funciones [8].

Para crear la LambdaLayer se debe crear un archivo comprimido con las bibliotecas que se desean insertar localmente. Se debe tener en cuenta que las librerías deben ser compatibles con la versión del lenguaje que se va a utilizar.

Debido a que generalmente los paquetes que contienen librerías son grandes, la mejor opción es subirlos a un Bucket S3 e importar los archivos a través de la URL del cubo en el que se ha almacenado. A continuación, se muestra cómo se ha realizado en este trabajo.

En primer lugar, se debe crear un cubo S3, para ello desde el servicio S3 clicamos el botón “crear bucket”:

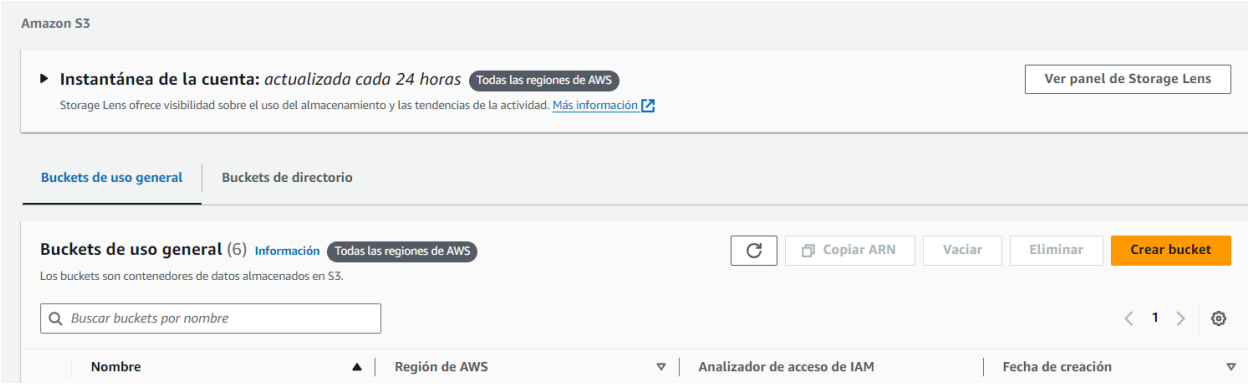


Ilustración 11: Visión general del servicio BucketS3

Al realizar esta acción se abre el panel de configuración del Bucket, en él lo único que se debe hacer es añadir el nombre que se le quiere dar al cubo y crearlo.

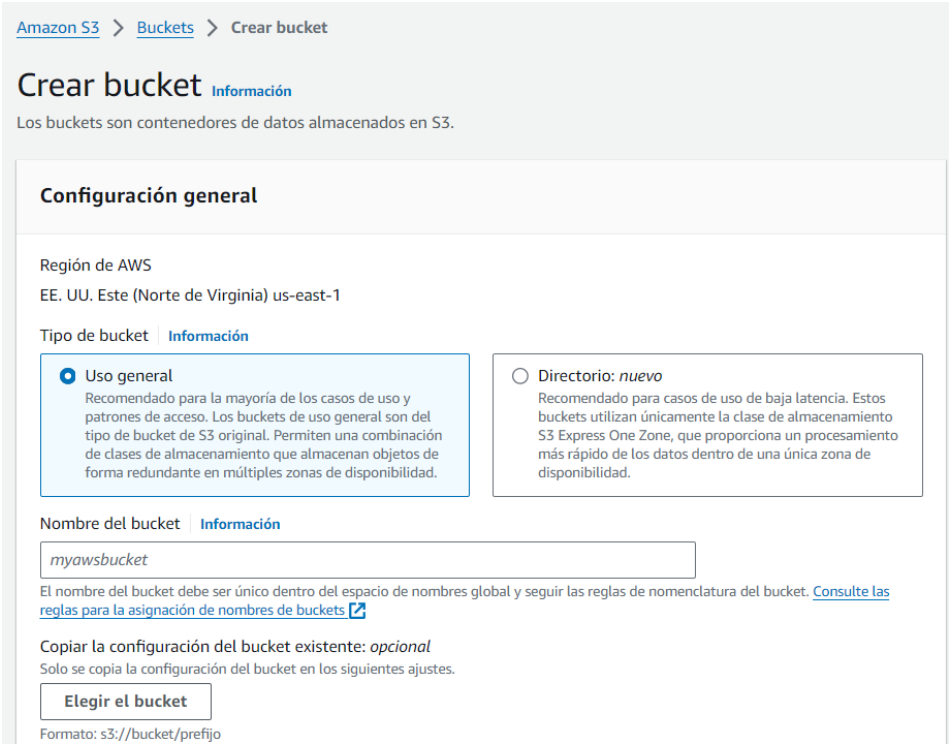


Ilustración 12: Configuración del cubo

Una vez creado el cubo se sube el archivo zip que contiene las librerías al bucketS3 creado, se selecciona la carpeta y se copia la URL de S3 con el botón "Copiar URL de S3".

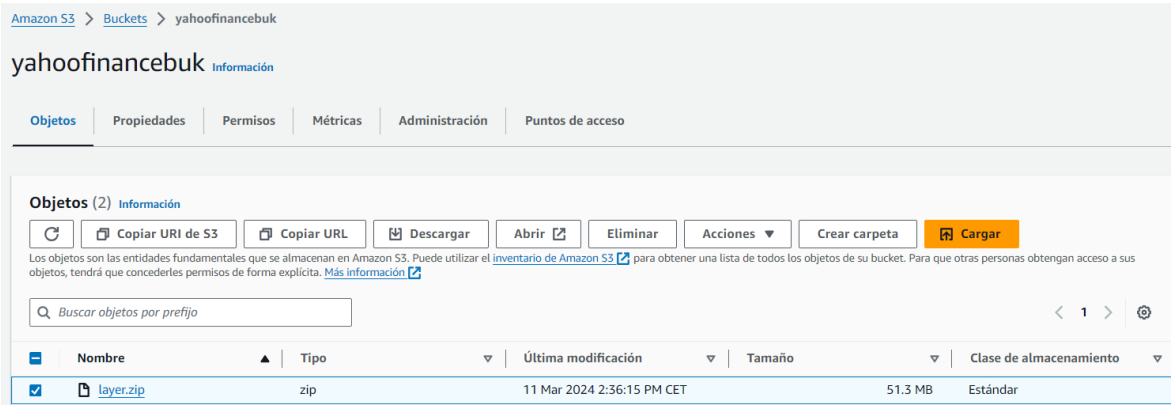


Ilustración 13: zip de las librerías necesarias en bucketS3

Cuando ya están las librerías en el Bucket, se abre el servicio Lambda y se crea una nueva capa, de la que hay que rellenar el nombre, seleccionar la opción "cargar un archivo de Amazon S3" y se pega la URL que se ha copiado anteriormente. Es recomendable elegir el tiempo de ejecución, que es el lenguaje y la versión con la que ejecutar. En este proyecto se ha utilizado *Python 10.0*.

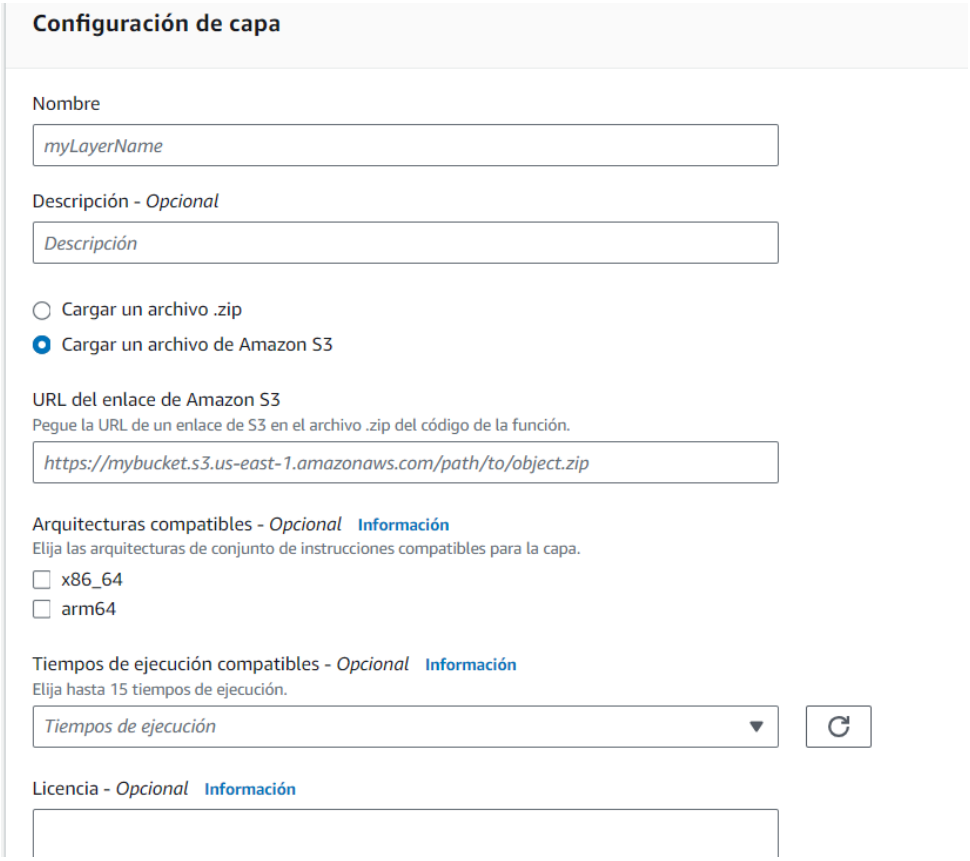


Ilustración 14: Configuración de una Lambda Layer

Finalmente, una vez creada la función y agregada a esta la Lambda Layer, se cuenta con un espacio para introducir el código que se quiere ejecutar “el cuerpo de la función”. Una vez se ha escrito el código se debe clicar en “Deploy” para guardar los cambios. En cuanto el “Deploy” haya cargado la función se puede ejecutar pulsando el botón “Test”.

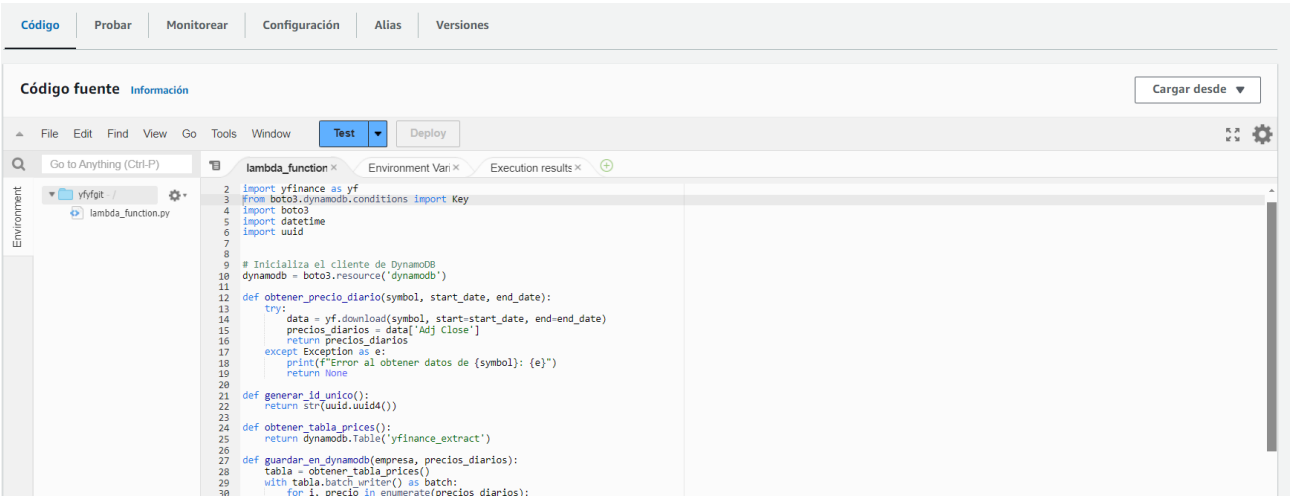


Ilustración 15: Cuerpo de la función Lambda

Esta función cuenta con un cuerpo que consiste en un script de Python adjuntado en el Anexo 1. Este script saca los datos del grupo de empresas seleccionado y los mete en una base de datos DynamoDB creando 4 columnas: la fecha, el nombre de la empresa, el precio diario y un identificador único para cada fila que se crea anteriormente en el mismo script. Este identificador único es necesario debido a que al ser una tabla incremental NoSQL cada fila debe tener su propio identificador para evitar la duplicación de las entradas.

Es script cuenta con los siguientes métodos:

- ***“obtener_precio_diario”*** utiliza la biblioteca yfinance para descargar el precio diario de una acción específica, identificada por su símbolo (symbol), entre dos fechas (start_date y end_date). Extrae el precio ajustado al cierre (Adj Close) para cada día en ese intervalo y lo devuelve. En caso de que ocurra un error durante la descarga, captura la excepción y devuelve None.
- ***“generar_id_unico”*** utiliza la biblioteca uuid para generar un identificador único (UUID). Este ID se usa para crear entradas únicas en una base de datos.
- ***“obtener_tabla_prices”*** configura y crea la tabla de DynamoDB llamada 'yfinance_extract' usando la biblioteca boto3. Esta función es utilizada para interactuar con la base de datos.
- ***“guardar_en_dynamodb”*** utiliza la función “obtener_tabla_prices” para obtener la tabla de DynamoDB, y para cada precio en la serie de precios diarios, crea un ítem que incluye un ID único, el nombre de la empresa, la fecha y el precio diario. Inserta cada ítem en la tabla utilizando un batch_writer, que optimiza y maneja múltiples operaciones de escritura en una sola llamada a la base de datos [26].

- ***lambda_handler:*** define una función que actúa como el método main para una función AWS Lambda, permitiendo que el script sea ejecutado como una respuesta a eventos o en un horario definido [27]. Para ello, define un diccionario de empresas y sus respectivos símbolos de bolsa, calcula las fechas de inicio y fin para la descarga de datos (los últimos 5 días hasta la fecha actual) e itera sobre cada empresa, descarga los precios diarios usando obtener_precio_diario y, si los precios están disponibles, los guarda en DynamoDB con guardar_en_dynamodb.

Debido a la cantidad de datos y a que en las funciones lambda hay un límite de tiempo de ejecución, para la primera carga de datos se ejecutó la función con solo 3 empresas cada vez. Para ello las fechas del script se ejecutaron como:

```
start_date = datetime.datetime(2015, 1, 1)
end  date = datetime.datetime(2024, 4, 30)
```

De esta manera se consigue que no exista duplicaciones de los datos y el código tenga una buena optimización y no supere el tiempo de ejecución.

Una vez realizada la carga inicial de datos se cambian las dos líneas de código referentes a las fechas para que únicamente se recojan los datos de lunes a viernes de cada semana, de forma que, a partir de ese momento, cada vez que se ejecute la función, únicamente se recojan los datos de los últimos 5 días y no se dupliquen datos.

```
end_date = datetime.datetime.now().date()
start_date = end_date - datetime.timedelta(days=5)
```

Es importante tener en cuenta que para la obtención de los datos de las empresas con yahooFinance hay que hacer referencia a ellas con los acrónimos que están establecidos en la api [19].

Una vez se ha comprobado que la función funciona correctamente se insertan los desencadenadores. Para este proyecto se ha decidido utilizar Cloud Watch EventBridge. Esta elección se debe a que CloudWatch permite ejecutar cada función en un horario preestablecido, lo que añade una capa adicional de control y precisión temporal al flujo de trabajo en AWS. Este enfoque permite que cada función en la cadena de procesamiento tenga un tiempo definido para completar su tarea antes de que la siguiente función inicie, lo cual es especialmente útil para asegurar que los procesos dependientes no intenten ejecutarse antes de que los datos necesarios estén disponibles y correctamente procesados. Esto minimiza errores como los archivos incompletos o la lectura de datos antiguos, y mejora la confiabilidad general del sistema automatizado, aprovechando la programación precisa que las expresiones cron ofrecen [14].

Debido a que los mercados cierran los viernes, pero los datos suelen tardar un poco más en estar disponibles en yfinance, los desencadenadores de las funciones se ejecutan los sábados de manera secuencial. La función “yfygit” es la primera en ejecutarse, ya que es la encargada de llevar a cabo la primera etapa del proceso.

Estos desencadenadores se configuran utilizando una función Cron, que indica que día y a qué hora se debe ejecutar la función. En este caso la regla indica: “30 9 * SAT *”, es decir la función se ejecutará los sábados a las 9:30 de la mañana. En la imagen a continuación se puede ver la configuración.

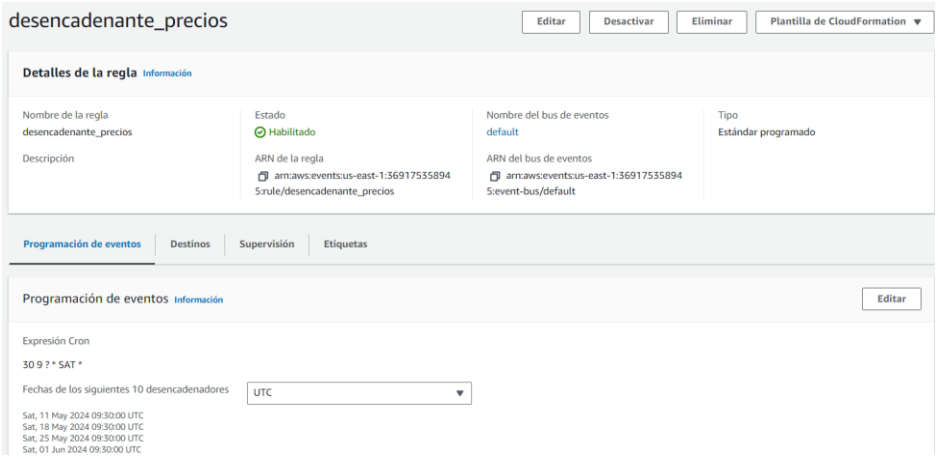


Ilustración 16: Desencadenante función "yfygit"

Esta es la vista general de la función ya completa, con el encabezado de la función labda, la lambda layer y el desencadenador:

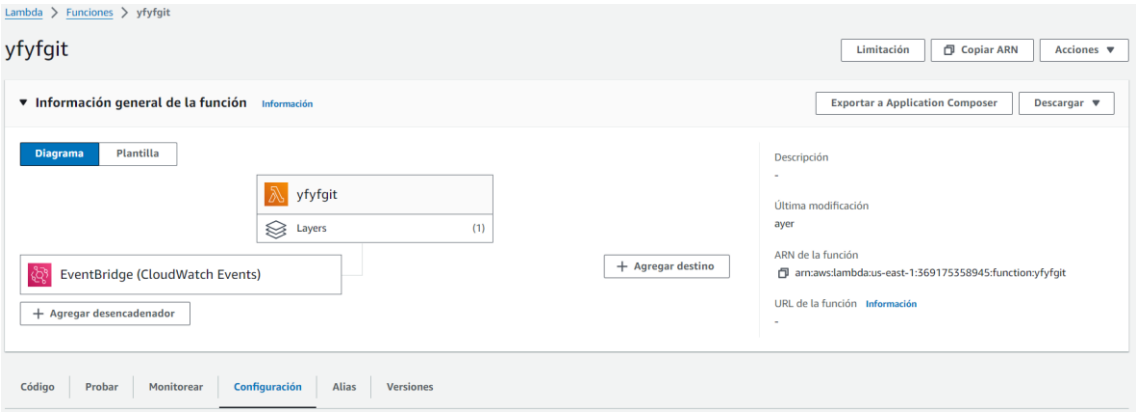


Ilustración 17: Función yfyfgit

Una vez se ejecuta la función el resultado se plasma la inserción los datos en la tabla DynamoDB. A continuación, se muestra una vista de la tabla creada en DynamoDB con los datos ya insertados:

id *(Cadena)*	Empresa	Fecha	Precio Diario
e701a4a7-24eb-4d35...	Amadeus	2020-02-11	72.7074050903...
9a851c03-191f-44b1...	Atresmedia	2021-05-31	3.02381443977...
3f746e85-7e27-4e2c-...	CIE Automo...	2023-11-22	23.6476993560...
607f120c-74cf-4281-...	Naturgy	2020-02-14	18.6928043365...
868f87e9-512f-4c74-...	Endesa	2022-10-07	13.7740640640...
0a96bbfc-5957-4ae2-...	CIE Automo...	2020-06-23	14.1006822586...
a26531fa-0080-4bf2-...	Applus	2023-11-14	9.86499977111...
0a8f4d12-c3c0-477d-...	Unicaja	2022-01-19	0.85170686244...
99a70f66-f8e7-46ab-...	CIE Automo...	2020-03-02	15.8411808013...
acab9174-0f05-4fda-...	Unicaja	2023-06-06	0.91850000619...
061c5566-e1c7-4e10...	Applus	2021-07-30	7.79620742797...

Ilustración 18: Tabla "yfinance_extract"

2.2.2 Etapa 2: almacenamiento de los datos

Debido a que PowerBi no tiene una conexión directa con AWS Academy, la opción más sencilla es la creación de un archivo JSON para la transferencia de los datos, ya que JSON está creado con el objetivo de facilitar la transferencia de datos. Para ello se crea la función ddb_to_s3, siguiendo los mismos pasos explicados en el apartado anterior a excepción de la creación de una Lambda Layer, ya que en este caso las librerías necesarias son nativas de Python en AWS Academy.

El objetivo de esta función consiste en meter los datos de la tabla DynamoDB en un archivo de tipo JSON y almacenar este archivo en un bucketS3. Esta función lambda está formada por un script de Python que se puede ver en el anexo 2, este está formado por una única función Lambda Handler. Dentro de esta función podemos encontrar las siguientes funciones y métodos.

- ***dynamodb = boto3.resource('dynamodb')*** crea un recurso de DynamoDB para interactuar con tablas.
- ***s3 = boto3.client('s3')*** crea un cliente S3 para operaciones de almacenamiento en el servicio S3.
- ***table = dynamodb.Table('yfinance_extract')*** selecciona la tabla 'yfinance_extract' de DynamoDB.
- ***response = table.scan()*** escanea la tabla para obtener todos los elementos y los almacena en data.
- Un ***bucle while*** se utiliza para manejar la paginación en DynamoDB. Si la respuesta incluye una clave LastEvaluatedKey, significa que no todos los datos fueron recuperados en una sola llamada debido a los límites de tamaño de respuesta de DynamoDB. Continúa escaneando hasta que se hayan recuperado todos los datos.
- ***json_data = json.dumps(data)*** convierte la lista de datos recuperados de DynamoDB a un string JSON.
- Define bucket_name y file_name para especificar el nombre del bucket de S3 y el nombre del archivo bajo el cual se almacenará el JSON.
- ***s3.put_object(Bucket=bucket_name,Key=file_name,body=json_data)*** almacena el string JSON en el bucket especificado bajo el nombre de archivo dado.

La función devuelve un objeto que indica el estado de la operación con un código de estado HTTP 200 y un mensaje que confirma que los datos han sido exportados correctamente a S3.

Una vez la función se encuentra en funcionamiento, se añade un desencadenador como en la anterior, pero en este caso se configura para que se ejecute 15 minutos después que la función “yfygit”, a las 9:45 de la mañana. Esto se ha decidido para dar margen de ejecución a la función y asegurar que cuando esta se ejecute ya estén los datos cargados y disponibles en la tabla de DynamoDB. A continuación, se muestra la configuración del desencadenador.

La automatización de esta función permitirá que el JSON contenga siempre los datos actualizados. La configuración del desencadenador es la siguiente:

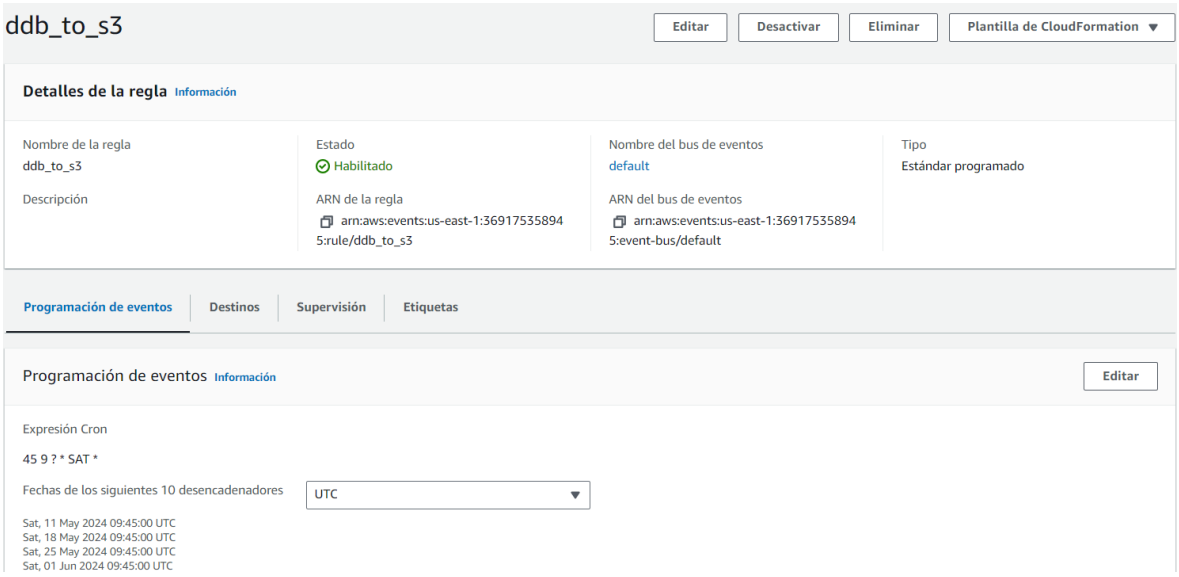


Ilustración 19: Desencadenador ddb_to_s3

Para ver el resultado de la ejecución de la función, accedemos a nuestro bucketS3 "yfinanceextract". Aquí podremos observar que se encuentra almacenado en el archivo JSON con los datos.

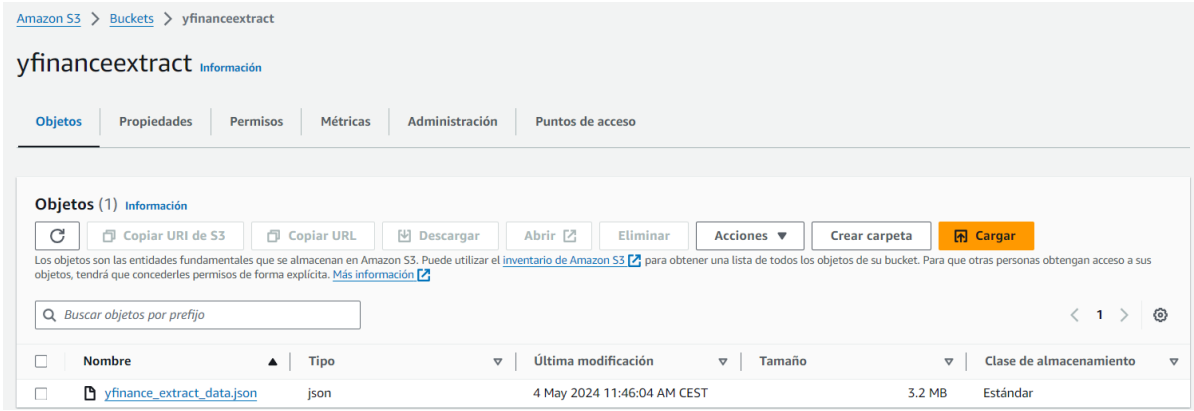


Ilustración 20: Archivo JSON almacenado en el bucketS3 creado

2.2.3 Etapa 3: conexión entre AWS Academy y PowerBi

Como ya se ha mencionado, debido a que no hay conexión directa entre AWS Academy y PowerBi, la opción más sencilla es pasarle los datos a través de una URL. Para ello se crea la última función Lambda de la aplicación de las etapas que involucran AWS Academy. Para la creación de esta función, se seguirán los mismos pasos que se han especificado en el primer apartado a excepción de la creación de una Lambda Layer, ya que de nuevo las librerías necesarias son nativas en Python de AWS Academy.

Esta función se encarga de subir el archivo JSON previamente almacenado en el BucketS3 a una URL. Esta acción permitirá a PowerBi adquirir los datos del JSON. De esta manera el cuadro de mando contará siempre con los datos actualizados.

El código que conforma la función lambda se puede ver en el Anexo 3. El script está formado por un único método, el Lambda Handler, en él podemos encontrar los siguientes métodos:

- ***s3 = boto3.client('s3'):*** crea un cliente para interactuar con el servicio Amazon S3.
- ***bucket_name = 'yfinanceextract':*** define el nombre del bucket de S3 del cual se recuperarán los datos.
- ***objects = s3.list_objects_v2(Bucket=bucket_name):*** utiliza el método list_objects_v2 para obtener una lista de todos los objetos en el bucket especificado.
- Se verifica si el diccionario objects contiene la clave 'Contents' y si hay objetos presentes. Esto se asegura de que el bucket no esté vacío.
- ***sorted_objects = sorted(objects['Contents'], key=lambda x: x['LastModified'], reverse=True):*** ordena los objetos en el bucket basándose en la fecha de última modificación de forma descendente. Esto significa que el objeto más reciente será el primero en la lista.
- ***file_name = sorted_objects[0]['Key']:*** extrae el nombre del archivo más reciente basándose en el ordenamiento anterior.
- ***obj = s3.get_object(Bucket=bucket_name, Key=file_name):*** obtiene el objeto S3 más reciente usando el nombre de archivo identificado.
- ***data = json.loads(obj['Body'].read()):*** lee el contenido del objeto (que se espera que esté en formato JSON), lo convierte de un flujo de bytes a texto y lo deserializa a un objeto Python usando json.loads.

Si se encuentran datos, la función retorna un objeto que incluye el estado HTTP 200, los datos en formato JSON, y el nombre del archivo. Esto se hace adecuadamente formateando el cuerpo como JSON y estableciendo el Content-Type apropiado. Si el bucket está vacío, se retorna un estado HTTP 404 con un mensaje indicando que no se encontraron archivos JSON.

Una vez se ha comprobado el funcionamiento de la función se añade el último desencadenador. Este se ejecuta a las 10:00 de la mañana para asegurar que la función anterior se ha ejecutado correctamente.

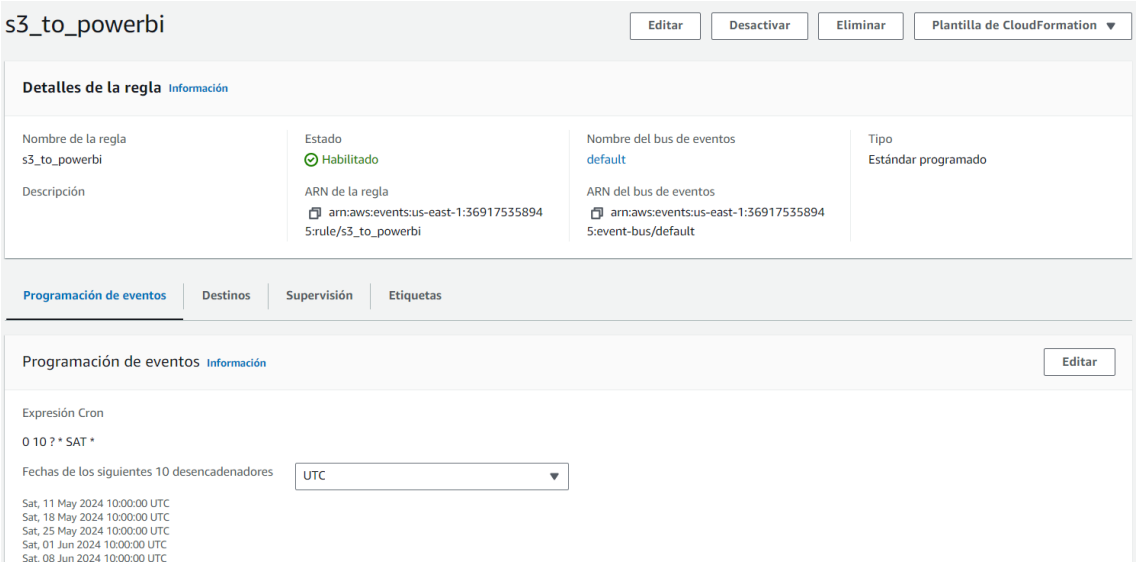


Ilustración 21: Desencadenado de la función "s3topowerbi"

En este caso como resultado obtenemos que en la URL de la función ahora alberga el JSON. Por lo que esta URL será la que se pasará a PowerBi para la obtención de los datos.

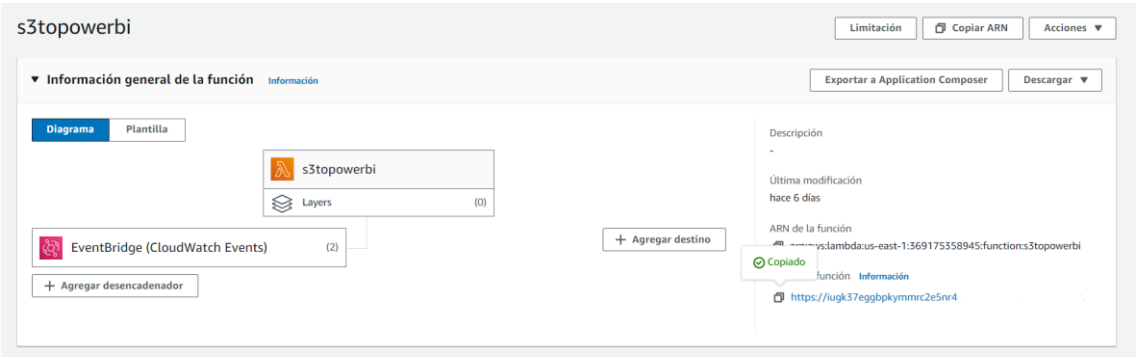


Ilustración 22: Resultado de la función s3topowerbi

2.2.4 Etapa 4: diseño del cuadro de mando y visualización de los datos

El objetivo de esta fase es la creación del cuadro de mando a través de los datos recogidos por la secuencia de funciones de AWS Academy. Primero se ha realizado una sección para dar contexto teórico al cuadro de mando y a los elementos que lo conforman y posteriormente se explicará la creación del cuadro paso por paso.

2.2.4.1 Contexto teórico del cuadro de mando

Se va a realizar una introducción a los diferentes conceptos que están involucrados en la creación del cuadro de mando.

- ***Finanzas éticas:*** son una nueva forma de ahorro e inversión, combina beneficios económicos con sociales y medioambientales. Se incorpora la ética a lo largo de todo el proceso de financiación, captando el ahorro de la ciudadanía y canalizándolo hacia la financiación de entidades y de empresas aplicando criterios éticos, sociales y ambientales. [28]
- ***Empresas socialmente responsables:*** aquellas que invierten en proyectos que repercuten positivamente sobre la calidad de vida de las personas, aportan una serie de beneficios sociales y promueven el desarrollo sostenible. Para la selección de estos proyectos, aplican criterios de evaluación ético-sociales para estudiar la responsabilidad ética, social y medioambiental de los mismos que solicitan financiación, garantizando así dar apoyo económico a proyectos con alto impacto social, ambiental y cultural. [29]
- ***Precio diario:*** es el valor de cierre ajustado de una acción en un mercado de valores durante una jornada de negociación. Este precio refleja el último nivel al que compradores y vendedores acordaron realizar un intercambio y se ajusta por acciones corporativas como dividendos, divisiones de acciones y derechos de suscripción, proporcionando una imagen precisa del valor del activo al final del día [30].
- ***Volatilidad:*** es la medida que evalúa la variabilidad, inestabilidad, oscilación en la trayectoria de un activo durante un tiempo determinado.

  $$Volatilidad\ (V) = \sqrt{\frac{\sum_{i=0}^{n}(p_i - R)^{-2}}{n}},$$

  donde R es el rendimiento y n el período de tiempo sobre el que se calcula el rendimiento [31].

- ***Rendimiento:*** es la medida que evalúa los beneficios o pérdidas que tiene una acción en un periodo de tiempo determinado con base en el capital inicial que se invirtió en el activo [32]. Se calcula como la media de los precios diarios obtenidos en un período de tiempo.

  $$Rendimiento\ (R) = \frac{\sum_{i=0}^{n} p_i}{n},$$

  donde p es el precio diario y n el período de tiempo sobre el que se calcula el rendimiento.

El objetivo es que el cuadro de mando muestre la evolución tanto de los precios diarios como de la volatilidad y del rendimiento. Además, mostrará los precios

máximos y mínimos registrados, el rendimiento máximo y la volatilidad mínima. Estas medidas han sido elegidas ya que para la elección de la inversión de un activo se debe seleccionar el que mejor rendimiento tenga, pero a la vez menos volatilidad, ya que cuanto mayor rendimiento mayores beneficios se obtienen y cuando menor volatilidad más estables son los beneficios.

2.2.4.2 Creación del cuadro de mando

En esta sección se van a explicar los diferentes pasos utilizados para la creación del cuadro de mando.

Una vez obtenidos los datos se procede al diseño de un cuadro de mando, que como se ha explicado anteriormente es un tablero que presenta diversos objetos visuales (gráficos) que permiten la visualización de indicadores clave para la toma de decisiones. [33]

En el cuadro de mando que se va a crear plasmará el comportamiento en el mercado de un conjunto de empresas que han sido seleccionadas para el estudio. Para ello como se ha mencionado anteriormente se va a utilizar la herramienta PowerBi.

En primer lugar, se deben insertar los datos. Debido a que en PowerBi no hay una conexión directa con AWS Academy como la hay con otros servicios como Azure, se ha optado por insertar los datos a través de una URL. Para ello creamos un nuevo informe:

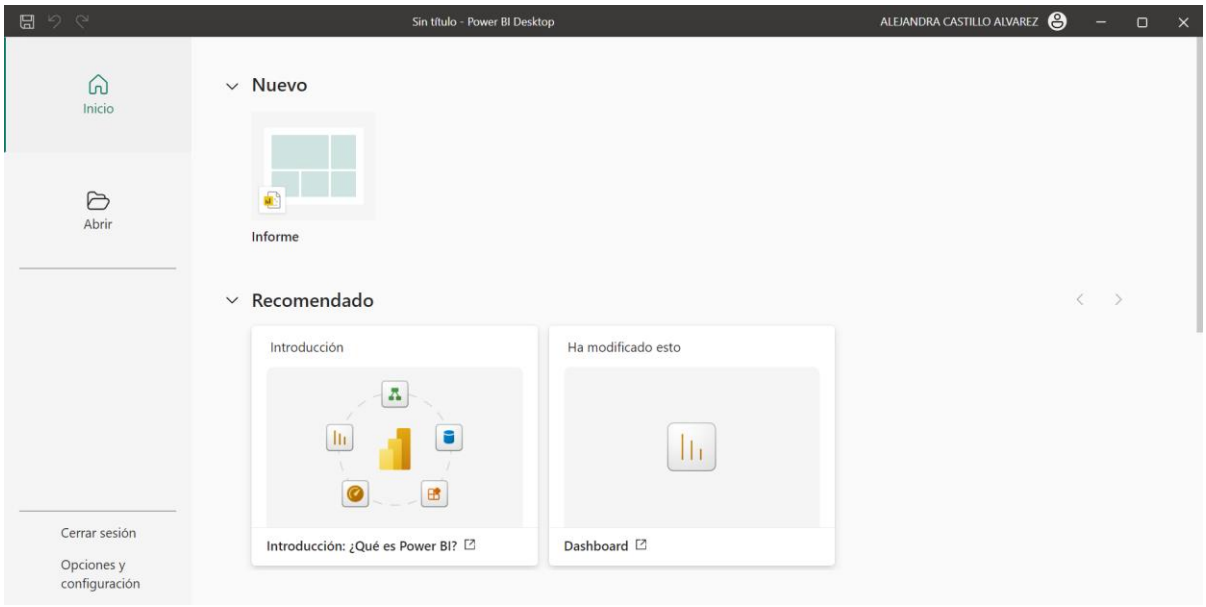


Ilustración 23: Página de inicio de PowerBi

Una vez creado el nuevo informe se obtiene una pantalla como la que se muestra a continuación.

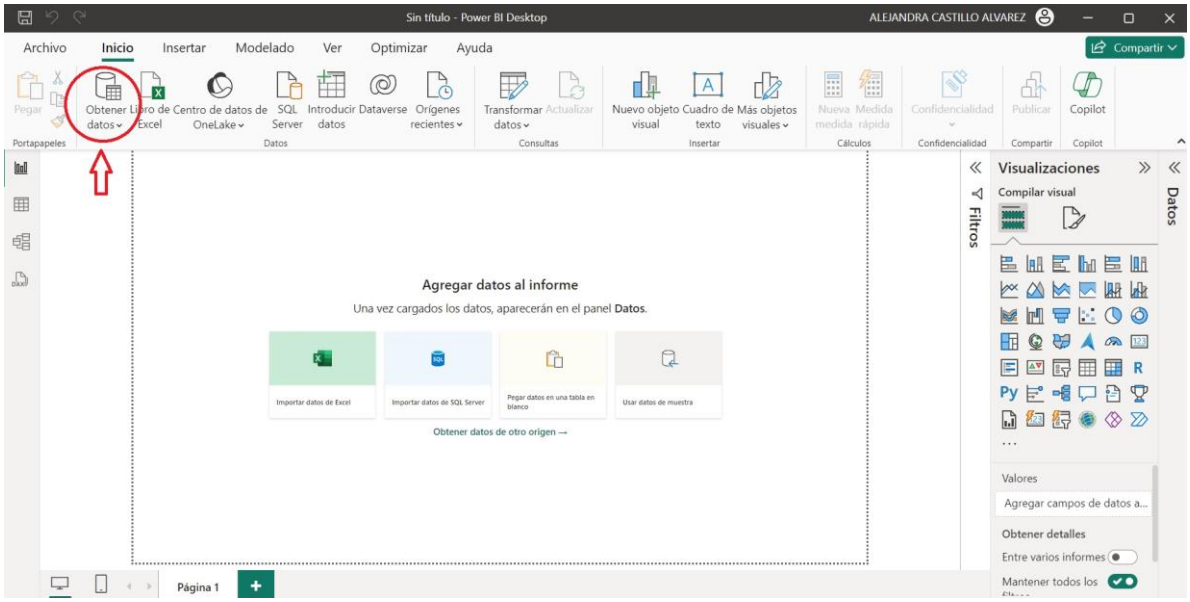


Ilustración 24: Página del proyecto creado

En esta pantalla se selecciona la opción “obtener datos” donde se muestra en la imagen anterior “ilustración 24”. A través de esta acción se obtiene un menú desplegable con las múltiples opciones que ofrece PowerBi para la inserción de datos:

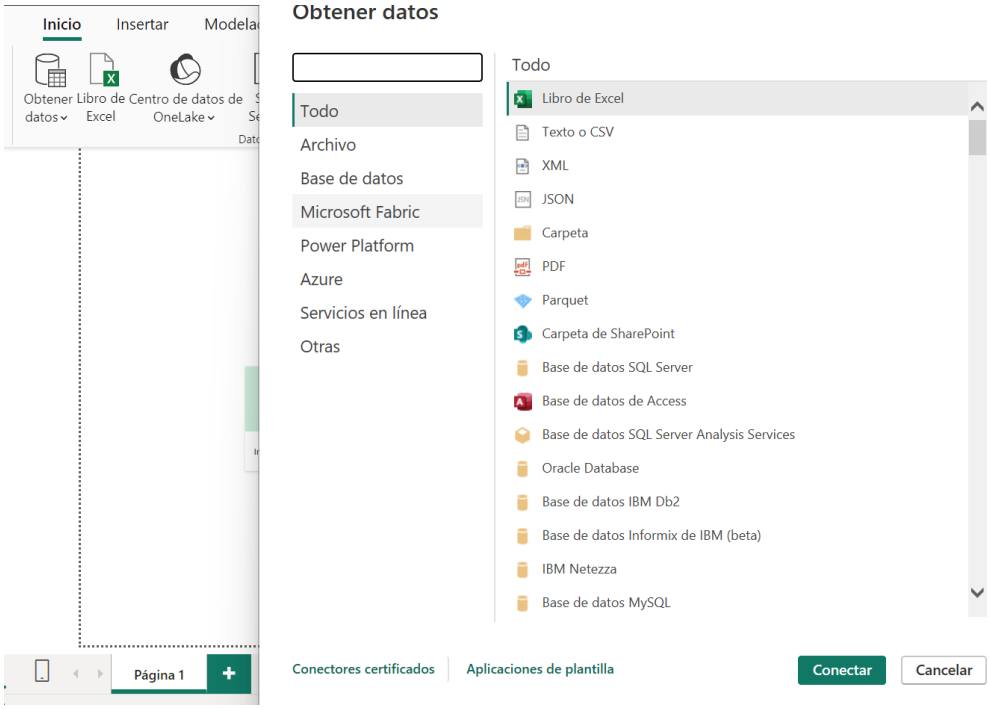


Ilustración 25: Opciones de obtención de datos desde PowerBi

En este menú desplegable se selecciona la opción “web” y pegaremos la URL en la que se encuentran los datos del JSON (la obtenida en la [función Lambda s3topowerbi] creada en la etapa 3):

Ilustración 26: Obtención de datos en PowerBi a través de una URL

Cuando se acepte la operación, los datos se cargarán en Power Query, que permite dar forma a los datos obtenidos cambiándoles el tipo, el número de decimales, etc. [34]. En este caso los pasos seguidos para el formato de los datos han sido los siguientes:

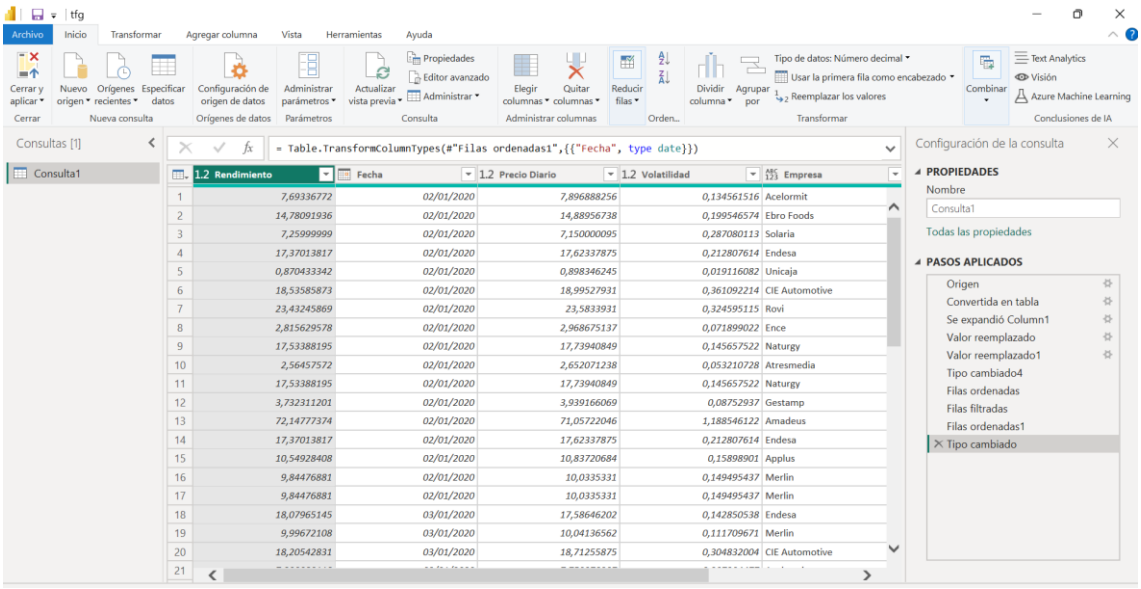


Ilustración 27: Pasos seguidos para la transformación en Power Query

El paso más importante en la transformación es cambiar el formato de los datos relativos a los precios en el momento de la inserción, ya que en JSON se separa la parte entera de la parte decimal con ".", y en Power Query hay que separar la parte entera de la decimal con ",". Para ello se utiliza el botón  de la barra de herramientas en la pestaña transformar.

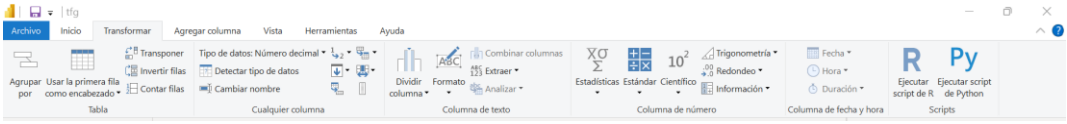
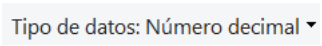


Ilustración 28: Barra de herramientas de Power Query

Luego se cambia el tipo de datos a decimal 

. En el caso de la columna fecha, se cambia el tipo de dato a fecha.

Una vez se tienen los datos en el formato deseado se guarda y se cierra esta pestaña pasando a la ventana principal que nos permitirá la creación del informe.

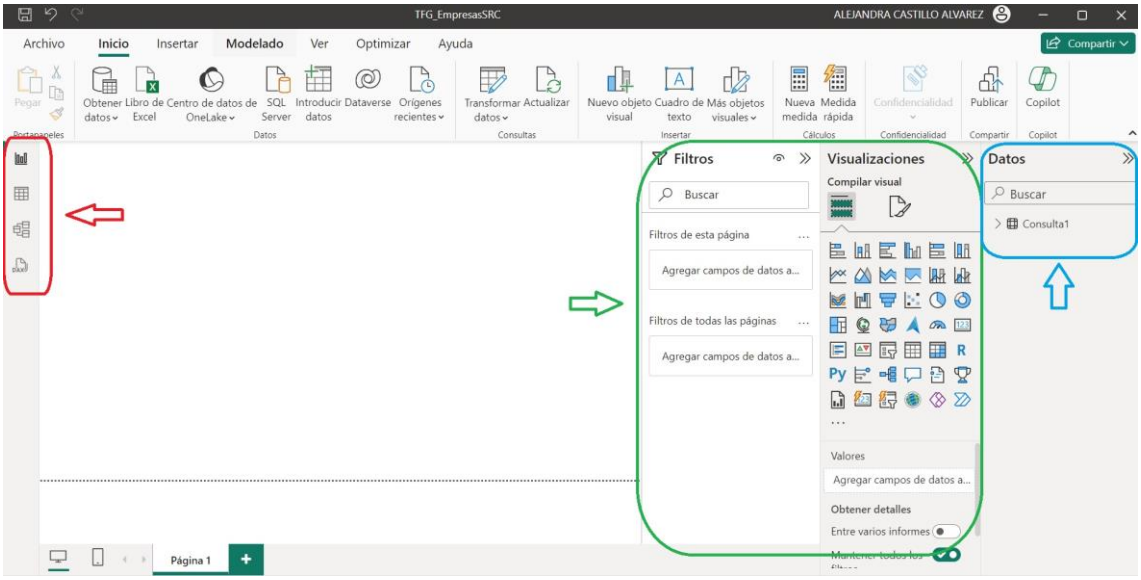


Ilustración 29: Página de creación del informe

A la derecha están los diferentes objetos visuales que se pueden elegir (rodeados en verde), y la consulta contiene todas las columnas de datos que hemos formateado previamente (rodeado en azul). A la izquierda disponemos de los 4 paneles que forman la herramienta (rodeado en rojo), el Dashboard (que es en el que está ahora mismo la vista), una tabla que nos permitirá realizar operaciones sobre los datos que hay en nuestra consulta, una vista general de las tablas que forman el modelo y finalmente funciones DAX, que son funciones proporcionadas para operar con los datos para su análisis, en PowerBi hay más de 250 funciones [35].

El funcionamiento consiste en ir añadiendo objetos visuales que muestren correctamente los datos para la posterior toma de decisiones.

En primer lugar, en la pantalla tabla se realizan las operaciones que se consideran necesarias. Para ello se pulsa el botón agregar medida o el de agregar columna y se insertan funciones DAX para obtener los cálculos necesarios. Se han utilizado las siguientes funciones:

```
Rendimiento móvil = AVERAGEX(
        DATESINPERIOD('Consulta1'[Fecha],LASTDATE('Consulta1'[Fecha]), -
        30, DAY),
         CALCULATE(SUM('Consulta1'[Precio Diario]))
)
VolatilidadFinal = CALCULATE(
        STDEV.P('Consulta1'[Precio Diario]),
        DATESINPERIOD('Consulta1'[Fecha], LASTDATE('Consulta1'[Fecha]), -
        30, DAY)
)
```

Para la creación de los objetos relativos a estas funciones y al precio diario, en primer lugar, se selecciona el tipo de objeto visual y luego se agregan los campos que se quieren pintar. Para la creación de este cuadro de mando se han creado 3 gráficos de líneas que muestran la evolución de los precios diarios, el rendimiento y la volatilidad. Para ello se ha seleccionado el objeto gráfico de líneas, en el eje X se ha seleccionado la fecha y en el eje Y se han seleccionado los precios diarios, el rendimiento y la volatilidad respectivamente. Además, en la leyenda se ha añadido el campo empresas para que en el objeto se pinte cada una de estas medidas para cada empresa y posteriormente se pueda filtrar por empresa las visualizaciones. A continuación, se muestra el gráfico respectivo a la evolución de los precios diarios:

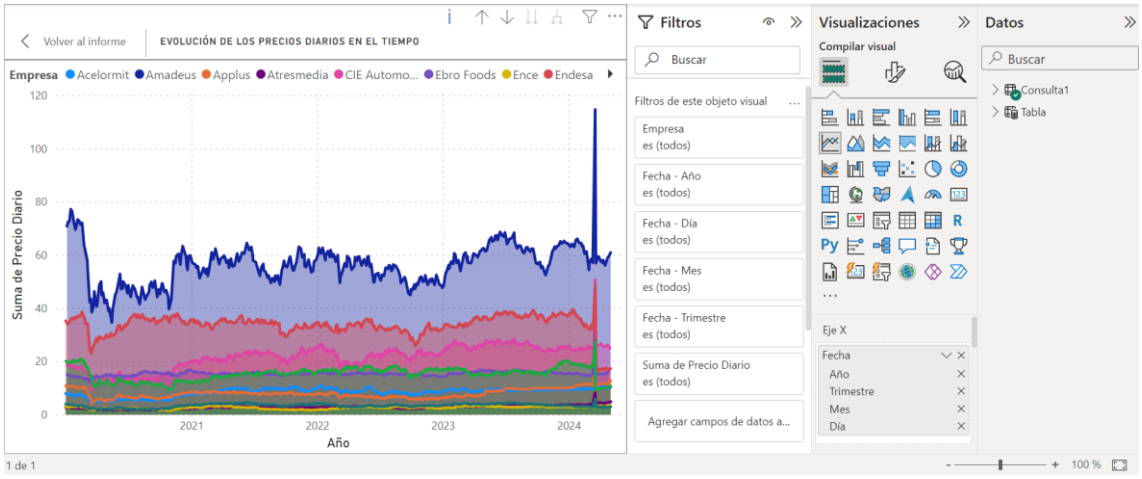


Ilustración 30: Gráfico de evolución de los precios diarios

Además, PowerBi proporciona herramientas que permiten la personalización tanto de los campos del objeto visual como de la apariencia de los objetos visuales que se crean:

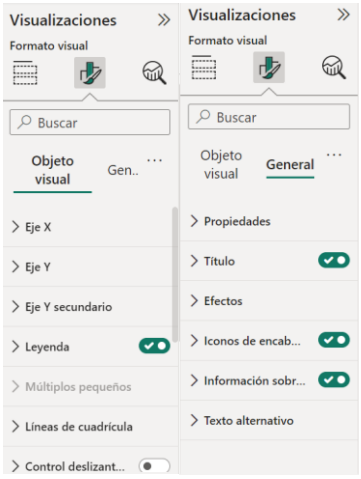


Ilustración 31: Configuración de objetos visuales

Por otro lado, se han añadido también 4 objetos visuales que indican el Precio máximo, el precio mínimo, el rendimiento máximo y la volatilidad mínima. Para la creación de estos objetos primero ha habido que calcular sus respectivos datos, para ello se han utilizado las funciones que se muestran a continuación:

```
PrecioMax = MAX('Consulta1'[Precio Diario])

PrecioMin = MIN('Consulta1'[Precio Diario])

RendimientoMax = MAX('Consulta1'[Ren])

VolatilidadMin = MIN('Consulta1'[volatilidad])
```

Una vez creadas las medidas, se utiliza el objeto etiqueta para cada uno de los valores. Una vez creados se les da formato cambiando el fondo y el título obteniendo el siguiente resultado:

Ilustración 32: Etiquetas de las medidas del cuadro de mando

Los colores seleccionados han sido el rojo, el verde, el azul y el marrón debido a que estos colores son los colores de la sostenibilidad. El rojo hace referencia a la responsabilidad social, el azul a la responsabilidad económica y el verde y el marrón a la responsabilidad medioambiental [36].

Adicionalmente, se han añadido dos objetos que crean segmentación de datos en forma de menú desplegable que permiten filtrar por un lado según la fecha, permitiendo seleccionar año, trimestre y mes, y por otro lado en función de la empresa. Esto permite al usuario final un análisis mucho más exacto pudiendo seleccionar un momento en el tiempo determinado y un conjunto de empresas determinadas.

Ilustración 33: Segmentación de datos por fecha y empresa

Finalmente, se ha añadido un cuadro de texto para poner el título del cuadro de mando y se ha insertado una imagen obteniendo el siguiente resultado:

Ilustración 34: Título del cuadro de mando

Estos dos últimos objetos se han añadido también en color verde que hace referencia a la sostenibilidad y al medioambiente.

Una vez finalizado el cuadro de mando, se va a automatizar la actualización de este. Para ello, el primer paso es publicar el informe, para ello se debe pulsar el botón “publicar”, y se elige la opción para publicar en “Mi área de trabajo”.

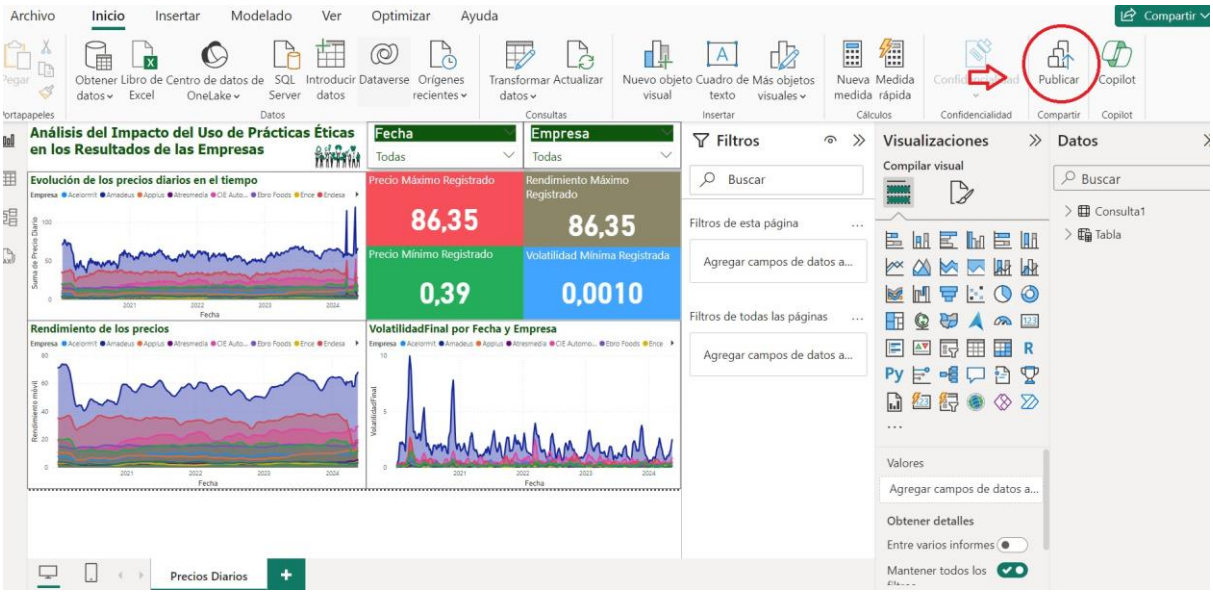


Ilustración 35: Botón de publicación del informe

Una vez se haya publicado, se accede a la cuenta de PowerBi desde un navegador.

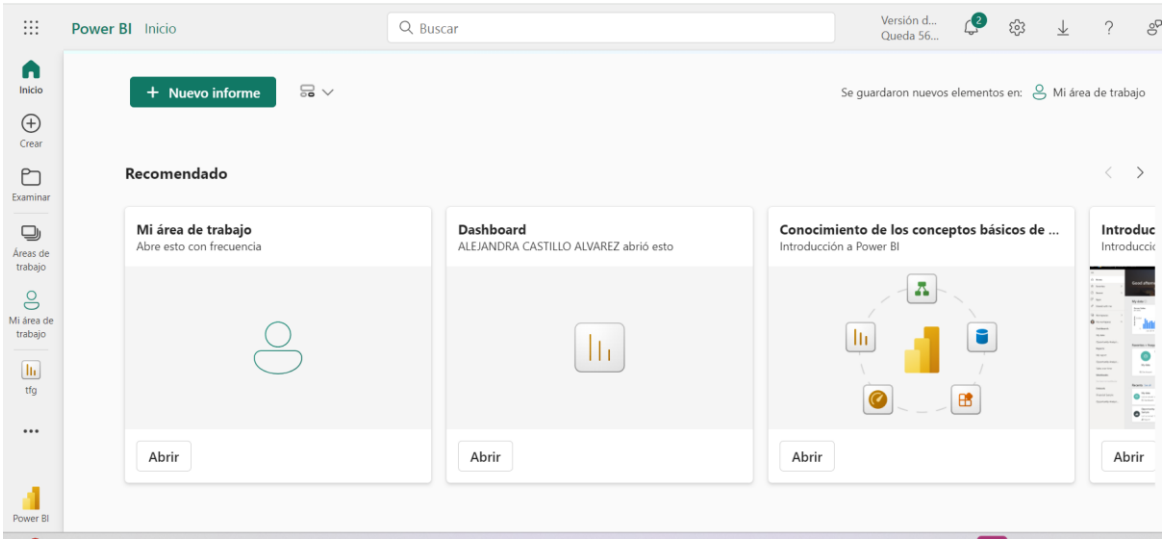


Ilustración 36: Página de inicio de PowerBi desde el navegador

Desde esta pantalla accederemos al área de trabajo, donde se encuentra el informe creado y su modelo semántico, que representa un origen de datos listos para la elaboración de informes, la visualización, la detección y el consumo.

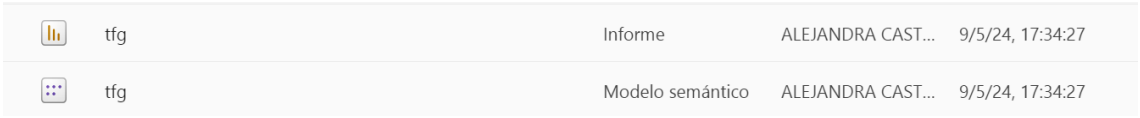


Ilustración 37: Informe creado y su modelo semántico

En el modelo semántico al seleccionarlo o pasar el cursor por encima, aparecerán 3 puntos. Se debe clicar en los 3 puntos y saldrá un menú, donde elegiremos la opción configuración.

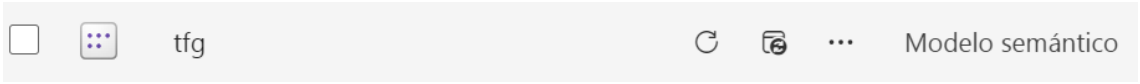


Ilustración 38: Acceso al menú desplegable

Una vez dentro de la configuración se obtiene una pantalla como la que se muestra a continuación.

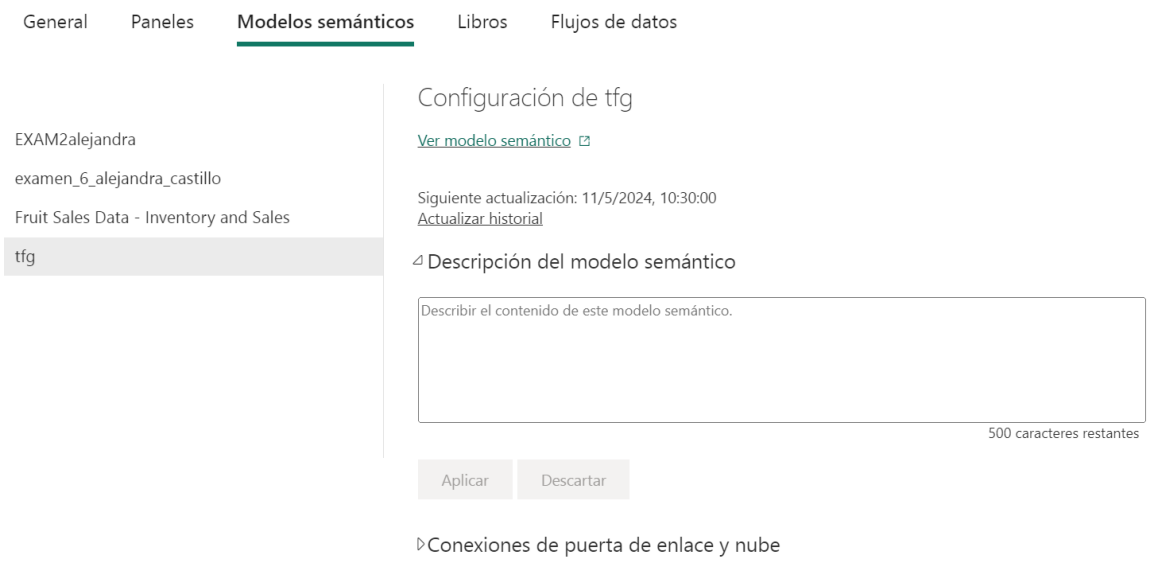


Ilustración 39: Pantalla de configuración del modelo semántico

Una vez dentro del menú de configuración, se busca la opción actualizar, que permitirá configurar la actualización del cuadro de mando eligiendo la frecuencia de actualización, la zona horaria, qué días y a qué horas se quiere actualizar el cuadro.

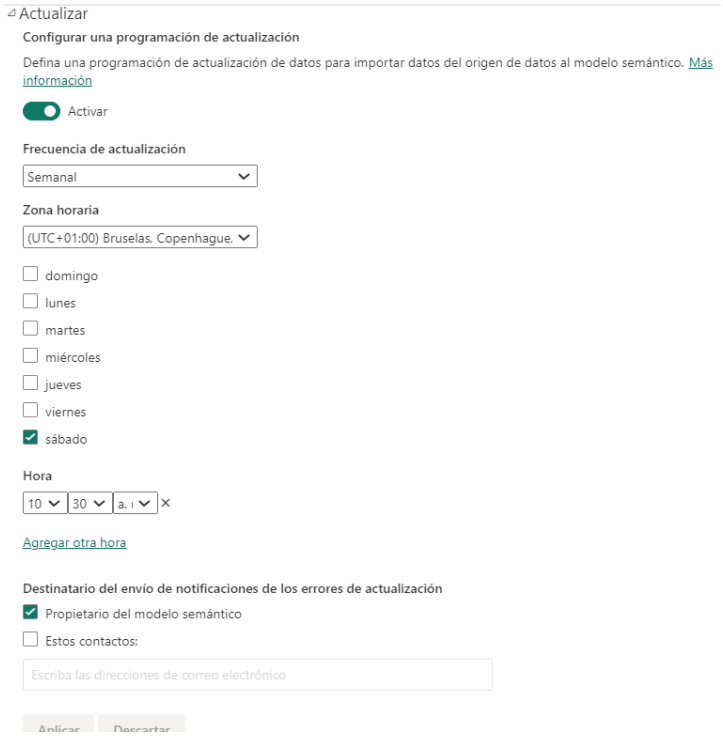


Ilustración 40: Configuración de la actualización del cuadro de mando

Se ha configurado la actualización semanalmente para que se realice los sábados a las 10:30 de la mañana. La hora de actualización se ha seleccionado en base a la ejecución de las funciones Lambda, es decir se ha seleccionado una hora posterior al último desencadenador para que cuando se ejecute el cuadro de mando estén los datos listos para importar.

Por último, de nuevo desde el área personal, se accede al informe. Se pulsa el botón “Archivo”, en el menú que se despliega la opción “Insertar informe” y finalmente en “Publicar en la web”. Esto generará una URL donde se encuentra el cuadro de mando, que será desde donde acceda el usuario final. De esta manera el usuario puede utilizar todas las funcionalidades del cuadro de mando, pero sin editar este, y obtiene una mejor visión de los datos en comparación a la que se obtiene desde la aplicación durante su generación.

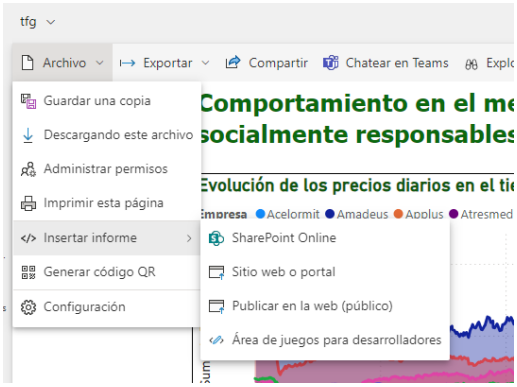


Ilustración 41: Publicación del informe en la web

Una vez se haya publicado el cuadro de mando desde la web se obtiene la siguiente pantalla:

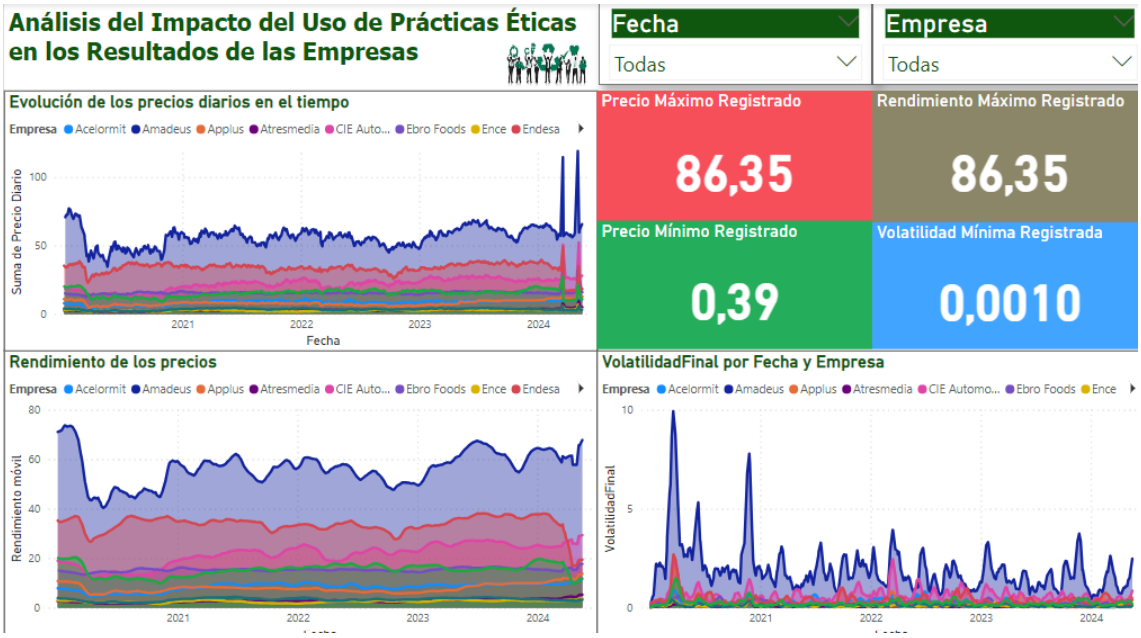


Ilustración 42: Cuadro de mando resultado

Gracias a la segmentación de datos en función de la fecha y de la empresa los gráficos variarán dando al usuario una visión más exacta de los datos que quiere conocer.

3. Resultados y conclusiones

El resultado es una herramienta que consiste en un cuadro de mando dirigido al inversor para apoyar en estudio del comportamiento de las empresas en el mercado a través de los precios diarios, el rendimiento de estos y su volatilidad pudiendo distinguir si las empresas con prácticas éticas proporcionan un mayor rendimiento y una menor volatilidad que las empresas convencionales. Además, este cuadro de mando se actualizará después de cada semana tras el cierre de los mercados gracias a la aplicación diseñada en AWS Academy y a la automatización ofrecida por PowerBi. La vista final de este cuadro de mando es la mostrada en la "ilustración 42".

A continuación, se muestra un ejemplo de uso para comparar las empresas Amadeus (convencional) y EbroFoods (responsable) en el año 2021:

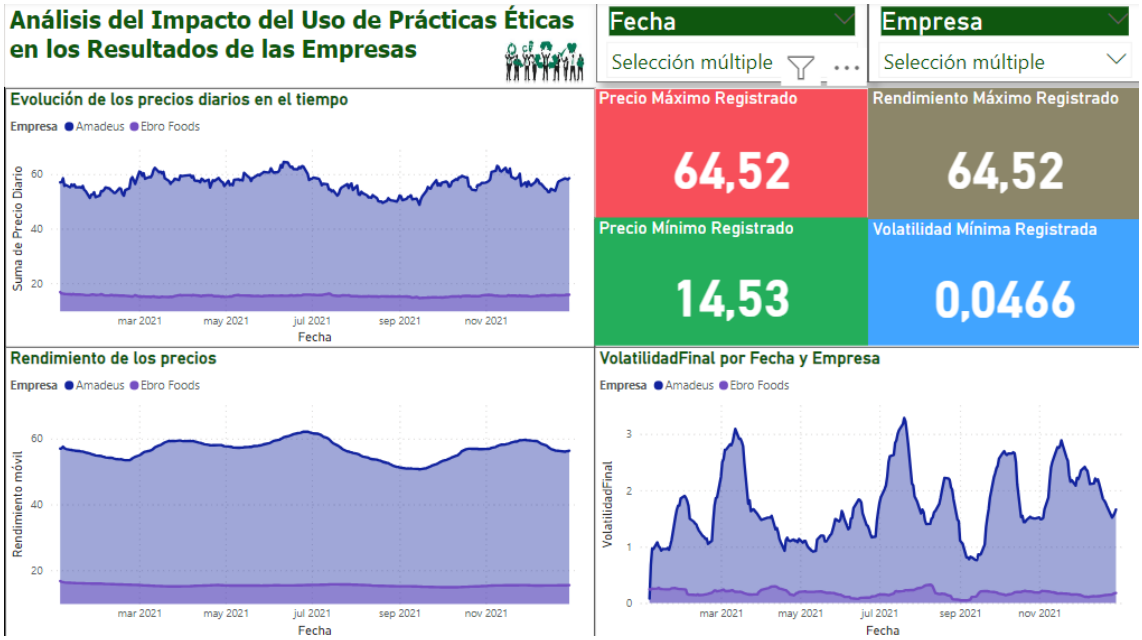


Ilustración 43: Resultado de la comparación de Amadeus y EbroFoods

Se puede deducir que Amadeus presenta un mayor precio diario (debido que su presencia en el Ibex35, que son las 35 empresas que cotizan en el mercado español con mayor liquidez) y por ello también presenta rendimientos mayores, pero presenta una volatilidad mucho mayor, por lo que es menos segura o más arriesgada su inversión que la inversión en EbroFoods.

De la realización general del proyecto se concluye destacando la utilización de la programación en la nube en comparación con la programación tradicional.

La programación en la nube proporciona mejor escalabilidad y flexibilidad. La escalabilidad permite la adaptación dinámica a las necesidades del proyecto y garantiza un rendimiento óptimo en todo momento. La flexibilidad permite ajustar los recursos de cómputo y almacenamiento según los requerimientos específicos, optimizando la eficiencia operativa y económica.

Esto mismo también conlleva una optimización significativa de los recursos y costes asociados. La eliminación de la necesidad de mantener infraestructuras físicas reduce los costes de mantenimiento y actualización, lo que resulta

especialmente beneficioso para proyectos con necesidades de soporte y mantenimiento continuos.

Finalmente, gracias a la cantidad de servicios ofrecidos por AWS Academy se facilita la innovación permitiendo al usuario el uso de tecnologías emergentes que aportan nuevas funcionalidades y formas de trabajo. Cabe destacar que AWS Academy proporciona una interfaz intuitiva que hace que sea fácil la comprensión de su uso y funcionamiento y además, ofrece una gran cantidad de documentación.

Académicamente la realización del trabajo ha servido para adquirir conocimientos en el ámbito de la programación en la nube, de la comprensión del funcionamiento del mercado y del descubrimiento de la existencia de las finanzas éticas y las empresas socialmente responsables, y lo más importante, una introducción al mundo del análisis de datos.

Personalmente el trabajo ha supuesto la resolución del reto planteado, utilizar una nueva herramienta (AWS), nuevos servicios (Lambda, DynamoDB, S3) y una mayor profundización en el uso de herramientas de visualización de datos (PowerBi). Todo ello ha hecho de este proyecto un trabajo emocionante y lleno de aprendizajes.

En un futuro se podrían seguir añadiendo funcionalidades a la aplicación para una mayor potenciación de su eficacia y su atractivo para los inversores. En primer lugar, se podrían integrar modelos predictivos que analicen datos históricos. De esta forma los usuarios podrían obtener proyecciones futuras de precios y rendimientos, facilitando decisiones de inversión más informadas y proactivas.

Por otro lado, se podrían enriquecer las opciones de personalización y filtrado, lo que permitiría a los usuarios profundizar en los datos según sus intereses específicos. Por ejemplo, añadir filtros por sector, criterios ESG detallados o periodos de tiempo más específicos. Esto haría que la herramienta fuera más versátil y ajustada a necesidades individuales del usuario final.

La implementación de alertas en tiempo real y notificaciones sobre cambios importantes en las métricas de las empresas como alertas sobre precios objetivo alcanzados o anomalías en la volatilidad.

Finalmente, se podría mejorar la interactividad de los informes y personalizar la experiencia del usuario permitiendo a los usuarios manipular gráficos, explorar detalles al pasar el cursor y recibir recomendaciones personalizadas basadas en su historial de uso. Esto aumentaría la utilidad de la aplicación y haría de ella una herramienta indispensable de apoyo para el inversor.

4. Análisis de Impacto

En este capítulo, se explora el impacto potencial de los resultados obtenidos a través del desarrollo del proyecto, para ello se va a visualizar el imparto desde los siguientes puntos de vista:

- ***Personal:*** a través de la realización de este trabajo he conocido tanto las finanzas éticas como las empresas socialmente responsables, además gracias a ello he aprendido como realizar un pequeño análisis de datos y el a la utilización de herramientas avanzadas como la nube.
- ***Empresarial:*** este cuadro de mando podría impulsar a algunas empresas a evaluar y ajustar sus estrategias de responsabilidad social para mejorar sus resultados.
- ***Social:*** socialmente, el cuadro de mando puede motivar a las empresas a adoptar comportamientos más éticos y responsables, lo cual beneficia a la sociedad en su conjunto al promover el desarrollo sostenible y la equidad.
- ***Económico:*** el cuadro de mando facilita la identificación de la correlación de las prácticas éticas con el desempeño económico, ofreciendo así un incentivo económico para la responsabilidad social. Además, el uso de herramientas de computación en la nube plantea la disminución de recursos y costes.
- ***Medioambiental:*** al promover las prácticas éticas, se promueve la reducción de la huella de carbono, que es crucial para la mitigación de los efectos del cambio climático. Además, la utilización de herramientas como Amazon Web Services aunque consume gran cantidad de energía puede ayudar a reducir la huella de carbono ya que, Al centralizar los recursos en centros de datos altamente eficientes y escalables, las empresas pueden evitar la necesidad de construir y mantener su propia infraestructura de TI. Además, AWS trabaja en el uso de energías renovables.
- ***Cultural:*** la herramienta refuerza la importancia de la responsabilidad social en el ámbito empresarial, promoviendo valores de sostenibilidad y ética en la comunidad empresarial y más allá.

Este proyecto apoya directamente a los objetivos 8 (trabajo decente y crecimiento económico), 12 (producción y consumo responsables) y 13 (acción por el clima), ya que el objetivo es que se invierta más en las empresas que adoptan prácticas éticas y de esta manera aumenten las empresas que adoptan estas medidas.

Ilustración 44: Objetivos de desarrollo sostenible. Fuente: https://www.un.org/sustainabledevelopment/es/objetivos-de-desarrollo-sostenible/

5. Bibliografía

[1] «Amazon Web Service Academy,» [En línea]. Available: https://aws.amazon.com/es/training/awsacademy/. [Último acceso: 2024].

[2] «PowerBi,» Microsoft, [En línea]. Available: https://powerbi.microsoft.com/es-es/landing/free-account/?ef_id=_k_Cj0KCQjw-_mvBhDwARIsAA-Q0Q6WoSL37ASabUg_x2nP7rshNz3AfJE0SJ0IfZGWvQ-qr8Y_wiK8fTEaAvyHEALw_wcB_k_&OCID=AIDcmm2x16xx83_SEM__k_Cj0KCQjw-_mvBhDwARIsAA-Q0Q6WoSL37ASabUg_x2nP7rshNz3AfJE0SJ0IfZGWv.

[3] «Amazon EC2,» [En línea]. Available: https://aws.amazon.com/es/ec2/?nc2=h_ql_prod_fs_ec2. [Último acceso: 2024].

[4] «Amazon S3,» 2024. [En línea]. Available: https://aws.amazon.com/es/s3/?nc2=h_ql_prod_fs_s3.

[5] «Amazon RDS,» [En línea]. Available: https://aws.amazon.com/es/rds/?nc2=h_ql_prod_fs_rds. [Último acceso: 2024].

[6] «Amazon DynamoDB,» [En línea]. Available: https://aws.amazon.com/es/dynamodb/?nc2=h_ql_prod_fs_ddb. [Último acceso: 2024].

[7] «Amazon Lambda,» [En línea]. Available: https://aws.amazon.com/es/lambda/?nc2=h_ql_prod_fs_lbd. [Último acceso: 2024].

[8] «AWS Lambda Layer,» Amazon Web Services, 2024. [En línea]. Available: https://docs.aws.amazon.com/lambda/latest/dg/chapter-layers.html. [Último acceso: 8 Mayo 2024].

[9] «Amazon Route 53,» [En línea]. Available: https://aws.amazon.com/es/route53/. [Último acceso: 2024].

[10] «Amazon Cloudfront,» [En línea]. Available: https://aws.amazon.com/es/cloudfront/. [Último acceso: 2024].

[11] «Amazon sns,» [En línea]. Available: https://aws.amazon.com/es/sns/. [Último acceso: 2024].

[12] «Amazon SQS,» [En línea]. Available: com. [Último acceso: 2024].

[13] «Amazon ECS,» [En línea]. Available: https://aws.amazon.com/es/ecs/. [Último acceso: 2024].

[14] «Amazon Cloudwatch,» [En línea]. Available: https://aws.amazon.com/es/cloudwatch/. [Último acceso: 2024].

[15] «Amazon Glue,» [En línea]. Available: https://aws.amazon.com/es/glue/. [Último acceso: 2024].

[16] «Características de PowerBi,» [En línea]. Available: https://www.arimetrics.com/glosario-digital/power-bi. [Último acceso: 2024].

[17] «Dashboards para usuarios de PowerBi,» Microsoft, 2024. [En línea]. Available: https://learn.microsoft.com/en-us/power-bi/consumer/end-user-dashboards. [Último acceso: 9 Mayo 2024].

[18] «Qué es Python,» Amazon Web Services, 2024. [En línea]. Available: https://aws.amazon.com/es/what-is/python/. [Último acceso: 9 Mayo 2024].

[19] «Yahoo! Finance,» Wikipedia, [En línea]. Available: https://es.wikipedia.org/wiki/Yahoo!_Finance.

[20] «Libreria yfinance,» [En línea]. Available: https://pypi.org/project/yfinance/.

[21] «AWS Lambda,» Manual AWS, [En línea]. Available: https://docs.aws.amazon.com/es_es/lambda/latest/dg/welcome.html.

[22] «Amazon Event Bridge,» Amazon Web Services, 2024. [En línea]. Available: https://aws.amazon.com/es/eventbridge/. [Último acceso: 4 Abril 2024].

[23] «DynamoDB,» Manual AWS, [En línea]. Available: https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/Introduction.html.

[24] «Amazon S3,» Amazon Web Services, [En línea]. Available: https://aws.amazon.com/es/pm/serv-s3/?gclid=Cj0KCQjw2PSvBhDjARIsAKc2cgPoHzzNBTBI9P6025B3_3rV9Kc3z8fRvr7AkoOVH5j6-mSrULbMB_waAplAEALw_wcB&trk=d0993ae4-4193-4d67-b2a9-e83cdb563369&sc_channel=ps&ef_id=Cj0KCQjw2PSvBhDjARIsAKc2cgPoHzzNBTBI9P6025B3_3rV9Kc3z8fRv.

[25] «JSON, qué es,» HubSpot, [En línea]. Available: https://blog.hubspot.es/website/que-es-json. [Último acceso: 8 Mayo 2024].

[26] «Batch Writer DynamoDB,» Amazon Web Services, 2024. [En línea]. Available: https://docs.aws.amazon.com/amazondynamodb/latest/APIReference/API_BatchWriteItem.html. [Último acceso: 10 Marzo 2024].

[27] «Lambda Handler,» Amazon Web Services, 2024. [En línea]. Available: https://docs.aws.amazon.com/lambda/latest/dg/python-handler.html. [Último acceso: 10 Marzo 2024].

[28] «Setem Andalucía,» 2024. [En línea]. Available: https://www.setem.org/andalucia/campanas/finanzas-eticas/#:~:text=Las%20finanzas%20%C3%A9ticas%20y%20solidarias,de%20objetivos%20sociales%20y%20ambientales.. [Último acceso: 18 Febrero 2024].

[29] «BBVA,» 2024. [En línea]. Available: https://www.bbva.com/es/salud-financiera/finanzas-eticas-ahorrar-e-invertir-en-clave-sostenible/. [Último acceso: 17 Febrero 2024].

[30] «¿Qué es el mercado diario?,» Audin for systems, 7 Mayo 2024. [En línea]. Available: https://www.audinforsystem.es/noticias-del-sector/mercado-diario-e-intradiario/#:~:text=El%20mercado%20diario%20es%20el,las%2024h%20del%20d%C3%ADa%20siguiente..

[31] «The investor,» 2024. [En línea]. Available: https://theinvestoru.com/blog/rendimiento-de-una-accion/#:~:text=El%20rendimiento%20de%20una%20acci%C3%B3n%20es%20una%20medida%20utilizada%20por,se%20invirti%C3%B3%20en%20el%20activo.. [Último acceso: 26 Marzo 2024].

[32] «Que es el rendimiento: Definición, fórmula y cálculo,» LiteFinance, 2021. [En línea]. Available: https://www.litefinance.org/es/blog/for-beginners/que-es-el-rendimiento/. [Último acceso: 1 Mayo 2024].

[33] «Tableros de control,» 2024. [En línea]. Available: https://misovirtual.virtual.uniandes.edu.co/codelabs/etl-Tableros/index.html?index=..%2F..ETL#0. [Último acceso: 2024].

[34] «Power Query,» Microsoft, 2024. [En línea]. Available: https://learn.microsoft.com/es-es/power-query/power-query-ui. [Último acceso: 7 Mayo 2024].

[35] «Referencia de funciones DAX,» Microsoft, 2024. [En línea]. Available: https://learn.microsoft.com/es-es/dax/dax-function-reference. [Último acceso: 7 Mayo 2024].

[36] «Los colores de la sostenibilidad,» Reaponsablia, 2024. [En línea]. Available: https://www.responsablia.com/los-colores-de-la-sostenibilidad-i-la-gestion-tricolor/. [Último acceso: 10 Mayo 2024].

6. Anexos

6.1 Anexo 1: Script Python función lambda yfygit:

```
import yfinance as yf
from boto3.dynamodb.conditions import Key
import boto3
import datetime
import uuid


# Inicializa el cliente de DynamoDB
dynamodb = boto3.resource('dynamodb')
# Metodo para obtener precios diatios
def obtener_precio_diario(symbol, start_date, end_date):
    try:
        data = yf.download(symbol, start=start_date, end=end_date)
        precios_diarios = data['Adj Close']
        return precios_diarios
    except Exception as e:
        print(f"Error al obtener datos de {symbol}: {e}")
        return None
# Generacion de un id unico para cada fila de la tabla
def generar_id_unico():
    return str(uuid.uuid4())
# Conexion con la tabla DynamoDB
def obtener_tabla_prices():
    return dynamodb.Table('yfinance_extract')
# Carga de datos en la tabla
def guardar_en_dynamodb(empresa, precios_diarios):
    tabla = obtener_tabla_prices()
    with tabla.batch_writer() as batch:
        for i, precio in enumerate(precios_diarios):
            unique_id = generar_id_unico()
            item = {
                'id': unique_id,
                'Empresa': empresa,
                'Fecha': str(precios_diarios.index[i].date()),
                'Precio Diario': str(precio),
            }
            batch.put_item(Item=item)
# Lambda handler
def lambda_handler(event, context):
    empresas = {
        "Applus": "APPS.MC",
        "Atresmedia": "A3M.MC",
        "CIE Automotive": "CIE.MC",
        "Ebro Foods": "EBRO.MC",
        "Ence": "ENC.MC",
        "Gestamp": "GEST.MC",
```

```
        "Amadeus": "AMS.MC",
        "Acelormit": "ACX.MC",
        "Endesa": "ELE.MC",
        "Merlin": "MRL.MC",
        "Naturgy": "NTGY.MC",
        "Rovi": "ROVI.MC",
        "Solaria": "SLR.MC",
        "Unicaja": "UNI.MC"
    }
    end_date = datetime.datetime.now().date()
    start_date = end_date - datetime.timedelta(days=5)

    for empresa, symbol in empresas.items():
        precios_diarios = obtener_precio_diario(symbol, start_date,
end_date)
        if precios_diarios is not None:
            guardar_en_dynamodb(empresa, precios_diarios)
        else:
            print(f"No se pudieron obtener los datos de {empresa}.")
```

6.2 Anexo 2: Script Pyhon función lambda yfinance_DDB_S3

```
import json
import boto3

def lambda_handler(event, context):
    dynamodb = boto3.resource('dynamodb')
    s3 = boto3.client('s3')

    table = dynamodb.Table('yfinance_extract')
    response = table.scan()
    data = response['Items']

    while 'LastEvaluatedKey' in response:
        response =
table.scan(ExclusiveStartKey=response['LastEvaluatedKey'])
        data.extend(response['Items'])

    # Convertir los datos a JSON
    json_data = json.dumps(data)

    # Especifica el nombre del archivo y el bucket de S3
    bucket_name = 'yfinanceextract'
    file_name = 'yfinance_extract_data.json'
    s3.put_object(Bucket=bucket_name, Key=file_name, Body=json_data)

    return {
        'statusCode': 200,
        'body': json.dumps(f'Datos exportados correctamente a
{bucket_name}/{file_name}')
    }
```

6.3 Anexo 3: Script Python función S3topowerbi

```
import json
import boto3

def lambda_handler(event, context):
    s3 = boto3.client('s3')
    bucket_name = 'yfinanceextract'

    # Listar objetos en el bucket ordenados por la fecha de ultima
modificación.
    objects = s3.list_objects_v2(Bucket=bucket_name)

    # Comprobar si hay objetos en el bucket.
    if 'Contents' in objects and objects['Contents']:
        # Ordenar objetos por la fecha de ultima modificación
(descendente).
        sorted_objects = sorted(objects['Contents'], key=lambda x:
x['LastModified'], reverse=True)

        # Obtener el nombre del archivo mas reciente.
        file_name = sorted_objects[0]['Key']

        # Obtener el objeto mas reciente.
        obj = s3.get_object(Bucket=bucket_name, Key=file_name)

        # Leer el contenido del objeto y cargarlo como JSON.
        data = json.loads(obj['Body'].read())

        # Retornar los datos obtenidos y el nombre del archivo.
        return {
            'statusCode': 200,
            'headers': {
                'Content-Type': 'application/json'
            },
            'body': json.dumps(data),
            'file_name': file_name
        }
    else:
        # Manejar el caso en que el bucket esta vacio.
        return {
            'statusCode': 404,
            'body': 'No se ha encontrado ningún archivo JSON en el
bucket.'
        }
```

6.4 Anexo 4: informe Turniting

turnitin

Recibo digital

Este recibo confirma quesu 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:	ALEJANDRA CASTILLO ALVAREZ
Título del ejercicio:	Turnitin Memoria Final (Moodle PP)
Título de la entrega:	Memoria final Alejandra Castillo.pdf
Nombre del archivo:	10328_ALEJANDRA_CASTILLO_ALVAREZ_Memoria_final_Alejan...
Tamaño del archivo:	3.01M
Total páginas:	50
Total de palabras:	11,698
Total de caracteres:	65,311
Fecha de entrega:	03-jun.-2024 09:45a. m. (UTC+0200)
Identificador de la entre...	2394423643


Universidad Politécnica de Madrid

Escuela Técnica Superior de Ingenieros Informáticos

Grado en Ingeniería Informática

Trabajo Fin de Grado

Cuadro de Mando para el Análisis del Impacto del Uso de Prácticas Éticas en los Resultados de las Empresas

Autor: Alejandra Castillo Álvarez

Tutor(a): Fernando Pérez Castejón

Madrid, Junio 2024

Este documento esta firmado por

	**Firmante**	CN=tfgm.fi.upm.es, OU=CCFI, O=ETS Ingenieros Informaticos - UPM, C=ES
	Fecha/Hora	Mon Jun 03 11:34:30 CEST 2024
	Emisor del Certificado	EMAILADDRESS=camanager@etsiinf.upm.es, CN=CA ETS Ingenieros Informaticos, O=ETS Ingenieros Informaticos - UPM, C=ES
	Numero de Serie	561
	Metodo	urn:adobe.com:Adobe.PPKLite:adbe.pkcs7.sha1 (Adobe Signature)