



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

– **TELECOM** ESCUELA
TÉCNICA **VLC** SUPERIOR
DE INGENIERÍA DE
TELECOMUNICACIÓN

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería de
Telecomunicación

Visualización interactiva de datos financieros abiertos en el
sector TIC mediante Python

Trabajo Fin de Grado

Grado en Ingeniería de Tecnologías y Servicios de
Telecomunicación

AUTOR/A: Jiménez Pérez, Blanca

Tutor/a: Guijarro Coloma, Luis Alejandro

CURSO ACADÉMICO: 2023/2024

Resumen

En este trabajo se ha llevado a cabo el desarrollo de una aplicación en Python para el análisis de datos financieros de empresas del sector de las tecnologías de la información y las comunicaciones. Utilizando repositorios de datos abiertos, la herramienta descarga y visualiza automáticamente información clave, como series temporales y comparativas de indicadores económicos y financieros. La aplicación busca la semiautomatización del proceso, permitiendo a los usuarios explorar gráficamente y comparar datos de manera eficiente. Este enfoque se alinea con los temas abordados en la asignatura de Política de Telecomunicaciones del Grado GITST. Además, presenta una utilidad añadida al poder ser empleado en la asignatura para presentar de forma mucho más visual y dinámica las distintas variables a evaluar, a través del tiempo, así como realizar una comparación entre las empresas estudiadas.

Palabras clave

Datos abiertos, Visualización interactiva, Datos financieros, Python

Abstract

In this work, a Python application for the financial data analysis of companies in the information and communication technology sector is developed. Using open data repositories, the tool automatically downloads and displays vital information, such as time series and comparative economic and financial indicators. The application seeks the semi-automation of the process, allowing users to graphically explore and compare data efficiently. This approach aligns with the topics covered in the Telecommunications Policy course of the GITST degree. Additionally, it presents an added utility by being employable in the course to show the various variables to be evaluated more visually and dynamically over time, as well as to perform a comparison between the studied companies.

Keywords

Open Data, Interactive Visualization, Financial Data, Python



Índice

Capítulo 1.	Introducción	1
1.1	Presentación	1
1.2	Motivación y agradecimientos	1
Capítulo 2.	Objetivos	2
Capítulo 3.	Contexto	4
3.1	Asignatura Política de las Telecomunicaciones	4
3.2	Datos abiertos en el sector financiero.....	5
3.3	Empresas y datos a comparar del sector TIC	8
3.3.1	Empresas del sector TIC	8
3.3.2	Datos a comparar.....	11
Capítulo 4.	Metodología / Herramientas	13
4.1	Python y uso de las bibliotecas	13
4.1.1	Python	13
4.1.2	Dash.....	15
4.1.3	Pandas	19
4.1.4	Requests	20
4.2	Explicación obtención de datos SEC y APIs.....	22
4.2.1	SEC	22
4.2.2	API	24
4.2.3	API para acceder a la SEC	26
Capítulo 5.	Análisis y diseño	28
5.1	Especificaciones de usuario	28
5.2	Diseño del código.....	30
5.2.1	Configuración inicial y definición de las tasas de conversión	31
5.2.2	Obtención y procesamiento de datos financieros	31



5.2.3	Inicialización dash y configuración de urls para datos financieros	32
5.2.4	Procesamiento de datos y preparación de la interfaz de usuario	32
5.2.5	Diseño y estructura de la interfaz de usuario	33
5.2.6	Actualización dinámica de gráficos basada en selecciones del usuario	34
5.2.7	Ejecución de la aplicación.....	35
Capítulo 6.	Referencias	36
Capítulo 7.	Anexo	43
7.1	Manual de usuario	43
7.2	Manual de administrador.....	49
7.3	Código	52



Índice de tablas

Tabla 1. Diferencia entre API y aplicación web	26
--	----

Índice de ilustraciones

Ilustración 1. Empresas TIC a comparar	8
Ilustración 2. Diagrama de acción del código	30
Ilustración 3. Ejecutar código en visualizador	43
Ilustración 4. Abrir link para abrir aplicación Dash	44
Ilustración 5. Aplicación Dash inicial	44
Ilustración 6. Desplegable 'Seleccione las empresas'	45
Ilustración 7. Desplegable 'Seleccione la fecha'	46
Ilustración 8. Desplegable 'Selecciona el tipo de dato'	46
Ilustración 9. Desplegable 'Seleccione el gráfico'	47
Ilustración 10. Ejemplo final gráfico líneas	47
Ilustración 11. Ejemplo final gráfico barras	48
Ilustración 12. Mensaje updating	48

Acrónimos empleados:

ADE Administración y Dirección de Empresas

API Application Programming Interface – Interfaz de Programación de Aplicaciones

CIK Clave de Índice Central

CPU Unidad Central de Procesamiento

CSS Cascading Style Sheets

GITST Grado Ingeniería de Tecnologías y Servicios de Telecomunicación

GSM Global System for Mobile Communication

HTML Hyper Text Markup Language

HTTP Hypertext Transfer Protocol

IA Inteligencia Artificial

IDE Entorno de Desarrollo Integrado

IFRS International Financial Reporting Standards

JSON JavaScript Object Notation

RMO Operadores Red Móvil

MVNO Operadores de Red Móvil Virtual

PVM Máquina Virtual de Python

REST Transferencia de Estado Representacional

RU Requisitos del Usuario

SEC Securities and Exchange Commission – Comisión de Bolsa y Valores de EEUU

SEK Corona Sueca

SMTP Simple Mail Transfer Protocol

SOAP Protocolo de Acceso a Objetos Simples



TFG Trabajo de Fin de Grado

TIC Tecnologías de la Información y Comunicación

URL Uniform Resource Locator

URS Requisitos del Usuario

USD Dólar estadounidense

US-GAAP Generally Accepted Accounting Principles

XBRL Extensible Business Reporting Language

XML Extensible Markup Language



Capítulo 1. Introducción

1.1 Presentación

Mi nombre es Blanca Jiménez Pérez y soy estudiante del doble Grado de Administración y Dirección de Empresas e Ingeniería de Tecnologías y Servicios de Telecomunicación, en la Universidad Politécnica de Valencia. Como culminación de mis estudios de Telecomunicaciones, este Trabajo de Fin de Grado se ha enfocado en el desarrollo de una aplicación Dash, con la finalidad de proporcionar una herramienta analítica avanzada diseñada para enriquecer la asignatura de Política de Telecomunicaciones.

1.2 Motivación y agradecimientos

Mi motivación principal para llevar a cabo este Trabajo de Fin de Grado ha sido la de ir más allá de la mera formalidad académica, buscando dejar una huella significativa tanto en mi trayectoria personal como en la comunidad educativa a la que pertenezco. Consciente de la importancia de aportar valor real a través de mis esfuerzos, me propuse desarrollar una herramienta que no solo sirviera como demostración de las competencias adquiridas durante mi formación en Ingeniería de Tecnologías y Servicios de Telecomunicación, sino que también ofreciera un recurso práctico y enriquecedor para los futuros estudiantes que cursen la asignatura de Política de las Telecomunicaciones.

Finalmente, me gustaría agradecer a todos mis profesores el haberme apoyado y motivado durante toda la carrera; y más en concreto, a mi profesor de Política de las Telecomunicaciones y tutor de este TFG por toda la ayuda brindada y el apoyo prestado. Gracias Luis.



Capítulo 2. Objetivos

El presente Trabajo Final de Grado se propone alcanzar una serie de objetivos específicos, concebidos para abordar tanto las necesidades teóricas como prácticas inherentes al campo de la Ingeniería de Telecomunicaciones. Este proyecto ha sido diseñado con la intención de aportar soluciones concretas y valor añadido al ámbito académico y profesional, mediante la implementación de tecnologías y metodologías avanzadas. Los objetivos delineados a continuación guían el desarrollo del trabajo, estableciendo un marco claro para la investigación y la aplicación práctica:

1. **Desarrollo de una herramienta analítica:** Crear una aplicación basada en Dash para el análisis exhaustivo de datos financieros relevantes para el sector de las tecnologías de la información y las comunicaciones. Esta herramienta se enfocará en mejorar el acceso y la interpretación de datos complejos a través de visualizaciones dinámicas.
2. **Integración de datos abiertos:** Utilizar fuentes de datos abiertos para alimentar la aplicación, asegurando que la información manejada sea de amplia disponibilidad, actualizada y fiable. Este enfoque promueve la transparencia y la replicabilidad del análisis.
3. **Fomento del aprendizaje interactivo:** Enriquecer el proceso educativo en la asignatura de Política de Telecomunicaciones, proporcionando a los estudiantes una plataforma que facilite el aprendizaje interactivo y la exploración autónoma de los datos.
4. **Contribución al conocimiento académico:** A través de la implementación y el análisis de la herramienta desarrollada, contribuir a la base de conocimiento existente en el ámbito de las telecomunicaciones, particularmente en lo que respecta a la aplicación de tecnologías de análisis de datos en la educación.
5. **Evaluación de usabilidad y eficacia:** Realizar una evaluación sistemática de la usabilidad y la eficacia de la aplicación desarrollada, con el fin de identificar áreas de



mejora y asegurar que la herramienta cumpla con las expectativas de los usuarios finales.

Cada uno de estos objetivos ha sido cuidadosamente seleccionado para asegurar que el TFG aporte valor significativo tanto a la comunidad académica como a la práctica profesional, alineándose con las tendencias actuales en tecnología y educación en ingeniería. La consecución de estos objetivos permitirá no solo validar la viabilidad técnica del proyecto, sino también su relevancia y aplicabilidad en contextos educativos y profesionales.



Capítulo 3. Contexto

3.1 Asignatura Política de las Telecomunicaciones

La asignatura Política de las Telecomunicaciones, es una asignatura que se puede cursar bien como optativa en cuarto año del GITST, o como asignatura obligatoria en el quinto año del Doble Grado GITST+ADE. Esta asignatura se basa en el análisis de la influencia que ejercen las autoridades reguladoras en el ámbito de las Tecnologías de la Información y Comunicación (TIC), con especial énfasis en las telecomunicaciones. La primera parte de la asignatura, en la cual se centra principalmente este trabajo, consta de una revisión comprensiva de la composición y las tendencias del sector, seguida por una inmersión en las directrices legales y las disposiciones reglamentarias que lo gobiernan, tanto a nivel nacional como global. [3]

La finalidad principal de esta asignatura es fortalecer la base técnica ya obtenida a través de otras materias, proporcionando una perspectiva amplia sobre la industria de las telecomunicaciones que engloba las perspectivas de negocios, financiera, legal y de control regulatorio. De esta forma, se presenta un panorama que incluye a operadores de red, fabricantes de equipo y proveedores de servicios de internet, delineando así el terreno en el que los ingenieros de telecomunicaciones encontrarán oportunidades profesionales.

Durante años anteriores, el profesor de la asignatura en cuestión, y tutor de este TFG, ha sido el encargado de revisar los datos de las distintas empresas consideradas relevantes, con el fin de crear diversas gráficas que facilitarán a los alumnos la visualización y comparación de los datos. Esta tarea repetitiva llevada a cabo cada año, es la que se va a tratar de evitar con la implementación del siguiente Trabajo de Fin de Grado. Con ello el análisis de las empresas seleccionadas se podrá realizar directamente sin necesidad de buscar todos los datos uno a uno cada año, lo que permitirá agilizar el proceso de creación de gráficas para la impartición de la asignatura, además de permitir a los estudiantes comparar de forma dinámica e interactiva los distintos valores, empresas y fechas que quieran conocer.

A continuación, se procede a explicar la importancia actual de los datos abiertos en el sector financiero; es decir, la existencia pública de toda la información financiera de las empresas.

3.2 Datos abiertos en el sector financiero

En la actualidad, los datos abiertos son uno de los activos más valiosos que existen, y aún en mayor medida en el sector de las finanzas. El concepto de datos abiertos financieros se basa en permitir el acceso universal a detalles financieros, tales como estadísticas de mercado, balances corporativos, índices de interés, y otros. Se busca que estos datos estén disponibles de forma que puedan ser accedidos, usados y compartidos libremente por todos, impulsando de esta manera la claridad y cooperación dentro del ámbito financiero. [48]

La transición hacia el Open Banking y el Open Finance, propiciada por los datos abiertos, está transformando el sector financiero hacia un ecosistema más competitivo y de calidad. La disponibilidad de información permite que, tanto las instituciones como los usuarios finales, puedan tomar decisiones más informadas, redundando en beneficios colectivos.

Tras diversos sucesos financieros ocurridos a lo largo de la historia, se pudo concluir que los organismos no estaban tecnológicamente lo suficiente preparados para la revolución tecnológica emergente, por ello, con el fin de mejorar la transparencia y la accesibilidad de los datos, se crearon dos nuevos marcos regulatorios. Por un lado, Basilea III, dirigido a las entidades bancarias (Directiva 2013/36/UE y Reglamento UE nº 575/2013); y por otro, Solvencia II, para las entidades aseguradoras (Directiva 2009/138/EC y Directiva 2014/51/UE). [17]

Las normativas mencionadas incorporan una sección crucial referente a la transparencia y la liberación de datos en el sector financiero, denominada Pilar 3. Este componente es esencial en los esquemas normativos, aludiendo a la divulgación de datos y al imperativo de claridad por parte de bancos y compañías de seguros.

Dicha claridad opera en dos dimensiones: la primera se trata de las instituciones hacia el mercado, abarcando tanto el aspecto comercial (clientes, accionistas, inversores) como el público en general; la segunda, de las instituciones hacia los entes reguladores, quienes a su vez deben adherirse a ciertos principios de divulgación de datos, sobre todo a nivel agregado por entidad y tipo de producto. En cumplimiento de estas normas, las autoridades pertinentes también difunden informes y estadísticas de acceso público, como



lo hace en España la Dirección General de Seguros y Pensiones, encargada de regular el ámbito asegurador. [17]

Además, ante la creciente amenaza de delitos cibernéticos, tales como el fraude y el robo de identidad, la adopción de prácticas de datos abiertos se ha convertido en una herramienta esencial. Estos datos, intercambiados a nivel individual y transaccional entre entidades financieras, facilitan la creación de sistemas de seguridad robustos, evitando que la información caiga en manos de ciberdelincuentes a través de bases de datos vulnerables, fortaleciendo de esta manera la seguridad cibernética. [47]

La compilación y difusión de esta información cuantitativa demanda un esquema de reporte que asegure la fiabilidad y uniformidad de los datos. Aunque la publicación de esta información por parte de las entidades financieras y los reguladores se encuentra en etapas iniciales de desarrollo tecnológico, es crucial establecer estándares comunes para la apertura y accesibilidad de los datos financieros lo antes posible, como se observa en otros sectores.

Para iniciar la estandarización de este esquema de reporte entre las entidades financieras, actualmente se utiliza el estándar XBRL (Extensible Business Reporting Language), que facilita la interoperabilidad y el análisis de la información financiera y de negocios conforme a las normativas europeas pertinentes.

De cara a perspectivas futuras, se anticipa que los datos abiertos desempeñarán un papel aún más crítico en el ámbito financiero. Además, la creciente accesibilidad a estos datos y su valor reconocido incentivarán a más empresas, tanto emergentes como tradicionales, a adoptar modelos basados en la transparencia y el libre flujo de información.

Por otro lado, el futuro de los datos abiertos en finanzas incluye el aprovechamiento de tecnologías como la inteligencia artificial (IA) y el aprendizaje automático (Machine Learning) para el análisis y predicción a partir de estos datos. Estas tecnologías pueden procesar enormes volúmenes de información de manera eficiente, ofreciendo insights para anticipar tendencias de mercado, detectar actividades fraudulentas y desarrollar soluciones financieras personalizadas. [33]

Por último destacar, que para que todos estos retos tecnológicos se puedan llevar a cabo, se deben incluir el desarrollo y mantenimiento de software y APIs capaces de



gestionar y transmitir datos de manera efectiva, a menudo en tiempo real. Esto abarca todos los aspectos de la ingeniería de datos, desde el alojamiento hasta el procesamiento de la información. [25]

3.3 Empresas y datos a comparar del sector TIC

Siguiendo con el hilo de lo comentado hasta ahora, y adelantando un poco la explicación del trabajo, se procede a explicar cuáles han sido las empresas seleccionadas para este estudio, en que grupos se pueden agrupar y el porqué de su elección. Del mismo modo, se exponen también los diferentes tipos de datos que se van a analizar, y de dónde ha sido extraída toda esta información.

3.3.1 Empresas del sector TIC

Las empresas seleccionadas para el estudio han sido divididas en tres grupos, dependiendo de la función que tengan dentro del mundo de las TIC, para hacer más fácil y lógica la posible comparación. En la siguiente imagen, se pueden ver cuáles han sido los grupos seleccionados a analizar y las empresas que van a conformar cada grupo:

OPERADORAS:	FABRICANTES:	PRESTADORES DE SERVICIOS:
<ul style="list-style-type: none">• Telefónica• Orange• Vodafone	<ul style="list-style-type: none">• Nokia• Ericsson	<ul style="list-style-type: none">• Alphabet• Microsoft

*Ilustración 1. Empresas TIC a comparar
Fuente: Recorte del Dash creado en este TFG*

Los operadores de telefonía móvil, también conocidos como proveedores inalámbricos, se clasifican principalmente en dos categorías: los Operadores de Red Móvil (MNO), los cuales poseen la infraestructura y el espectro necesarios para brindar el servicio, y los Operadores de Red Móvil Virtual (MVNO), que adquieren el servicio al por mayor de un MNO y lo revenden a sus clientes. Estos operadores proporcionan servicios GSM. [5]

Según la CNMC (Comisión Nacional de los Mercados y la Competencia), las principales operadoras tanto de telefonía móvil como fija, actualmente en nuestro país son: Telefónica, Orange, Vodafone, MásMóvil y Digi. [30] De las cinco indicadas, se han seleccionado las tres primeras para el análisis de este trabajo.

Telefónica: caracterizada por su amplia capitalización bursátil y extensa base de usuarios (más de 383 millones, abarcando Europa y Latinoamérica). Actuando como una entidad de carácter privado con participación en los mercados bursátiles de España, Nueva York y Lima, se dedica a la oferta de una diversidad de servicios de comunicación, tanto inalámbricos como alámbricos, enriquecidos con una gama de opciones digitales destinadas a satisfacer las necesidades de particulares y empresas. [24][55]

Orange: segundo mayor operador en el mercado de telecomunicaciones español, con 298 millones de clientes globales al cierre del año 2023. Es además, uno de los principales inversores extranjeros en España. A través de sus marcas Orange, Jazztel y Simyo, la empresa atiende a más de 21,6 millones de clientes en España, ofreciendo servicios de telefonía fija, móvil y televisión. [19][36]

Vodafone: atiende a más de 300 millones de clientes en 17 países, extendiendo su alcance mediante acuerdos de red en otros 46 países. Vodafone cuenta con un total de 17 millones de clientes en España para junio de 2023, sumando los clientes de servicios móviles, banda ancha, televisión, y el incremento reportado por Lowi. Además, se caracteriza por permitir que diversas compañías, incluyendo filiales directas de Vodafone y Operadores Móviles Virtuales (OMV) independientes, se beneficien de su red. Entre las empresas que utilizan la cobertura de Vodafone se encuentran: Lowi, Finetwork, Vodafone Yu, Lebara, Hits Mobile, Aloha, Econet. [56][58]

Por otro lado, se encuentran los fabricantes, los cuales abarcan tanto el hardware como los equipos, siendo esenciales para el envío y la recepción de información digital. Este campo ha experimentado una fusión con la tecnología de la información (TI) desde los años 90, impulsada por el auge de Internet y su papel en la transferencia de datos.

En 2015, los líderes globales en ventas de estos equipos, excluyendo dispositivos móviles, fueron: Nokia Networks (Finlandia), Ericsson (Suecia), Cisco Systems (EE.UU.), Juniper Networks (EE.UU.), Huawei Technologies, y ZTE Corporation, ambos de China. Nuevamente se han seleccionado los dos primeros, ya que son los principales, y estos se describen a continuación. [6]

Ericsson: multinacional sueca fundada en 1876, especializada en la fabricación de equipos y soluciones para telecomunicaciones, incluyendo telefonía fija y móvil, comunicaciones multimedia e Internet. En España, destaca por su significativa presencia

desde 1922, contando con más de 2.200 empleados y tres centros de I+D repartidos en Madrid, Málaga y Barcelona, lo que establece al país como un punto de referencia clave para la compañía a nivel europeo y global. [7][18]

Nokia: establecida en Finlandia en 1865 y conocida por ser un líder en el mercado de teléfonos móviles entre 1998 y 2011. La empresa ha evolucionado hacia las redes de telecomunicaciones y tecnologías avanzadas. Tras una alianza con Microsoft en 2011 y la venta de su división de dispositivos móviles a Microsoft en 2014, Nokia ha recalibrado su enfoque hacia las infraestructuras de red y la innovación tecnológica. [8][59]

Por último, los prestadores de servicios son entidades que ofrecen soluciones tecnológicas en áreas como ciberseguridad, almacenamiento de datos y telecomunicaciones. El ranking mundial de estas compañías, basado en la facturación de 2022, sitúa a Apple, Alphabet, Samsung Electronics, Microsoft y Huawei como los cinco principales actores globales. Estas empresas se distinguen por su liderazgo en ingresos y su capacidad para adaptarse y prosperar incluso frente a desafíos económicos y geopolíticos. [53][13][14]

En este caso, se ha optado por seleccionar a Alphabet y Microsoft como empresas representantes de este grupo.

Alphabet: matriz de Google, ofrece servicios destacados como Gmail, YouTube y un motor de búsqueda avanzado. Con su tecnología de mapeo, Google asiste a usuarios a nivel mundial, incluyendo la cartografía de terrenos oceánicos y terrestres. [2][61]

Microsoft: destaca por ofrecer soluciones de telecomunicaciones integradas con servicios en la nube, facilitando la transición de los operadores a redes 5G con herramientas como la virtualización de funciones de red. En España, proporciona infraestructuras de red avanzadas y servicios de computación en la nube, apoyados por desarrollos como Azure Edge Zones. [9][34]

3.3.2 Datos a comparar

Por otro lado, se procede a explicar que tipos de datos se han seleccionado para analizar con este trabajo, basándose nuevamente en la asignatura cursada en el GITST, Política de las Telecomunicaciones:

- **Activos:** constituyen recursos económicos que proporcionan beneficios futuros, como efectivo, propiedades y patentes. Se clasifican en activos corrientes y no corrientes.

[32]

- **Activos corrientes:** recursos financieros de una empresa que se espera convertir en efectivo dentro del ciclo operativo anual, incluyendo efectivo en bancos, inventarios y valores negociables.[28]
- **Activos no corrientes:** bienes de larga duración en la empresa, difíciles de liquidar rápidamente, que figuran en el balance general y no son idóneos para resolver necesidades de liquidez inmediata. [49]

- **Propiedad, planta y equipo:** representan los activos fijos tangibles que una empresa utiliza en sus operaciones de producción, alquiler o gestión, y se registran en los estados financieros para reflejar los recursos duraderos dedicados a las operaciones empresariales.[1]

- **Intangibles:** recursos no físicos de valor de una compañía, como la imagen de marca, el conocimiento exclusivo y las relaciones con clientes. Aportan ventajas competitivas y contribuyen al éxito a largo plazo, pero son difíciles de medir en términos monetarios.[52]

- **Fondo de comercio:** refleja el valor adicional de una empresa, basado en activos no físicos como la marca, clientes, posicionamiento, conocimientos especializados y patentes. Se determina por la diferencia entre el precio de compra de la empresa y su valor contable.[37]

- **Ingresos:** suma del dinero obtenido por vender productos o servicios, también conocidos como ingresos por ventas o ventas brutas. [44]

- **Deuda a corto plazo:** deber de reembolsar los fondos prestados por terceros en menos de un año. [12][20]



- **Deuda a largo plazo:** deber de reembolsar los fondos prestados por terceros en un periodo mayor a un año. [12][20]
- **Patrimonio neto:** activos y derechos totales de una empresa en un momento específico, obtenido al sumar el capital, reservas y ganancias del año en curso.[57]
- **Resultado neto:** refleja las ganancias residuales tras deducir los gastos de venta de productos y operativos de la empresa, ofreciendo una medida más precisa de la rentabilidad de una empresa que sus ingresos totales. [39]
- **Resultado operativo:** informe financiero que muestra la diferencia entre ingresos y gastos (excluyendo el IVA) de una empresa durante un período específico, permitiéndote evaluar rápidamente las ganancias o pérdidas. [40]
- **ROA:** Indicador que mide la eficiencia con la que la empresa utiliza sus activos para generar ganancias. Un ROA alto indica que la empresa es capaz de convertir sus inversiones en activos en ingresos de manera eficaz.[45]
- **ROE:** Indicador que evalúa la rentabilidad generada por la empresa a partir del capital de los accionistas. Un ROE alto señala que la empresa crea valor significativo para sus accionistas con el capital invertido.[45]
- **Ratio activos corrientes / activos totales:** indicador de la parte de activos correspondientes a activos corrientes medido en porcentaje.
- **Ratio activos no corrientes / activos totales:** indicador de la parte de activos correspondientes a activos no corrientes medido en porcentaje.

Un motivo adicional para seleccionar estos valores radica en su inclusión en los informes anuales y trimestrales de las empresas, que están disponibles públicamente a través de la Comisión de Bolsa y Valores (SEC, por sus siglas en inglés), permitiendo su recolección automática anual de las entidades elegidas.



Capítulo 4. Metodología / Herramientas

Para poder crear el código de este trabajo, se tomaron una serie de decisiones iniciales sobre cuáles eran las mejores herramientas a emplear. Es por esto, que en este capítulo se va a proceder a explicar qué lenguaje de programación y que bibliotecas fueron seleccionadas, tras la realización de un profundo estudio, para ver cuáles se adaptaban mejor a las necesidades de este trabajo.

4.1 Python y uso de las bibliotecas

4.1.1 Python

Python es un lenguaje de programación de nivel elevado que destaca principalmente por su enfoque para leer el código.

Por contextualizar un poco, fue diseñado por Guido van Rossum en 1991 con el fin de proporcionar una solución de programación accesible para aquellos con altos conocimientos en programación, tanto para los que no, ganando desde entonces una gran popularidad en diversos sectores tecnológicos: desarrolladores, analistas de datos, IA, Machine Learning, automatización de procesos y tareas, etc. [29]. Esto es posible gracias a la sintaxis simple y legible que emplea Python, utilizando un lenguaje muy semejante al inglés, facilitando la creación y comprensión del código a todo el mundo.

Además, al compararlo con otros lenguajes de programación como podría ser C++, Python cuenta con una gran ventaja, y es que convierte directamente en código máquina ejecutable por la CPU el código fuente, compilándolo a bytecode. Este proceso intermedio genera un código sencillo que se ejecuta dentro de una Máquina Virtual Python (PVM), permitiendo así su ejecución en múltiples plataformas sin necesidad de adaptación, lo que subraya su portabilidad.

Algunas de las características que más resaltan en Python frente a otros lenguajes son:



Tipado dinámico: Python es de tipado dinámico, lo que significa que el tipo de una variable se determina en tiempo de ejecución y no necesita ser declarado explícitamente, lo que agiliza el desarrollo.

Gestión automática de la memoria: El lenguaje maneja la memoria automáticamente a través de un recolector de basura, lo que elimina la necesidad de liberar memoria manualmente y reduce el riesgo de fugas de memoria y otros errores relacionados.

Extensa biblioteca estándar: Python viene con una gran biblioteca estándar que ofrece módulos y funciones para tareas de programación comunes, lo que reduce la necesidad de escribir o instalar código adicional para muchas tareas.

Interoperabilidad: Python puede integrarse con otros lenguajes de programación y tecnologías, incluyendo C, C++, y Java, permitiendo el uso de bibliotecas y sistemas legados dentro de proyectos Python.

Portabilidad: Python es altamente portátil; es decir, el código Python puede ejecutarse en múltiples plataformas sin necesidad de modificarlo, incluyendo sistemas operativos populares como Windows, Linux, y macOS.

Amplia comunidad de soporte: La amplia y activa comunidad de desarrolladores de Python contribuye con una vasta cantidad de bibliotecas de terceros y frameworks, facilitando el desarrollo de aplicaciones en campos especializados y proporcionando soporte y documentación extensiva.

Frameworks y herramientas para desarrollo en áreas específicas: Python ofrece frameworks y herramientas robustas para el desarrollo web (como Django y Flask), análisis de datos y ciencia de datos (pandas, NumPy, SciPy, Matplotlib), y aprendizaje automático (scikit-learn, TensorFlow, PyTorch), entre otros.

Verificación de errores: en Python se realiza principalmente en tiempo de ejecución, con pocas comprobaciones en el momento de la compilación. Esto puede resultar en errores detectados solo durante la ejecución específica de partes del código, pero facilitando la interpretación de estos. [27]

Es por esto por lo que se ha seleccionado este lenguaje para la creación del código que concierne este trabajo, además de ser uno de los tipos de lenguaje estudiados a lo largo de la carrera.

Concluyendo, son todas estas características las que han convertido a Python en uno de los lenguajes más utilizados en el mundo de la programación, y principalmente en el área de la visualización de datos, donde gracias a su amplia colección de bibliotecas especializadas se pueden crear interfaces sencillas, con un gran atractivo para los desarrolladores. Es el caso de la biblioteca Dash, conocida por la creación de interfaces web analíticas interactivas, sin requerir experiencia previa en desarrollo web. [41]

Se procede a explicar con mayor profundidad esta biblioteca, así como otras empleadas también en el código creado.

4.1.2 Dash

Dash es una herramienta versátil de Python que se ha establecido como una solución innovadora para el desarrollo de aplicaciones web y visualización de datos. Distinguiéndose por su facilidad de uso y eficacia, ya que facilita la creación de interfaces de usuario ricas e interactivas, adaptándose a las necesidades cambiantes de los analistas de datos y desarrolladores.

Dash se sostiene sobre pilares tecnológicos robustos, destacando por su integración de: [26]

- Flask: Proporciona la infraestructura básica de las aplicaciones web, incluyendo el manejo del servidor y las rutas.
- Plotly: Central para la generación de visualizaciones dinámicas, Plotly enriquece Dash con la capacidad de crear gráficos interactivos de alta calidad.
- ReactJS: Este framework de JavaScript es esencial para la creación de componentes de interfaz de usuario eficientes y reactivos en Dash.

Dash sobresale en el panorama de las herramientas de desarrollo por sus atributos distintivos, especialmente por su adaptabilidad en una amplia gama de entornos digitales. Esta flexibilidad se manifiesta en su capacidad para operar de manera eficiente en cualquier navegador web, ofreciendo una experiencia de usuario interactiva y profundamente personalizada que se ajusta a las necesidades específicas de cada proyecto. Su integración con tecnologías de vanguardia, como Bootstrap, permite una

adaptación fluida del diseño a una multitud de dispositivos y tamaños de pantalla, desde ordenadores de escritorio hasta dispositivos móviles, asegurando una interfaz de usuario coherente y accesible en todos los contextos.

En el ámbito del desarrollo, Dash se distingue por su agilidad y flexibilidad, capacitando a los desarrolladores para avanzar desde la creación de prototipos rápidos hasta la implementación de sistemas complejos y escalables con una eficiencia notable. Esta versatilidad hace de Dash una herramienta indispensable para proyectos de cualquier magnitud, permitiendo una iteración rápida y la experimentación sin comprometer la robustez o la funcionalidad del sistema final.

La filosofía de código abierto de Dash es un pilar central de su innovación, ofreciendo un marco colaborativo que fomenta la participación comunitaria y el desarrollo conjunto. La comunidad activa de Dash no solo contribuye a un ecosistema en constante enriquecimiento sino que también asegura una evolución continua de la plataforma, adaptándose a las últimas tendencias tecnológicas y a las demandas emergentes del mercado. Este enfoque colaborativo garantiza que Dash permanezca en la vanguardia del desarrollo de aplicaciones, proporcionando una solución robusta, escalable y de futuro seguro para la visualización de datos y la construcción de interfaces de usuario complejas en el ámbito digital.

Un ejemplo de la capacidad de Dash es la creación de dashboards para el seguimiento de tendencias en redes sociales o para el análisis en tiempo real del comportamiento de usuarios en plataformas digitales. Estos casos ilustran cómo Dash puede ser una herramienta clave en el ámbito del marketing digital y la investigación de mercados.

En el entorno de desarrollo de Dash, la interactividad y el diseño se entrelazan a través de una arquitectura robusta y modular, diseñada para facilitar la creación de aplicaciones web interactivas y ricas en datos. La plataforma se distingue por su capacidad para integrar componentes interactivos avanzados, tales como menús desplegados, controles deslizantes, y botones, que son fundamentales para la interacción del usuario. Estos elementos permiten una experiencia de usuario altamente personalizable y dinámica, mejorando significativamente la usabilidad y el engagement.

La visualización de datos es otro pilar central de Dash, con Plotly sirviendo como el motor detrás de su poderosa capacidad de representación gráfica. Los gráficos de Plotly,

integrados dentro de Dash, ofrecen una amplia gama de tipos de visualización, desde diagramas de barras y líneas hasta gráficos complejos de dispersión y mapas de calor. Esta diversidad permite a los usuarios presentar datos complejos de forma intuitiva, facilitando el análisis y la comprensión de patrones, tendencias y correlaciones. [15]

Los callbacks en Dash representan una innovación clave en la interactividad de las aplicaciones web. Mediante la definición de funciones de callback, los desarrolladores pueden establecer relaciones dinámicas entre los inputs de los usuarios y los outputs de la visualización de datos, permitiendo que las actualizaciones de los componentes se realicen sin la necesidad de recargar la página. Esto se logra a través de una sintaxis Pythonica que refleja la filosofía de diseño de Dash de ser altamente legible y accesible para los desarrolladores de diversos niveles de experiencia.

Diferenciándose del desarrollo web tradicional basado en HTML, CSS y JavaScript, Dash introduce una metodología única para la definición de la estructura y el estilo de las aplicaciones. Los estilos se definen mediante diccionarios de Python, y la propiedad `className` se utiliza en lugar de la tradicional `class` de HTML, lo que facilita la integración con el ecosistema de Python y promueve un flujo de trabajo más coherente para los desarrolladores acostumbrados a este lenguaje.

La capacidad de Dash para transformar datos complejos en narrativas visuales es una de sus características más potentes. Utilizando `dash_core_components`, los desarrolladores pueden crear visualizaciones detalladas y narrativas interactivas que hacen que los datos sean accesibles y comprensibles para una audiencia amplia. Este enfoque no solo mejora la capacidad de los usuarios para analizar y extraer conclusiones de los datos, sino que también promueve una mayor interacción y exploración.

Otra de sus características, es que se ejecuta sobre un servidor web, similar a cómo Flask opera en el ecosistema de Python, y su modo de depuración avanzado permite a los desarrolladores realizar iteraciones rápidas, ajustando y mejorando las aplicaciones en tiempo real. Esta característica es crucial para un desarrollo eficiente y ágil, permitiendo una respuesta rápida a los cambios y la optimización continua de la experiencia del usuario.

Cabe explicar que construir una aplicación con Dash implica un proceso simplificado pero detallado que comienza con la configuración inicial, donde se lleva a cabo la

instalación de Dash junto con bibliotecas complementarias esenciales como Pandas. Esta etapa establece la base sólida sobre la cual se desarrollará la aplicación, asegurando que todas las herramientas necesarias estén disponibles desde el inicio. [26]

A continuación, la importación estratégica de módulos desempeña un papel crucial en el proceso de desarrollo. La selección de los módulos apropiados de Dash, junto con elementos de Flask para el backend y Plotly para las visualizaciones, es fundamental para garantizar una funcionalidad óptima de la aplicación. Esta etapa requiere una comprensión clara de los componentes necesarios para cumplir con los objetivos específicos del proyecto y cómo estos módulos interactúan entre sí para crear una experiencia de usuario fluida y efectiva.

La preparación de datos para la visualización es otro paso esencial en el proceso de construcción de la aplicación. Utilizando Pandas, los datos se estructuran y manipulan para alinearse con los objetivos de la aplicación, permitiendo una integración fluida con las capacidades de visualización de Dash. Esta etapa es crucial para garantizar que los datos no solo sean precisos y relevantes, sino también presentados de una manera que sea intuitiva y accesible para los usuarios finales.

Finalmente, el diseño y despliegue de la interfaz de usuario marcan la culminación del proceso de desarrollo. En esta fase, se configura la aplicación y su interfaz para satisfacer las necesidades específicas del proyecto, desde la estética hasta la funcionalidad. La interfaz de usuario debe ser intuitiva y atractiva, asegurando que los usuarios puedan navegar e interactuar con la aplicación de manera eficiente. El despliegue adecuado de la interfaz es vital para el éxito de la aplicación, ya que determina cómo los usuarios finales experimentarán y utilizarán la herramienta desarrollada. [4]

En resumen, Dash emerge como una solución integral para la visualización de datos y el desarrollo web, ofreciendo una plataforma que combina interactividad, adaptabilidad y una curva de aprendizaje accesible, adecuada tanto para proyectos sencillos como para aplicaciones analíticas avanzadas.

4.1.3 Pandas

Pandas, una biblioteca de software libre diseñada para el análisis y la manipulación de datos en Python, destaca por su capacidad, flexibilidad y facilidad de uso. Facilita operaciones como la carga, alineación, manipulación y fusión de datos con un rendimiento sobresaliente, gracias a su implementación en C y Python. Originario del término "Panel Data", que refiere a conjuntos de datos que contemplan observaciones a través del tiempo, Pandas se desarrolló para ser una solución de análisis de datos de alto nivel en Python.

Su ambición es convertirse en la herramienta de código abierto dominante para la manipulación y análisis de datos, superando las capacidades ofrecidas por cualquier otro lenguaje de programación. Pandas también es ampliamente utilizado en el ámbito de la preparación de datos, o "Data Wrangling", que abarca técnicas para transformar datos no estructurados en formatos listos para el análisis.

Especialmente eficaz en el manejo de datos estructurados en formatos tabulares, matrices y series temporales, Pandas se integra armoniosamente con otras bibliotecas de Python. Su núcleo funcional, el "DataFrame", es una estructura de datos bidimensional que organiza los datos en columnas (variables) y filas (observaciones). Los DataFrames facilitan el almacenamiento de información tanto numérica como textual, siendo una herramienta esencial para los científicos de datos y programadores acostumbrados al lenguaje R para estadísticas, especialmente en aplicaciones de aprendizaje automático (Machine Learning). [16]

Pandas soporta la importación y exportación de datos en múltiples formatos como CSV, JSON, y permite realizar limpieza de datos (Data Cleaning). Es invaluable para trabajar con datos estadísticos, tabulares (como tablas SQL o hojas de cálculo Excel), series temporales y matrices con etiquetas en filas y columnas. Entre sus capacidades se incluyen [60]:

- La lectura y escritura de datos en una amplia variedad de formatos: CSV, JSON, Excel, bases de datos SQL, Parquet y HDF5.
- La selección y filtración de datos de manera intuitiva, basándose en posición, valor o etiquetas.
- La fusión y unión de conjuntos de datos.



- La transformación de datos mediante la aplicación de funciones de manera global o por ventanas.
- La manipulación avanzada de series temporales.
- La generación de gráficos para la visualización de datos.

Esta reformulación presenta una descripción técnica y unificada de Pandas, resaltando su propósito, funcionalidades y aplicaciones en el contexto del análisis de datos y Machine Learning.

4.1.4 Requests

La biblioteca `requests` en Python se distingue como una interfaz robusta y accesible para ejecutar operaciones de solicitud y respuesta HTTP. Esta herramienta provee una amplia gama de funcionalidades, incluyendo el manejo de encabezados personalizados, autenticación y cookies, brindando así la flexibilidad y el dominio necesarios para efectuar transferencias de datos por Internet de manera efectiva. [31]

Implementar una solicitud HTTP utilizando `requests` requiere comprensión de los parámetros esenciales para la comunicación con el servidor y la forma de transmitir la información pertinente. El dominio de estas capacidades, junto con la práctica adecuada, permite realizar peticiones HTTP de forma eficiente, facilitando la recuperación de datos de manera rápida y fiable.

Al emplear esta biblioteca, se interactúa primordialmente con dos entidades: se inicia creando un objeto `requests` que encapsula todos los detalles necesarios para la petición, la cual se ejecuta contra un servidor. Posteriormente, se recibe un objeto `Response` del servidor, el cual incluye la información retornada por este y, además, encapsula el objeto `requests` inicial, ofreciendo un acceso integral a los datos de la solicitud y la respuesta. [35]

`Requests` se ha consolidado como un estándar de facto para la realización de solicitudes HTTP en el desarrollo de aplicaciones web del lado del servidor en Python.



Esta biblioteca simplifica significativamente el proceso de desarrollo gracias a su API sencilla, optimizando la experiencia de programación.

Es crucial entender el protocolo HTTP en este contexto, el cual se define como un protocolo de solicitud y respuesta que opera bajo el modelo cliente-servidor. Este modelo se apoya en conexiones TCP/IP para el intercambio de mensajes de solicitud y respuesta, estableciendo los fundamentos para la comunicación en la web. [23]

4.2 Explicación obtención de datos SEC y APIs

4.2.1 SEC

La Comisión de Bolsa y Valores de los Estados Unidos (SEC, por sus siglas en inglés, Securities and Exchange Commission) es la entidad federal encargada de la regulación y supervisión de los mercados financieros del país, asegurando la protección de los inversores y la integridad de los mercados. Creada en 1934 tras el colapso del mercado bursátil de 1929, su función esencial es la aplicación de la ley contra la manipulación del mercado, y se rige por diversas legislaciones que incluyen la Ley del Mercado de Valores de 1934 y la Ley de Valores de 1933. [10]

La SEC promueve la divulgación completa y pública de la información financiera por parte de las empresas cotizadas y supervisa a participantes clave en los mercados de valores como corredores, asesores de inversiones, fondos de inversión y bancos. La autoridad de la Comisión se extiende a imponer multas y llevar a cabo procedimientos judiciales contra aquellos que incumplen las normativas.

Sus objetivos principales son proteger a los inversores, asegurar la transparencia y la equidad de los mercados, y combatir el fraude, como el uso de información privilegiada o la contabilidad fraudulenta. Además, la SEC coordina con reguladores internacionales para enfrentar desafíos transfronterizos, dada la globalización de los mercados financieros.

La estructura organizativa de la SEC incluye distintas divisiones especializadas en áreas como finanzas corporativas, trading y mercados, gestión de inversiones, y cumplimiento de la ley. Además, supervisa las bolsas de valores y otros entes, y gestiona la base de datos EDGAR, que facilita el acceso público a informes financieros y otros documentos relevantes. [11]

La SEC también enfrenta desafíos continuos, incluida la crítica por la falta de aplicación efectiva de las leyes y por no prevenir adecuadamente los fraudes financieros. A lo largo de los años, se han propuesto reformas para mejorar su eficacia y transparencia, tales como aumentar su financiamiento, mejorar la protección de los denunciantes y fortalecer la supervisión de las tecnologías financieras emergentes.



La SEC juega un papel crucial en el fomento de un ambiente de confianza para los inversores, algo vital para el funcionamiento eficiente de los mercados de capital. Al requerir la transparencia y la divulgación de información financiera, la SEC permite a los inversores tomar decisiones informadas, promoviendo así la formación de capital y el crecimiento económico. La supervisión de la SEC sobre las prácticas de mercado busca prevenir los abusos y proteger a los inversores de prácticas desleales y fraudulentas.

Además, la SEC tiene un enfoque proactivo en la educación de los inversores, proporcionando recursos y herramientas para ayudarles a comprender el mercado y tomar decisiones financieras prudentes. Este esfuerzo educativo complementa su papel regulador y de supervisión, subrayando la importancia de una participación informada en los mercados financieros.

La adaptación a las nuevas tecnologías también ha sido un enfoque clave para la SEC. La emergencia de criptomonedas, plataformas de financiamiento colectivo (crowdfunding), y otras innovaciones tecnológicas han requerido que la SEC evalúe y, en algunos casos, ajuste su enfoque regulatorio para garantizar la protección del inversor mientras facilita la innovación y el desarrollo de nuevos mercados.

La colaboración internacional es otro aspecto crucial de la labor de la SEC, dada la naturaleza global de los mercados financieros. La Comisión trabaja en estrecha colaboración con reguladores de todo el mundo para promover estándares regulatorios altos y cooperar en la supervisión y ejecución de las leyes de valores. Esta cooperación es fundamental para abordar los desafíos de la evasión regulatoria y el fraude que trascienden las fronteras nacionales. [46]

En resumen, la SEC es una piedra angular en la arquitectura del sistema financiero de Estados Unidos, esencial para mantener la confianza en los mercados de valores. A través de su trabajo de regulación, supervisión, y educación, la SEC busca asegurar que los mercados de valores de Estados Unidos sean seguros, justos y eficientes para todos los participantes, contribuyendo así a la salud y estabilidad de la economía estadounidense. Con el avance de la tecnología y la evolución de los mercados financieros, la SEC continúa adaptándose y evolucionando para enfrentar los retos del siglo XXI, manteniendo su compromiso de proteger a los inversores y asegurar la integridad de los mercados.

4.2.2 API

La Interfaz de Programación de Aplicaciones (API por sus siglas en inglés, Application Programming Interface) constituye un mecanismo esencial que facilita la interacción y el intercambio de datos entre distintas plataformas de software. Tienen su origen en las etapas iniciales de la informática, precediendo significativamente la aparición de la computadora personal. Durante aquellos tiempos, su implementación típica se hallaba en las bibliotecas de los sistemas operativos, donde su funcionamiento se limitaba mayormente al ámbito local de los sistemas específicos en los que se ejecutaban. No obstante, ocasionalmente facilitaban el intercambio de mensajes entre los sistemas centrales. Tras un periodo cercano a tres décadas, las APIs trascendieron las limitaciones de los entornos locales, evolucionando hacia una herramienta crucial para la integración de datos a distancia. Para el inicio de los años 2000, ya se habían consolidado como una tecnología fundamental en el ámbito de la integración remota de datos. [42]

Las APIs funcionan como un mediador que posibilita que una aplicación solicite servicios o información de otra, utilizando un conjunto predefinido de operaciones y protocolos. Este enfoque permite a los desarrolladores integrar y extender funcionalidades sin la necesidad de crear soluciones desde cero, mejorando así la eficiencia y la interoperabilidad entre sistemas.

Las APIs se caracterizan por su capacidad para abstraer la complejidad, permitiendo la interacción entre las capas de software de alto y bajo nivel mediante un conjunto de interfaces bien definidas. Este nivel de abstracción ofrece a los desarrolladores la ventaja de acceder a funcionalidades complejas de manera simplificada, evitando la necesidad de implementar detalles de bajo nivel.

En cuanto a las especificaciones técnicas, cabe comentar la posible clasificación dependiendo de su visibilidad y el tipo de acceso que ofrecen, incluyendo APIs públicas (o abiertas), privadas (o internas), de aliados comerciales (partners), y compuestas. [43]

Las APIs privadas se caracterizan por su uso exclusivo interno dentro de una organización. Este enfoque permite un control riguroso, asegurando que solo el personal autorizado tenga acceso a ellas. La ventaja principal radica en la capacidad de las empresas para mantener un alto grado de control y seguridad sobre sus recursos digitales.

Las APIs de partners se diseñan para ser compartidas con socios comerciales selectos. Esta modalidad ofrece la posibilidad de generar ingresos adicionales y extender la funcionalidad de los servicios sin comprometer la integridad y calidad. Al abrir estas APIs a un círculo confiable de socios, las empresas pueden fomentar la innovación colaborativa y expandir su alcance en el mercado.

Las APIs públicas, accesibles por cualquier entidad externa, promueven la máxima apertura e innovación. Al permitir que desarrolladores externos y otras empresas interactúen con estas, se facilita la creación de un ecosistema de aplicaciones enriquecido que puede impulsar la adopción y el reconocimiento de la marca. Las APIs públicas pueden ser catalizadores de transformación en diversos sectores, alentando el surgimiento de nuevos modelos de negocio y servicios innovadores.

Por otro lado, existen dos principales en el desarrollo de APIs: SOAP (Protocolo de Acceso a Objetos Simples) y REST (Transferencia de Estado Representacional). SOAP, que utiliza XML para el formato de mensajes y puede operar sobre HTTP o SMTP, es conocido por su rigidez y compatibilidad entre diferentes sistemas y lenguajes de programación. REST, por otro lado, adopta un enfoque más flexible y menos prescriptivo, funcionando como un estilo de arquitectura más que como un protocolo estricto. Las APIs RESTful, que siguen las limitaciones arquitectónicas delineadas por Roy Fielding, ofrecen una interfaz uniforme y un sistema sin estado que facilita la escalabilidad y la integración entre plataformas web.

Este contraste entre SOAP y REST subraya la evolución hacia prácticas de desarrollo más ágiles y orientadas a la web, con un enfoque creciente en la simplicidad, la eficiencia y la interoperabilidad. La elección entre SOAP y REST depende de las necesidades específicas de la aplicación, el entorno de operación y los objetivos de negocio, resaltando la importancia de una estrategia bien definida para la gestión de APIs.[54]

Para sintetizar, una API representa un compendio de normativas y protocolos diseñados para habilitar la comunicación y el intercambio de información entre diversas aplicaciones de software. En contraste, una aplicación web se define como un software accesible mediante un navegador web, el cual ofrece una interfaz de usuario diseñada para la interacción de los usuarios finales. Comúnmente, las aplicaciones web incorporan APIs para facilitar su conexión con servicios y sistemas externos.

A continuación se muestran las principales características que distinguen una API de una aplicación web:

	API	APLICACIÓN WEB
PROPÓSITO	Comunicar diferentes aplicaciones de software entre sí	Permitir a los usuarios interactuar con las interfaces de usuario
INTERFAZ DE USUARIO	No tienen. Son usadas por los desarrolladores	Tiene para permitir a los usuarios interactuar directamente, permitiendo realizar tareas
INTERACCIÓN	Interacción programática	Interacción humana
INTERCAMBIO DE DATOS	Función principal. Proporcionan forma para que los desarrolladores recuperen o envíen datos entre sistemas	Usan APIs y otros servicios backend para recuperar y manipular los datos, gestionando la experiencia de usuario
USUARIOS FINALES vs DESARROLLADORES	Usadas por desarrolladores en sus aplicaciones. Usuarios finales no interactúan directamente	Diseñadas para los usuarios finales
EJEMPLOS	API de redes sociales (Twitter), API de pasarelas de pago (Paypal), API servicios en la nube (AWS),...	Aplicaciones web como Facebook, Gmail, Google Docs,...

Tabla 1. Diferencia entre API y aplicación web

Fuente: Creación propia a partir de [54]

4.2.3 API para acceder a la SEC

La API de la Comisión de Bolsa y Valores (SEC) brinda un acceso directo y rápido para revisar las últimas presentaciones ante la SEC y acceder a la información de documentos corporativos almacenados en el archivo EDGAR desde el año 1994. Esta API habilita la visualización en tiempo real de dichas presentaciones, simplifica el proceso de búsqueda y análisis de informes financieros auditados y no auditados, y permite la extracción de texto de los documentos de EDGAR, así como su conversión a formatos como PDF, Word o Excel. [51]

En el portal "data.sec.gov", se encuentran disponibles APIs REST que suministran datos en formato JSON a usuarios y sitios web en SEC.gov sin necesidad de autenticación o claves de acceso. Entre los datos ofrecidos por estas APIs se incluyen los historiales de presentaciones de los emisores y la información XBRL de los informes financieros



(incluyendo los formularios 10-Q, 10-K, 8-K, 20-F, 40-F, 6-K y sus variantes). Estas estructuras de datos JSON se actualizan constantemente a lo largo del día, conforme se realizan nuevas presentaciones. Adicionalmente, se proporciona un archivo ZIP para la descarga masiva de estas estructuras JSON, el cual se actualiza y vuelve a publicar diariamente cerca de las 3:00 a.m. ET.

Para acceder al historial de presentaciones más reciente de una entidad específica, se puede visitar la URL:

<https://data.sec.gov/submissions/CIK#####.json>

Aquí, "#####" corresponde a la Clave de Índice Central (CIK) de diez dígitos de la entidad. La estructura JSON abarca metadatos importantes como el nombre actual e histórico de la entidad, así como las bolsas de valores y los símbolos de cotización de las compañías que cotizan en bolsa. En caso de que una entidad cuente con presentaciones adicionales, los archivos incluirán un conjunto de archivos JSON adicionales junto con el intervalo de fechas que abarca cada uno.

Desde 2009, la SEC requiere el uso de datos XBRL, un formato basado en XML para la presentación de informes financieros. Las APIs de XBRL consolidan información de las presentaciones utilizando taxonomías estandarizadas (tales como us-gaap, ifrs-full) relevantes para la entidad emisora, garantizando así la coherencia y la capacidad de comparación entre las distintas compañías y a través del tiempo.

La actualización de las APIs se realiza en tiempo real, con retrasos de procesamiento habituales inferiores a un segundo para las presentaciones y menores a un minuto para los datos XBRL, aunque estos tiempos pueden incrementarse durante picos de actividad de presentación.[50]

Capítulo 5. Análisis y diseño

5.1 Especificaciones de usuario

Los Requisitos del Usuario (RU), también conocidos como URS por sus siglas en inglés, representan las expectativas y necesidades que una entidad espera que un sistema satisfaga, ya sea para alcanzar objetivos de negocio o para cumplir con regulaciones específicas. Constituyen el núcleo de la justificación para el desarrollo o implementación del sistema, delineando el "qué" y el "por qué" detrás de la necesidad del sistema.

Existen dos categorías principales de RU, los obligatorios y los deseables. Los requisitos obligatorios son aquellos esenciales que el sistema debe cumplir para garantizar la calidad del producto, la seguridad del paciente, la conformidad con las normativas y la integridad de los datos. La falta de cumplimiento de estos requisitos representa riesgos inaceptables para la entidad. Los requisitos deseables corresponden a mejoras que, aunque no críticas, pueden beneficiar al sistema en términos estéticos, de optimización o de rendimiento, y poseen un riesgo mínimo en caso de no implementarse. Estos se consideran para una inclusión futura y, de no cumplirse, no impactan negativamente a la entidad. [38]

Dentro del marco del desarrollo de esta aplicación Dash enfocada en el análisis de datos financieros, se identifican diversas expectativas y necesidades fundamentales por parte de los usuarios finales. Estas expectativas delimitan las funcionalidades esenciales y los objetivos que la aplicación debe alcanzar para ser considerada exitosa y útil para su público objetivo. Se debe tener en cuenta, que en esta ocasión se establecen dos usuarios finales: el profesor de la asignatura de Política de las Telecomunicaciones y los alumnos de esta (comentado en los objetivos de este trabajo). A continuación, se detallan los requisitos obligatorios para esta aplicación:

- Acceso a Datos Financieros: Los usuarios requieren la capacidad de consultar información financiera esencial, como balances, ingresos, y datos sobre activos y pasivos de distintas empresas. Esta información debe ser precisa y recoger tanto datos históricos como actualizaciones recientes, provenientes de fuentes verificables como la SEC.



- Visualización Interactiva: La aplicación debe permitir a los usuarios visualizar los datos financieros en gráficos dinámicos e interactivos, facilitando el análisis comparativo entre diferentes empresas y periodos de tiempo.

- Filtrado y Selección de Datos: Los usuarios deben ser capaces de filtrar y seleccionar datos basándose en empresas específicas, rangos de fechas, y tipos de datos financieros.

- Actualización Dinámica: La interfaz debe actualizar los gráficos y visualizaciones de manera dinámica en respuesta a las selecciones y filtros aplicados por el usuario, sin necesidad de recargar la página.

- Seguridad y Privacidad: Los datos de usuario y las interacciones con la aplicación deben ser seguros y protegidos contra accesos no autorizados.

A continuación, se detallan los requisitos deseables considerados para esta aplicación:

- Análisis Avanzado de Datos Financieros: Se espera que la aplicación proporcione acceso a análisis financieros más complejos, incluyendo el cálculo de ratios financieros, análisis de tendencias y comparativas entre diferentes empresas.

- Facilidad de Uso: La aplicación debe ser intuitiva y fácil de usar, con una curva de aprendizaje mínima para usuarios con diversos niveles de experiencia en análisis financiero.

- Rendimiento y Velocidad: La aplicación debe responder rápidamente a las interacciones del usuario, incluso cuando se manejan grandes volúmenes de datos.

- Selección de Formatos de Visualización: La aplicación debe ofrecer opciones variadas de representación gráfica para los datos financieros, permitiendo a los usuarios escoger el formato más adecuado para una interpretación eficaz y personalizada de la información.

5.2 Diseño del código

Este documento proporciona un análisis técnico detallado del código fuente empleado para construir una aplicación Dash, diseñada para la visualización y análisis de datos financieros. Se abordará la arquitectura del código en segmentos claramente definidos, cada uno con un propósito específico dentro del ecosistema de la aplicación. La exposición se inicia desde la importación de librerías esenciales, pasando por la definición de estructuras de datos para el manejo de tasas de conversión de monedas, hasta la implementación de funciones para la obtención y procesamiento de datos a través de APIs externas. Posteriormente, se detalla la inicialización de la aplicación Dash, configuración de componentes de la interfaz de usuario, y la lógica detrás de la actualización dinámica de gráficos basada en interacciones del usuario. Este análisis busca ofrecer una comprensión integral de las técnicas de programación aplicadas y los principios de diseño que subyacen a la creación de aplicaciones web analíticas con Dash.

La siguiente figura muestra un diagrama de acción de cómo funcionaría el código, con los posibles errores que se crearían.

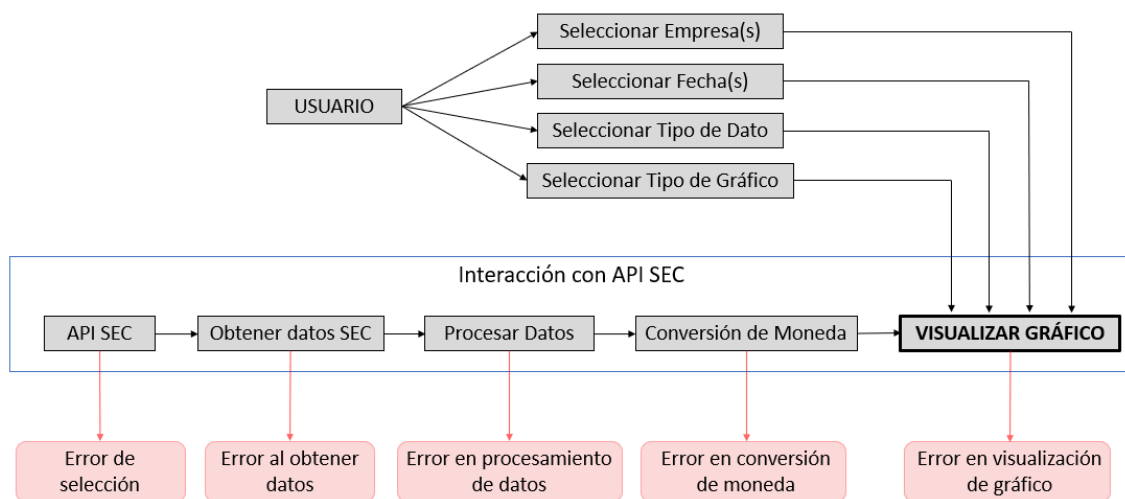


Ilustración 2. Diagrama de acción del código

Fuente: Creación propia basada en el código creado en este trabajo

5.2.1 Configuración inicial y definición de las tasas de conversión

El primer bloque del código se encarga de dos aspectos fundamentales de la aplicación. En primer lugar, se realiza la importación de librerías necesarias para el funcionamiento de la aplicación (ya explicadas en apartados anteriores), incluyendo Dash, pandas, y requests. Esto establece el entorno de trabajo y las herramientas que se utilizarán para desarrollar la funcionalidad de la aplicación.

Y en segundo lugar, se define un diccionario `CONVERSION_RATES` que contiene las tasas de conversión históricas de las diferentes monedas de las empresas seleccionadas a euros. Las monedas incluidas son el dólar estadounidense (USD) y la corona sueca (SEK), con tasas de conversión anuales desde 2009 hasta 2023. Esta estructura de datos es esencial para realizar conversiones de moneda dentro de la aplicación, permitiendo calcular valores en euros a partir de cantidades en USD o SEK basándose en el año especificado. [21][22]

5.2.2 Obtención y procesamiento de datos financieros

En la segunda parte, se define la función `get_sec_data` destinada a recuperar datos de la Comisión de Bolsa y Valores de Estados Unidos (SEC) utilizando una API. La función toma como parámetros una URL y una lista de URLs para procesar. Inicia configurando encabezados específicos en la solicitud HTTP GET para evitar ser bloqueado por el servidor de la SEC. Luego, realiza la solicitud a la URL dada y, si es exitosa, procesa la respuesta JSON.

La función busca específicamente la clave 'units' dentro de los datos JSON, la cual debe ser un diccionario. Para cada unidad de datos encontrada, normaliza los datos JSON a un DataFrame de pandas, convierte la columna "filed" a formato datetime, y ordena los datos por fecha. Todos los DataFrames generados se combinan en uno solo. Además, extrae el nombre de la entidad y un tag de la URL para incluirlos en el resultado.

Finalmente, devuelve un diccionario con los datos procesados, el nombre de la entidad, las unidades (claves de 'units') y el tag. Si se encuentra un error en la solicitud

HTTP o la clave 'units' no está presente o no es un diccionario, la función gestiona estos casos mostrando mensajes de error adecuados y retorna `None`.

5.2.3 Inicialización dash y configuración de urls para datos financieros

La siguiente sección del código se centra en la inicialización de la aplicación web utilizando la biblioteca Dash y la configuración de múltiples URLs para la obtención de diversos datos financieros de empresas a través de la API de la SEC. La aplicación Dash se inicializa con `dash.Dash(__name__)`, estableciendo el punto de partida para la creación de componentes web interactivos.

A continuación, se definen varias listas de URLs, cada una destinada a recuperar diferentes tipos de datos financieros de la SEC de las distintas empresas seleccionadas (estos datos y empresas se encuentran explicados en apartados anteriores), según las normas contables US-GAAP o IFRS.

5.2.4 Procesamiento de datos y preparación de la interfaz de usuario

A continuación se procesan y organizan los datos financieros obtenidos para todas las empresas indicadas, y a configurar elementos de la interfaz de usuario para la visualización de estos datos en la aplicación Dash. Inicialmente, se utiliza una comprensión de lista junto con la función `get_sec_data` para recuperar y almacenar los datos financieros de activos de cada empresa listada en `urls_empresas_assets`.

Posteriormente, se procede a obtener todas las fechas disponibles de los datos financieros, extrayendo los años de la columna "filed" de cada conjunto de datos y ordenándolos de manera descendente. Estas fechas se utilizan para definir las opciones del selector de fechas en la interfaz de usuario, permitiendo a los usuarios filtrar la información financiera por año específico, ver todos los años, o elegir no mostrar datos.

Además, se mapean los nombres de las empresas a versiones más legibles y uniformes mediante un diccionario ``nombre_empresas``, facilitando la identificación de las empresas por parte de los usuarios. Finalmente, se actualizan las opciones del desplegable para seleccionar empresas en la interfaz de usuario, asignando a cada empresa un valor único y su nombre legible correspondiente, lo que mejora significativamente la experiencia de usuario al interactuar con la aplicación para explorar los datos financieros de las empresas seleccionadas.

5.2.5 Diseño y estructura de la interfaz de usuario

En esta sección del código se establece la estructura principal de la aplicación Dash y se define la interfaz de usuario con varios componentes para la interacción. La estructura se organiza en contenedores ``Div`` que agrupan elementos de la interfaz, como menús desplegables, etiquetas y un gráfico para visualizar los datos.

1. Selección de Empresas y Fechas: Se crean menús desplegables con identificadores únicos (``empresa-selector``, ``fecha-selector``) para que los usuarios elijan múltiples empresas y fechas. Las opciones de estos menús se definen a partir de los datos procesados previamente.

2. Selección del Tipo de Dato y Gráfico: Se ofrece un menú desplegable para seleccionar el tipo de dato financiero a visualizar (activos, ingresos, patrimonio neto, etc.) y otro para elegir el tipo de gráfico (línea o barra). Estas opciones permiten personalizar la visualización de los datos según las preferencias del usuario.

3. Categorización de Empresas: Se incluyen secciones de texto que categorizan a las empresas en "Operadoras", "Fabricantes", y "Prestadores de Servicios", proporcionando un contexto claro sobre el sector al que pertenece cada empresa listada.

4. Información Personal: Un ``Div`` situado en la esquina inferior derecha muestra el nombre de la persona que desarrolló la aplicación, "BLANCA JIMÉNEZ PÉREZ", otorgando crédito y un punto de contacto.

5. Visualización de Datos: Se reserva espacio para un componente `Graph` de Dash (`time-series-plot`), donde se mostrarán los datos financieros de las empresas seleccionadas en forma de gráfico, basado en las selecciones de usuario para empresa, fecha, tipo de dato, y tipo de gráfico.

Este diseño de la interfaz de usuario es intuitivo y flexible, permitiendo a los usuarios explorar y analizar datos financieros de manera interactiva, con la capacidad de filtrar por empresa, periodo de tiempo, y categoría de datos, y visualizar la información en formatos gráficos ajustables.

5.2.6 Actualización dinámica de gráficos basada en selecciones del usuario

Este segmento del código define una función de callback en Dash, `update_plot`, que se activa al cambiar los valores seleccionados en los menús desplegables de la interfaz de usuario. La función tiene como objetivo actualizar el gráfico (`time-series-plot`) dinámicamente basándose en las selecciones de empresas, fechas, tipo de dato financiero, y tipo de gráfico realizadas por el usuario.

- Inputs del Callback: La función escucha cambios en cuatro inputs específicos correspondientes a los menús desplegables para selección de empresas, fechas, tipo de dato, y tipo de gráfico.

- Procesamiento de Datos: Para cada empresa seleccionada, la función obtiene y procesa datos financieros relevantes. Esto puede incluir la recuperación de datos de la SEC mediante `get_sec_data`, el cálculo de ratios financieros específicos como ROE o ROA, y la preparación de los datos para su visualización.

- Construcción de Trazos para el Gráfico: La función prepara trazos (series de datos) para el gráfico basándose en los datos procesados, aplicando filtros de fecha (para únicamente seleccionar el último valor de cada año y eliminar duplicados) y realizando cálculos específicos cuando es necesario.

- Configuración del Gráfico: Se configura el título del gráfico, los estilos de los ejes, y otros parámetros de visualización en función del tipo de dato seleccionado. Esto asegura que la visualización sea clara y coherente con los datos mostrados.

- Manejo de Errores: La función incluye manejo de errores para asegurar que, en caso de problemas al procesar los datos o actualizar el gráfico, se muestre una configuración básica del gráfico o se evite la interrupción de la aplicación.

La función ``update_plot`` es central para la interactividad de la aplicación Dash, permitiendo a los usuarios explorar datos financieros complejos de manera intuitiva y visual. Al ajustar las selecciones en la interfaz, los usuarios pueden analizar diferentes aspectos financieros de las empresas seleccionadas a lo largo del tiempo, facilitando el análisis comparativo y la identificación de tendencias o patrones importantes.

5.2.7 Ejecución de la aplicación

Esta última sección del código contiene el comando para ejecutar la aplicación Dash desarrollada. Este comando solo se ejecuta si el script actual es el punto de entrada principal, es decir, si el script se está ejecutando directamente y no siendo importado desde otro script.

- Comando de Ejecución: ``app.run_server(debug=True)`` inicia el servidor de Dash y activa el modo de depuración (``debug=True``). El modo de depuración facilita el desarrollo y la prueba de la aplicación al proporcionar recarga automática ante cambios en el código y mostrar información detallada de errores en el navegador.

Este comando es esencial para poner en marcha la aplicación y hacerla accesible a través de un navegador web, permitiendo a los usuarios interactuar con la interfaz gráfica, explorar los datos financieros mediante las selecciones disponibles, y visualizar los resultados en forma de gráficos dinámicos.

Capítulo 6. Referencias

- [1] Actualícese. *Características de la propiedad, planta y equipo*. Recuperado de <https://cdn.actualicese.com/Imagen-editoriales/Caracteristicas-propiedad-planta-equipo.png>
- [2] *Alphabet International España*. Recuperado de <https://www.alphabet.com/es-es>
- [3] *Asignaturas*.
https://intranet.upv.es/pls/soalu/sic_gdoc.get_content?P_ASI=12421&P_IDIOMA=c&P_VISTA=intranet&P_TIT=190&P_CACA=2023
- [4] Chapuis, D. (2023, 10 abril). *Crea una web app en Python con Dash*. Listopro Community.
<https://community.listopro.com/crea-una-web-app-en-python-con-dash/>
- [5] Colaboradores de Wikipedia. (2023b, octubre 19). *Operador de telefonía móvil*. Wikipedia, la Enciclopedia Libre.
https://es.wikipedia.org/wiki/Operador_de_telefon%C3%ADa_m%C3%B3vil
- [6] Colaboradores de Wikipedia. (2023a, abril 25). *Equipos de telecomunicaciones*. Wikipedia, la Enciclopedia Libre. https://es.wikipedia.org/wiki/Equipos_de_telecomunicaciones
- [7] Colaboradores de Wikipedia. (2024a, enero 16). *Ericsson*. Wikipedia, la Enciclopedia Libre.
<https://es.wikipedia.org/wiki/Ericsson#:~:text=Telefonaktiebolaget%20LM%20Ericsson%20com%C3%BAmente%20conocida,las%20comunicaciones%20multimedia%20e%20internet>
- [8] Colaboradores de Wikipedia. (2024b, febrero 10). *Nokia*. Wikipedia, la Enciclopedia Libre.
https://es.wikipedia.org/wiki/Nokia#Los_dispositivos_m%C3%B3viles_bajo_la_marca_Nokia_desde_2016
- [9] Colaboradores de Wikipedia. (2024c, febrero 25). *Microsoft*. Wikipedia, la Enciclopedia Libre. <https://es.wikipedia.org/wiki/Microsoft>
- [10] *Comisión de Bolsa y Valores (SEC) / USAGov*.
<https://www.usa.gov/es/agencias/comision-de-bolsa-y->

[valores#:~:text=La%20Comisi%C3%B3n%20de%20Bolsa%20y,mercados%20de%20valor es%20sean%20justos.](#)

- [11] Comisión de Bolsa y Valores de EE. UU. (2022). *Justificación del presupuesto del Congreso para el año fiscal 2023 y Plan de rendimiento anual*. Recuperado de https://www.sec.gov/files/FY%202023%20Congressional%20Budget%20Justification%20Annual%20Performance%20Plan_FINAL.pdf
- [12] Communications. (2023, 1 agosto). Qué es la deuda: para qué sirve y cómo se mide el endeudamiento financiero. *BBVA NOTICIAS*. <https://www.bbva.com/es/salud-financiera/que-es-la-deuda-para-que-sirve-y-como-se-mide-el-endeudamiento-financiero/>
- [13] Computing, R., & Computing, R. (2023, 30 septiembre). *¿Cuáles son las 50 compañías TIC que más facturan en España?* Computing. <https://www.computing.es/noticias/cuales-son-las-50-companias-tic-que-mas-facturan-en-espana/>
- [14] Computing. (2023, 30 septiembre). Ranking Líderes 2023: Las 30 compañías TIC que más facturan a nivel mundial. *Computing*. <https://www.computing.es/noticias/ranking-lideres-2023-las-30-companias-tic-que-mas-facturan-a-nivel-mundial/>
- [15] *DASH Documentation & User Guide | Plotly*. <https://dash.plotly.com/>
- [16] DataScientest. (2023, 30 octubre). *Pandas : La biblioteca de Python dedicada a la Data Science*. Formación En Ciencia de Datos | DataScientest.com. <https://datascientest.com/es/pandas-python>
- [17] Datos.gob.es. (2016, 27 diciembre). Introducción al ecosistema de datos abiertos en el sector financiero. *datos.gob.es*. <https://datos.gob.es/es/noticia/introduccion-al-ecosistema-de-datos-abiertos-en-el-sector-financiero>
- [18] Ericsson. *Ericsson en España*. Recuperado de <https://www.ericsson.com/es/about-us/company-facts/ericsson-worldwide/spain>
- [19] Espagne, O. *Orange España - Conócenos*. Orange. <https://www.orange.es/acercadeorange/orangeespana/conocenos#:~:text=Espa%C3%B1a%20es%2C%20en%20t%C3%A9rminos%20de,31%20de%20diciembre%20de%202023.>

- [20] Estrategias de Inversión. *Dividendos más rentables de la bolsa Española*.
<https://www.estrategiasdeinversion.com/herramientas/diccionario/analisis-fundamental/composicion-del-endeudamiento-t-571#:~:text=La%20deuda%20de%20la%20empresa,las%20que%20exceden%20el%20a%20C3%B1o.>
- [21] *EXR.A.SEK.EUR.SP00.A* / *ECB Data Portal*.
<https://data.ecb.europa.eu/data/datasets/EXR/EXR.A.SEK.EUR.SP00.A>
- [22] *EXR.A.USD.EUR.SP00.A* / *ECB Data Portal*.
<https://data.ecb.europa.eu/data/datasets/EXR/EXR.A.USD.EUR.SP00.A>
- [23] Fernández, E. C. (2024, 31 enero). *¿Qué es y para qué sirve la librería Python Request?* Tokio School. <https://www.tokioschool.com/noticias/python-request/>
- [24] Fundación Telefónica España. (2023, 24 octubre). *Acerca de Telefónica* / *Fundación Telefónica España*. <https://www.fundaciontelefonica.com/conocenos/acerca-de-telefonica/#:~:text=Telef%C3%B3nica%20es%20uno%20de%20los,digitales%20para%20p%C3%A1ginas%20web%20y%20aplicaciones,articulos%20de%20la%20fundaci%C3%B3n%20de%20telefon%C3%ADa%20de%20espa%C3%B1a.>
- [25] *¿Hacia dónde apunta el sector financiero en la era de los datos abiertos? – Ebanking News*. <https://www.ebankingnews.com/noticias/hacia-donde-apunta-el-sector-financiero-en-la-era-de-los-datos-abiertos-0058433>
- [26] *Introducción a Dash, diseño de una aplicación Dash y principales componentes* / *AI Planet (formerly DPhi)*. AI Planet (Formerly DPhi). <https://aiplanet.com/learn/data-visualization-with-plotly-and-dash-es/crear-aplicaciones-de-dashboard-usando-dash/1645/introduccion-a-dash-diseno-de-una-aplicacion-dash-y-principales-componentes>
- [27] *Introducción a Python*. Google For Developers.
<https://developers.google.com/edu/python/introduction?hl=es-419#:~:text=Python%20is%20a%20dynamic%20and%20interpreted,checking%20of%20the%20source%20code.>

- [28] J.Samper. (2024, 22 febrero). *¿Qué es el activo corriente? Clasificación, fórmula y ejemplos*. Economipedia. <https://economipedia.com/definiciones/activo-corriente.html>
- [29] Karani, D. (2021, 13 diciembre). How does Python work? - Towards Data Science. *Medium*. <https://towardsdatascience.com/how-does-python-work-6f21fd197888>
- [30] *La CNMC publica la relación de operadores principales en los mercados de telefonía fija y móvil* | CNMC. <https://www.cnmc.es/prensa/operadores-principales-telecos-20231011#:~:text=Telef%C3%B3nica%2C%20Orange%2C%20Vodafone%2C%20M%C3%A1sM%C3%B3vil,primeros%20puestos%20en%20ambos%20mercados>.
- [31] Losapuntesde. (2024, 4 febrero). Uso de la biblioteca requests para solicitudes HTTP en Python: Manejo de solicitudes y respuestas HTTP. *Apuntes de Programador*. <https://apuntes.de/python/uso-de-la-biblioteca-requests-para-solicitudes-http-en-python-manejo-de-solicitudes-y-respuestas-http/#gsc.tab=0>
- [32] Lucena, P. (2023, 10 mayo). *¿Qué es el activo y el pasivo en la contabilidad?* | 2024. Maestrías y MBA. <https://www.cesuma.mx/blog/que-es-el-activo-y-el-pasivo-en-la-contabilidad.html#:~:text=En%20t%C3%A9rminos%20sencillos%2C%20un%20activo,dinero%20o%20derechos%20de%20autor>.
- [33] Medrano, S. (2023, 3 marzo). *Open Data Day y fintech. Una nueva era para los datos abiertos*. Expansión. <https://expansion.mx/opinion/2023/03/03/open-data-day-fintech-nueva-era-datos-abiertos>
- [34] Microsoft. *Microsoft se asocia con el sector de las telecomunicaciones para implementar la tecnología 5G y mucho más*. Recuperado de <https://azure.microsoft.com/es-es/blog/microsoft-partners-with-the-telecommunications-industry-to-roll-out-5g-and-more/>
- [35] Navarro, A. (2023, 8 febrero). *Solicitudes HTTP en Python con Requests*. Junco TIC. <https://juncotic.com/solicitudes-http-en-python-con-requests/>

- [36] Pablo. (2023, 26 abril). *Orange España confirma su evolución positiva en el primer trimestre del año y alcanza unos ingresos de 1.164 millones de euros*. El Blog de Orange. <https://blog.orange.es/noticias/evolucion-positiva-primer-trimestre-2023/>
- [37] Pariente, R. (2023, 29 septiembre). *¿Qué es el fondo de comercio?* BBVA NOTICIAS. <https://www.bbva.com/es/salud-financiera/que-es-el-fondo-de-comercio/>
- [38] QbD Group. (2022, 21 noviembre). *Requisitos de usuario*. <https://qbdgroup.com/es-es/blog/validacion-de-sistemas-informatizados-requisitos-de-usuario-urs/>
- [39] *¿Qué es el beneficio neto?* | Glosario de Square. Square. <https://squareup.com/es/es/glossary/beneficio-neto>
- [40] *¿Qué es el Resultado Operacional, cómo se construye y para qué sirve?* | Centro de Ayuda :: Chipax. <https://ayuda.chipax.com/es/articles/2869345-que-es-el-resultado-operacional-como-se-construye-y-para-que-sirve>
- [41] *¿Qué es Python? - Explicación del lenguaje Python - AWS*. Amazon Web Services, Inc. <https://aws.amazon.com/es/what-is/python/#:~:text=Python%20permite%20que%20los%20desarrolladores,reutilizables%20para%20casi%20cualquier%20tarea.>
- [42] *¿Qué es una API y cómo funciona?* <https://www.redhat.com/es/topics/api/what-are-application-programming-interfaces>
- [43] *¿Qué es una interfaz de programación de aplicaciones (API)?* | IBM. <https://www.ibm.com/es-es/topics/api>
- [44] *Qué son los ingresos* | Glosario de Square. Square. <https://squareup.com/es/es/glossary/ingresos#:~:text=Los%20ingresos%20brutos%20se%20corresponden,alquiler%2C%20materias%20primas%20o%20salarios.>
- [45] Quipu, R. D.-. C. (2024, 2 febrero). *ROA y ROE: qué son y cómo calcularlos*. Autónomos, Empresas y Asesorías. <https://getquipu.com/blog/que-es-el-roa-y-el-roe/>
- [46] Rankia. *Securities and Exchange Commission (SEC)*. Recuperado de <https://www.rankia.com/diccionario/bolsa/securities-and-exchange-commission-sec>

- [47] Redacción, L. (2023, 7 marzo). *U-GOB*. u-GOB. https://u-gob.com/5-cosas-que-los-datos-abiertos-han-hecho-por-el-sector-financiero/#google_vignette
- [48] Ruiz, F. (2023, 3 noviembre). *Los Datos Abiertos harán la revolución financiera*. Finerio Connect. <https://blog.finerioconnect.com/los-datos-abiertos-haran-la-revolucion-financiera/#:~:text=Los%20datos%20abiertos%20en%20finanzas%20se%20refieren%20a%20la%20pr%C3%A1ctica,acceder%2C%20utilizar%20y%20redistribuir%20libremente.>
- [49] Sage. (2024, 19 febrero). *Activo no corriente: qué significa el término* | Sage Advice. Sage Advice España. <https://www.sage.com/es-es/blog/diccionario-empresarial/activo-no-corriente/>
- [50] SEC EDGAR Filings API. *SEC EDGAR Filings API*. <https://sec-api.io/>
- [51] *SEC.gov | EDGAR Application Programming Interfaces*. (2021, 9 junio). <https://www.sec.gov/edgar/sec-api-documentation>
- [52] Sociales, R. (s. f.-b). Los activos intangibles de una empresa: ¿Qué son? Ejemplos (infografía). *Bankinter*. <https://www.bankinter.com/blog/empresas/activos-intangibles-empresa-que-son>
- [53] Statista. (2023, 21 noviembre). *Ranking mundial de las empresas tecnológicas con mayor número de trabajadores en 2021*. <https://es.statista.com/estadisticas/657341/ranking-mundial-de-las-empresas-tecnologicas-con-mayor-numero-de-trabajadores/>
- [54] SYDLE. *Post title | SYDLE*. <https://www.sydle.com/es/blog/api-6214f68876950e47761c40e7>
- [55] Telefónica. (2023, 17 agosto). *Misión - telefónica*. <https://www.telefonica.com/es/nosotros/mision/>
- [56] *Una de las mayores compañías de redes de telecomunicaciones*. Vodafone Particulares. <https://www.vodafone.es/c/conocenos/es/vodafone-espana/quienes-somos/grupo-vodafone-espana/#:~:text=Vodafone%20proporciona%20servicios%20m%C3%B3viles%20y,millones%20de%20dispositivos%20y%20plataformas.>



- [57] Unir, V. (2023, 7 noviembre). ¿Qué es el patrimonio neto y cómo se calcula? *UNIR*.
<https://www.unir.net/empresa/revista/patrimonio-neto/>
- [58] Vodafone España. (2020, 5 de febrero). *Vodafone España ingresa 966,4 millones de euros por servicio en el tercer trimestre del año fiscal y sigue creciendo en Clientes móviles de contrato, fibra y televisión*. Recuperado de
https://www.saladeprensa.vodafone.es/c/notas-prensa/np_resultados_Q3_FY19/20/
- [59] *We are Nokia / Nokia*. Nokia. <https://www.nokia.com/we-are-nokia/>
- [60] Zapata, J. R. *PANDAS - Manipulacion de Datos con Python / Jose Ricardo Zapata*. Jose Ricardo Zapata. <https://joserzapata.github.io/courses/python-ciencia-datos/pandas/>
- [61] Zaragoza, J. M. (2015, 16 agosto). El ABC de Alphabet, el nuevo Google. *heraldo.es*.
<https://www.heraldo.es/noticias/sociedad/2015/08/16/que-es-alphabet-259233.html>

Capítulo 7. Anexo

7.1 Manual de usuario

El siguiente manual de usuario constituye un recurso esencial que detalla la operatividad y las funcionalidades del software desarrollado. La importancia de este apartado radica en su capacidad para proporcionar instrucciones claras y concisas, asegurando que el usuario pueda maximizar el potencial de la herramienta sin necesidad de asistencia adicional.

1. Para iniciar la aplicación, se requiere ejecutar el código en un entorno de desarrollo integrado (IDE) adecuado, tal como Visual Studio Code, PyCharm, Jupyter Notebook, o cualquier otro entorno que soporte la ejecución de scripts en Python. Estos entornos proporcionan las herramientas necesarias para correr, depurar y gestionar el código de manera eficiente.

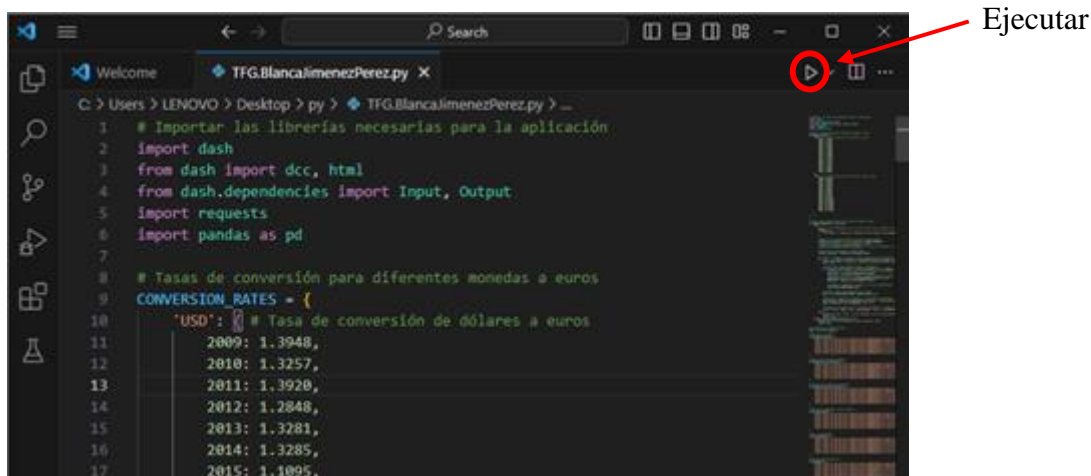


Ilustración 3. Ejecutar código en visualizador

Fuente: Captura código creado

2. Tras ejecutar el código, se mostrará un mensaje en la consola de Terminal que incluirá una URL. Al presionar simultáneamente la tecla 'Control' y hacer clic en el enlace mostrado, se abrirá en su navegador web predeterminado la aplicación Dash que ha sido desarrollada.

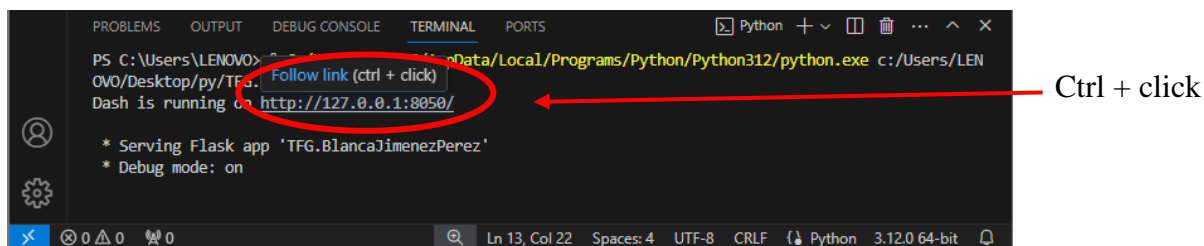


Ilustración 4. Abrir link para abrir aplicación Dash

Fuente: Captura código creado

3. Al acceder al enlace proporcionado a través de su navegador, se presentará la interfaz de la aplicación como se ilustra en la captura de pantalla adjunta. En esta vista inicial, se destacan tres grupos claramente diferenciados de empresas —Operadoras, Fabricantes y Prestadores de Servicios— que están disponibles para la selección y análisis. Además, se mostrarán cuatro menús desplegables que permiten a los usuarios ajustar y personalizar la visualización de los datos según distintos parámetros como la empresa, la fecha, el tipo de dato y el tipo de gráfico deseado.

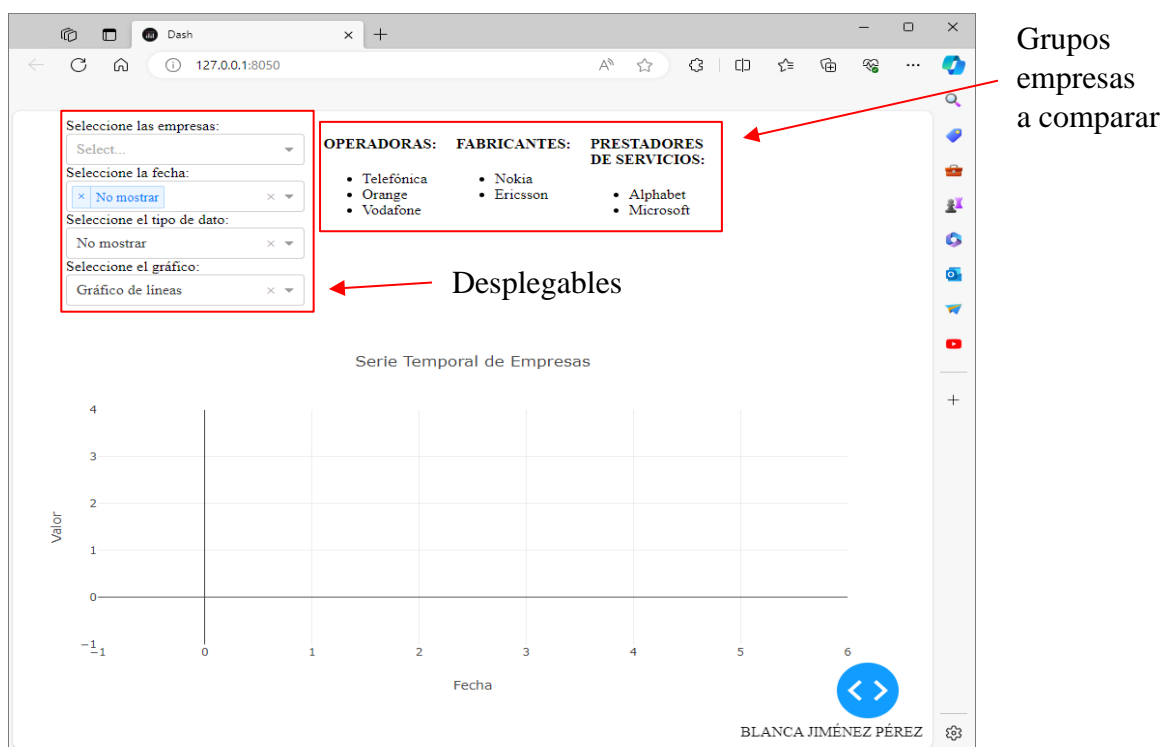


Ilustración 5. Aplicación Dash inicial

Fuente: Captura aplicación Dash creado

4. A continuación, el usuario deberá proceder a la selección de empresas a través del primer menú desplegable 'Seleccione las empresas'. Para facilitar un análisis coherente, es recomendable comparar empresas dentro del mismo grupo mostrado a la derecha del desplegable; no obstante, la aplicación permite seleccionar múltiples empresas de cualquier categoría. Desplazándose por la barra del menú, se desvelarán más opciones de empresas. También es posible teclear directamente el nombre de una empresa en el campo de búsqueda del menú para una selección rápida. Para deseleccionar una empresa específica, basta con hacer clic en la 'x' junto a su nombre. Si se desea eliminar todas las empresas seleccionadas de una sola vez, se puede hacer clic en la 'x' ubicada en el campo del menú desplegable.

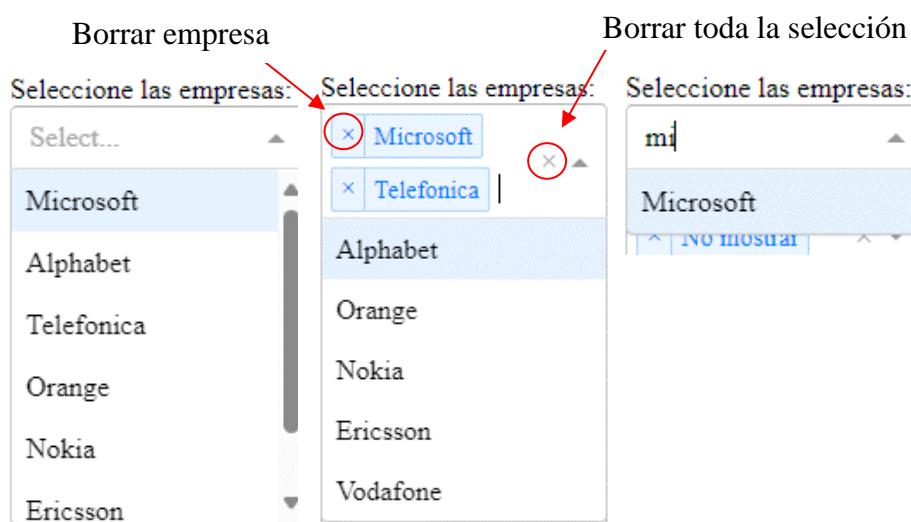


Ilustración 6. Desplegable 'Seleccione las empresas'

Fuente: Captura aplicación Dash creado

5. El procedimiento es similar con el segundo menú desplegable 'Seleccione la fecha1', que controla la selección de fechas. Es importante tener en cuenta que si la opción 'No mostrar' se encuentra seleccionada, no se visualizará ningún dato. Este menú ofrece la flexibilidad de mostrar todas las fechas disponibles mediante la opción 'Visualizar todos los años', seleccionar una única fecha específica o elegir múltiples fechas que se representarán de forma simultánea en función de las empresas escogidas previamente. También es posible buscar escribiendo en el buscador el año que se desea.

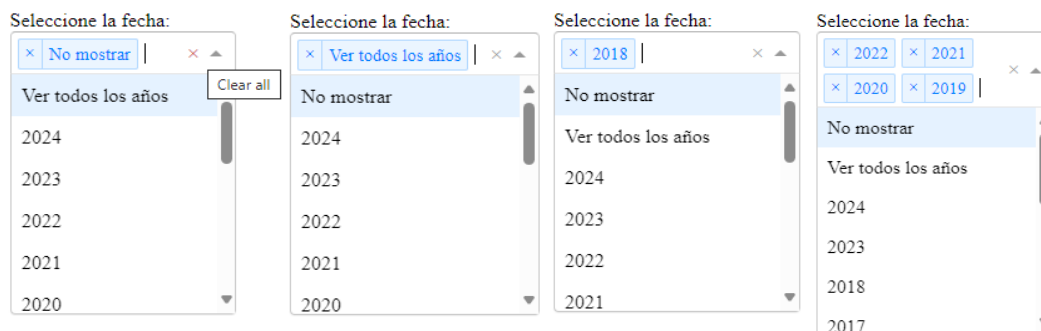


Ilustración 7. Desplegable 'Seleccione la fecha'

Fuente: Captura aplicación Dash creado

6. El tercer menú desplegable 'Seleccione tipo de dato' detalla los diferentes tipos de datos disponibles para el análisis. A diferencia del desplegable de fechas y empresas, aquí sólo es posible seleccionar un tipo de dato a la vez, ya que comparar simultáneamente diferentes tipos de datos para múltiples empresas y fechas podría resultar confuso y menos práctico para el análisis. Al igual que en el desplegable de fechas, existe una opción 'No mostrar' que, si se selecciona, ocultará los datos del análisis. Al elegir un nuevo tipo de dato del menú, este reemplazará automáticamente al que estaba previamente seleccionado, actualizando la visualización de acuerdo con la nueva selección. También se puede buscar directamente el tipo de dato deseado.

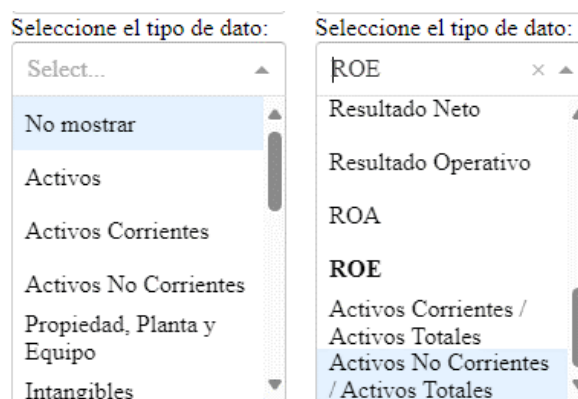


Ilustración 8. Desplegable 'Selecciona el tipo de dato'

Fuente: Captura aplicación Dash creado

7. El último menú desplegable permite elegir el formato de visualización de los datos: gráfico de líneas o de barras. El gráfico de barras se recomienda para un número reducido de empresas —por ejemplo, una, dos o tres— ya que facilita una comparativa clara entre pocas entidades. Por otro lado, el gráfico de líneas es preferible cuando se analizan múltiples empresas, ya que proporciona una

visualización más clara de tendencias y patrones a lo largo del tiempo. Al igual que con los menús anteriores, este desplegable también incluye una función de búsqueda que permite localizar y seleccionar rápidamente el tipo de gráfico deseado.

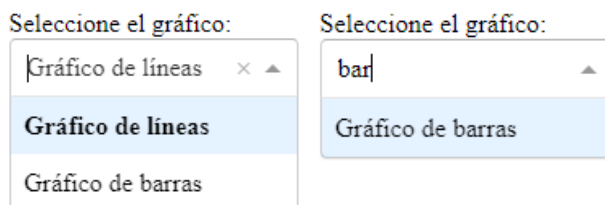


Ilustración 9. Desplegable 'Seleccione el gráfico'

Fuente: Captura aplicación Dash creado

- Una vez que se hayan realizado selecciones en todos los menús desplegables, los datos correspondientes se representarán visualmente en los gráficos. Al desplazar el cursor sobre los puntos de datos individuales en el gráfico, se mostrarán detalles adicionales, proporcionando cifras exactas y facilitando una lectura más precisa de los valores específicos presentados.

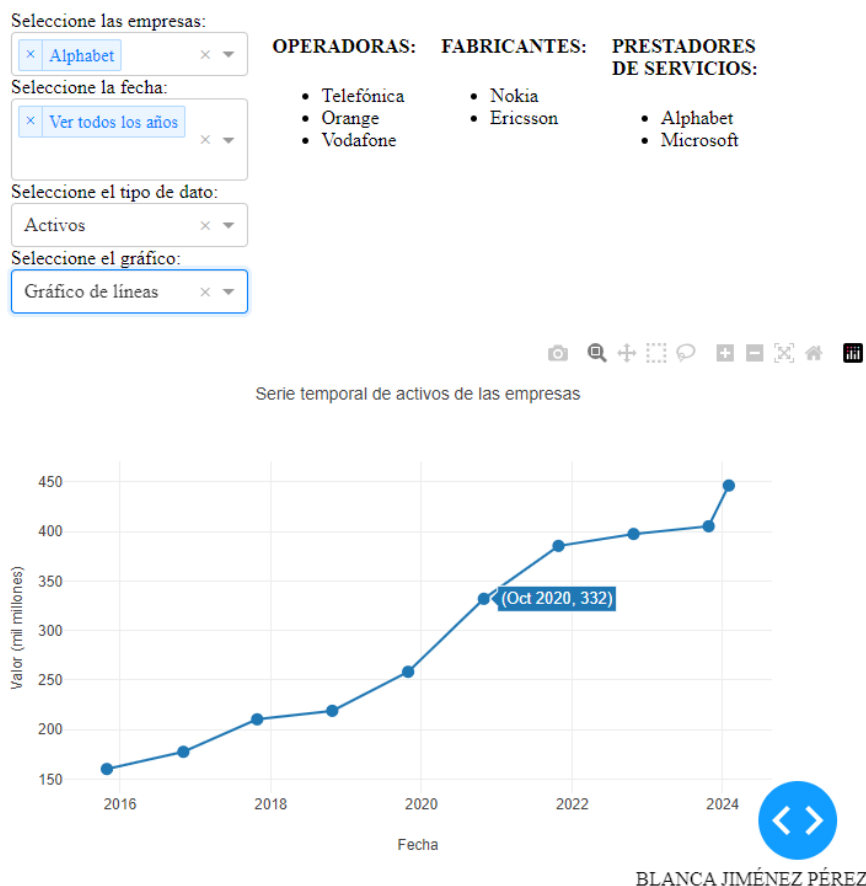


Ilustración 10. Ejemplo final gráfico líneas

Fuente: Captura aplicación Dash creado

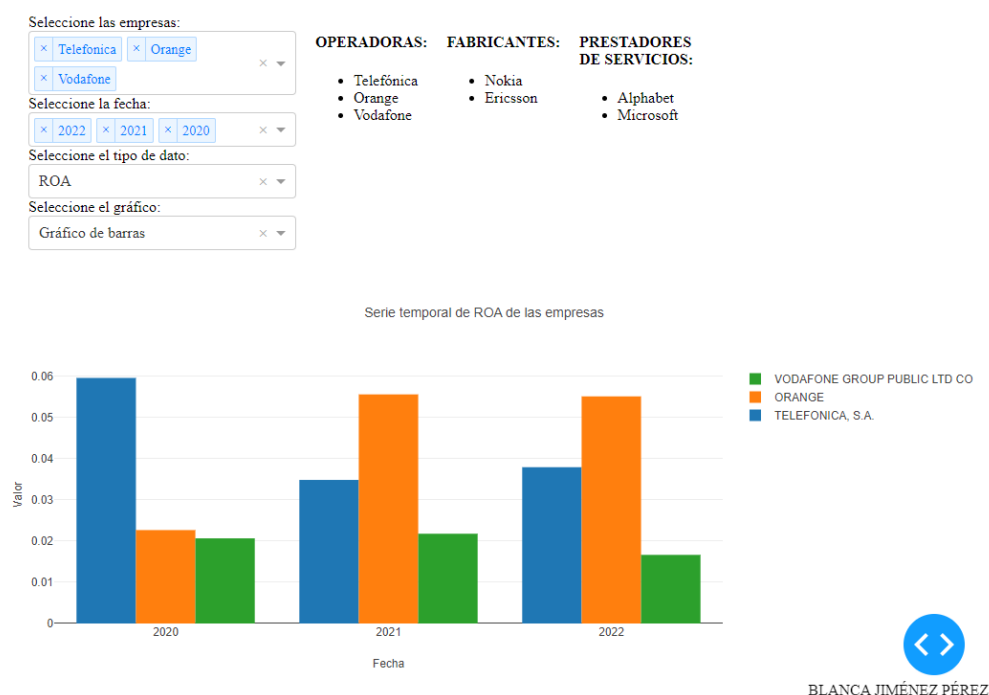


Ilustración 11. Ejemplo final gráfico barras

Fuente: Captura aplicación Dash creado

Es crucial considerar algunos detalles operativos al utilizar la aplicación. Primero, si el entorno de desarrollo o la ventana de ejecución de Python se cierra, la aplicación dejará de funcionar y los datos no se actualizarán en tiempo real, lo cual afectará la visualización de las gráficas. Además, es importante notar que la ausencia de datos en las gráficas para ciertas empresas o fechas no indica un mal funcionamiento de la aplicación; esto se debe a que algunos datos, como los correspondientes a la deuda a corto plazo de ciertas empresas, pueden no estar disponibles públicamente en la base de datos de la SEC. Esta limitación en la disponibilidad de datos debe ser tomada en cuenta durante el análisis y la interpretación de los gráficos generados.

Finalmente, es importante prestar atención al indicador de actualización en la parte superior de la pantalla. Cuando se visualiza el mensaje "Updating...", significa que la aplicación está procesando y recuperando los datos requeridos. Durante este tiempo, es necesario esperar a que el proceso finalice, tras lo cual el gráfico se actualizará automáticamente con la nueva información.



Ilustración 12. Mensaje updating

Fuente: Captura aplicación Dash creado

7.2 Manual de administrador

Este documento sirve como un manual de administrador para la aplicación Dash diseñada para el análisis de datos financieros. Su propósito es orientar al administrador a través de las diversas responsabilidades técnicas asociadas con la gestión y el mantenimiento de la aplicación. Aquí se detallan los procedimientos esenciales para asegurar que la aplicación funcione de manera óptima y segura, proporcionando un entorno estable para los usuarios finales.

Configuración del Entorno: Antes de la implementación, es fundamental asegurarse de que el entorno de servidor cumpla con los requisitos necesarios para ejecutar la aplicación Dash. Esto incluye la instalación de Python, las librerías pertinentes, y un servidor web adecuado. Deberá también configurarse el entorno virtual de Python para aislar las dependencias del proyecto. Los siguientes pasos preparatorios son necesarios para la funcionalidad integral de la aplicación y deben ejecutarse en un entorno de desarrollo compatible, como Visual Studio Code.

1. Instalación de Python: Para comenzar, es necesario instalar Python, lenguaje empleado para la creación del código. La instalación puede realizarse accediendo al sitio web oficial de Python en (<http://www.python.org>), donde se ofrecen las últimas versiones del intérprete y se pueden seguir las instrucciones detalladas para una correcta configuración en su sistema operativo.
2. Instalación de la Biblioteca Dash: Para agregar las capacidades de Dash a Python, abra la línea de comandos y ejecute el comando ``pipinstalldash``. Esto instalará la biblioteca necesaria para la creación de gráficos interactivos y componentes de la aplicación.
3. Instalación de la Biblioteca Pandas: Pandas es una herramienta esencial para la manipulación y análisis de datos. Instálela ingresando ``pipinstall pandas`` en la ventana de comandos, lo cual permitirá procesar y analizar datos con facilidad dentro de la aplicación.

4. Instalación de la Biblioteca Requests: Esta biblioteca es crucial para realizar solicitudes HTTP. Instálela ejecutando ``pipinstallrequests`` en la línea de comandos para habilitar la recuperación de datos de APIs y otras fuentes en línea.

Cabe comentar que para la implementación del código, las versiones de las bibliotecas empleadas fueron las siguientes: Python 3.12.0, Dash 2.14.2, Requests 2.31.0 y Pandas 2.1.2.

Posteriormente, para mantener la aplicación al día, se deben considerar varios aspectos de actualización y ampliación:

Actualización de Tasas de Conversión: En la sección ``CONVERSION_RATES`` del código, es imperativo incluir tasas de cambio actualizadas de USD y SEK a EUR para cada año correspondiente. Las tasas actualizadas se pueden obtener del Banco Central Europeo en la siguiente dirección:

<https://data.ecb.europa.eu/data/datasets/EXR/EXR.A.USD.EUR.SP00.A>

Ampliación de Datos Visualizados: Para mostrar más empresas o diferentes tipos de datos, se pueden añadir las URLs de los JSON deseados en las listas correspondientes, asegurándose de incluir el nombre de la empresa en el diccionario ``nombre_empresas`` para una presentación estandarizada.

Inclusión de Nuevos Tipos de Datos: Si se requiere añadir un nuevo tipo de dato que no se deriva de una URL de la SEC sino de un cálculo, este debe ser incorporado en ``app.layout`` y el procedimiento para calcular dicho valor debe ser definido en la función ``update_plot``. Si el nuevo dato proviene de una URL de la SEC, la URL para cada empresa debe ser añadida a la lista de URLs y configurada correctamente en ``app.layout`` y en la función de callback del gráfico.

Creación de Nuevos Tipos de Gráficos: La introducción de un nuevo formato de visualización gráfica implica su inclusión en la sección ``html.Label`` correspondiente y la adecuada configuración en el conjunto de trazos (``trace``) para la representación de los datos.

El propósito de este manual de administrador es ofrecer una breve guía para la actualización y expansión de la aplicación Dash. Siguiendo estas pautas técnicas, los administradores podrán garantizar que la aplicación se mantenga actualizada, ampliar su



alcance y funcionalidades, y mejorar continuamente la experiencia del usuario. Con una implementación meticulosa y un mantenimiento regular, la aplicación no solo servirá como una herramienta educativa para la asignatura de Política de Telecomunicaciones, sino que también proporcionará una plataforma robusta para la exploración de datos financieros en el ámbito académico y profesional.

7.3 Código

A continuación se adjunta todo el código creado para este trabajo con los comentarios explicativos del mismo:

```
1. # Importar las librerías necesarias para la aplicación
2. importdash
3. fromdashimportdcc, html
4. fromdash.dependenciesimportInput, Output
5. importrequests
6. importpandasaspd
7.
8. # Tasas de conversión para diferentes monedas a euros
9. CONVERSION_RATES= {
10.     'USD': { # Tasa de conversión de dólares a euros
11.         2009: 1.3948,
12.         2010: 1.3257,
13.         2011: 1.3920,
14.         2012: 1.2848,
15.         2013: 1.3281,
16.         2014: 1.3285,
17.         2015: 1.1095,
18.         2016: 1.1069,
19.         2017: 1.1297,
20.         2018: 1.1810,
21.         2019: 1.1195,
22.         2020: 1.1422,
23.         2021: 1.1827,
24.         2022: 1.0530,
25.         2023: 1.0813,
26.     },
27.     'SEK': { # Tasa de conversión de coronas suecas a euros
28.         2009: 10.6191,
29.         2010: 9.5373,
30.         2011: 9.0298,
31.         2012: 8.7041,
32.         2013: 8.6515,
33.         2014: 9.0985,
34.         2015: 9.3535,
35.         2016: 9.4689,
36.         2017: 9.6351,
37.         2018: 10.2583,
38.         2019: 10.5891,
39.         2020: 10.4848,
40.         2021: 10.1465,
41.         2022: 10.6296,
42.         2023: 11.4788,
```

```
43.
44.     }
45. }
46.
47. # Función para obtener datos de la SEC utilizando una API
48. def get_sec_data(url, urls_list):
49.     try:
50.         # Definir los encabezados de la solicitud para evitar bloqueos
           por el servidor
51.         headers= {
52.             'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64)
           AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.124
           Safari/537.36'
53.         }
54.
55.         # Realizar la solicitud HTTP GET a la URL proporcionada
56.         tickers_cik_response=requests.get(url, headers=headers)
57.         # Verificar que la solicitud haya sido exitosa
58.         tickers_cik_response.raise_for_status() # Lanza una excepción
           si la solicitud no fue exitosa
59.
60.         # Convertir la respuesta en formato JSON
61.         tickers_cik_data=tickers_cik_response.json()
62.
63.         # Procesar los datos si la clave 'units' existe y es un
           diccionario
64.         if 'units' in tickers_cik_data and isinstance(tickers_cik_data['uni
           ts'], dict):
65.             data_frames= [] # Lista para almacenar DataFrames de cada
           unidad de datos
66.
67.             # Iterar sobre cada unidad de datos y procesarla
68.             for unit_key,
           unit_values in tickers_cik_data['units'].items():
69.                 # Normalizar los datos JSON y convertirlos en un
           DataFrame
70.                 data_unit=pd.json_normalize(unit_values)
71.                 # Convertir la columna "filed" a formato datetime y
           ordenar por fecha
72.                 data_unit["filed"] =pd.to_datetime(data_unit["filed"])
73.                 data_unit=data_unit.sort_values("filed")
74.
75.                 # Agregar el DataFrame procesado a la lista de
           DataFrames
76.                 data_frames.append(data_unit)
77.
78.             # Combinar todos los DataFrames en uno solo
```

```
79.         data=pd.concat(data_frames, ignore_index=True)
80.
81.         # Extraer el nombre de la entidad y el tag a partir de la
URL
82.         entity_name=tickers_cik_data.get('entityName',
f'NombreEmpresa{urls_list.index(url) +1}')
83.         tag=url.split("/")[-1].replace(".json", "")
84.
85.         # Devolver un diccionario con los datos procesados, el
nombre de la entidad, las unidades y el tag
86.         return {'data': data, 'name': entity_name, 'units':
list(tickers_cik_data['units'].keys()), 'tag': tag}
87.     else:
88.         print(f"La clave 'units' no está presente o no es un
diccionario en el JSON de {url}.")
89.         returnNone
90.     exceptrequests.RequestException:
91.         print(f"Error en la solicitud de {url}. Detalles: {e}")
92.         returnNone
93.
94.# Inicializar la aplicación Dash
95.app=dash.Dash(__name__)
96.
97.# URLs de los activos
98.urls_empresas_assets= [
99.     "https://data.sec.gov/api/xbrl/companyconcept/CIK0000789019/us-
gaap/Assets.json", # Assets Microsoft
100.     "https://data.sec.gov/api/xbrl/companyconcept/CIK0001652044/
us-gaap/Assets.json", # Assets Alphabet
101.     "https://data.sec.gov/api/xbrl/companyconcept/CIK0000814052/
ifrs-full/Assets.json", #Assets Telefónica
102.     "https://data.sec.gov/api/xbrl/companyconcept/CIK0001038143/
ifrs-full/Assets.json", # Assets Orange
103.     "https://data.sec.gov/api/xbrl/companyconcept/CIK0000924613/
ifrs-full/Assets.json", # Assets Nokia
104.     "https://data.sec.gov/api/xbrl/companyconcept/CIK0000717826/
ifrs-full/Assets.json", # Assets Ericsson
105.     "https://data.sec.gov/api/xbrl/companyconcept/CIK0000839923/
ifrs-full/Assets.json", # Assets Vodafone
106. ]
107.
108. # URLs de los activos corrientes
109. urls_empresas_currentassets= [
110.     "https://data.sec.gov/api/xbrl/companyconcept/CIK0000789019/
us-gaap/AssetsCurrent.json", # AC Microsoft
111.     "https://data.sec.gov/api/xbrl/companyconcept/CIK0001652044/
us-gaap/AssetsCurrent.json", # AC Alphabet
```

```
112.     "https://data.sec.gov/api/xbrl/companyconcept/CIK0000814052/  
ifrs-full/CurrentAssets.json", # AC Telefónica  
113.     "https://data.sec.gov/api/xbrl/companyconcept/CIK0001038143/  
ifrs-full/CurrentAssets.json", # AC Orange  
114.     "https://data.sec.gov/api/xbrl/companyconcept/CIK0000924613/  
ifrs-full/CurrentAssets.json",# AC Nokia  
115.     "https://data.sec.gov/api/xbrl/companyconcept/CIK0000717826/  
ifrs-full/CurrentAssets.json", # AC Ericsson  
116.     "https://data.sec.gov/api/xbrl/companyconcept/CIK0000839923/  
ifrs-full/CurrentAssets.json", # AC Vodafone  
117.     ]  
118.  
119.     # URLs de los activos no corrientes  
120.     urls_empresas_noncurrentassets= [  
121.         "https://data.sec.gov/api/xbrl/companyconcept/CIK0000789019/  
us-gaap/NoncurrentAssets.json", # ANC Microsoft  
122.         "https://data.sec.gov/api/xbrl/companyconcept/CIK0001652044/  
us-gaap/NoncurrentAssets.json", # ANC Alphabet  
123.         "https://data.sec.gov/api/xbrl/companyconcept/CIK0000814052/  
ifrs-full/NoncurrentAssets.json", # ANC Telefónica  
124.         "https://data.sec.gov/api/xbrl/companyconcept/CIK0001038143/  
ifrs-full/NoncurrentAssets.json", # ANC Orange  
125.         "https://data.sec.gov/api/xbrl/companyconcept/CIK0000924613/  
ifrs-full/NoncurrentAssets.json",# ANC Nokia  
126.         "https://data.sec.gov/api/xbrl/companyconcept/CIK0000717826/  
ifrs-full/NoncurrentAssets.json", # ANC Ericsson  
127.         "https://data.sec.gov/api/xbrl/companyconcept/CIK0000839923/  
ifrs-full/NoncurrentAssets.json", # ANC Vodafone  
128.     ]  
129.  
130.     # URLs de la propiedad, planta y equipo  
131.     urls_nuevas_ppae= [  
132.         "https://data.sec.gov/api/xbrl/companyconcept/CIK0000789019/  
us-gaap/PropertyPlantAndEquipmentGross.json", # PPAE Microsoft  
133.         "https://data.sec.gov/api/xbrl/companyconcept/CIK0001652044/  
us-gaap/PropertyPlantAndEquipmentGross.json", # PPAE Alphabet  
134.         "https://data.sec.gov/api/xbrl/companyconcept/CIK0000814052/  
ifrs-full/PropertyPlantAndEquipment.json", # PPAE Telefónica  
135.         "https://data.sec.gov/api/xbrl/companyconcept/CIK0001038143/  
ifrs-full/PropertyPlantAndEquipment.json", # PPAE Orange  
136.         "https://data.sec.gov/api/xbrl/companyconcept/CIK0000924613/  
ifrs-full/PropertyPlantAndEquipment.json", # PPAE Nokia  
137.         "https://data.sec.gov/api/xbrl/companyconcept/CIK0000717826/  
ifrs-full/PropertyPlantAndEquipment.json", # PPAE Ericsson  
138.         "https://data.sec.gov/api/xbrl/companyconcept/CIK0000839923/  
ifrs-full/PropertyPlantAndEquipment.json", # PPAE Vodafone  
139.     ]
```

```
140.
141.     # URLs de intangibles
142.     urls_intangibles= [
143.         "https://data.sec.gov/api/xbrl/companyconcept/CIK0000789019/
us-gaap/IntangibleAssetsNetExcludingGoodwill.json", # Intangibles
Microsoft
144.         "https://data.sec.gov/api/xbrl/companyconcept/CIK0001652044/
us-gaap/IntangibleAssetsNetExcludingGoodwill.json", # Intangibles
Alphabet
145.         "https://data.sec.gov/api/xbrl/companyconcept/CIK0000814052/
ifrs-full/IntangibleAssetsOtherThanGoodwill.json", # Intangibles
Telefónica
146.         "https://data.sec.gov/api/xbrl/companyconcept/CIK0001038143/
ifrs-full/IntangibleAssetsOtherThanGoodwill.json", # Intangibles
Orange
147.         "https://data.sec.gov/api/xbrl/companyconcept/CIK0000924613/
ifrs-full/IntangibleAssetsOtherThanGoodwill.json", # Intangibles
Nokia
148.         "https://data.sec.gov/api/xbrl/companyconcept/CIK0000717826/
ifrs-full/IntangibleAssetsOtherThanGoodwill.json", # Intangibles
Ericsson
149.         "https://data.sec.gov/api/xbrl/companyconcept/CIK0000839923/
ifrs-full/IntangibleAssetsOtherThanGoodwill.json", # Intangibles
Vodafone
150.     ]
151.
152.     # URLs de Fondo de Comercio
153.     urls_goodwill= [
154.         "https://data.sec.gov/api/xbrl/companyconcept/CIK0000789019/
us-gaap/Goodwill.json", # Goodwill Microsoft
155.         "https://data.sec.gov/api/xbrl/companyconcept/CIK0001652044/
us-gaap/Goodwill.json", # Goodwill Alphabet
156.         "https://data.sec.gov/api/xbrl/companyconcept/CIK0000814052/
ifrs-full/Goodwill.json", #Goodwill Telefónica
157.         "https://data.sec.gov/api/xbrl/companyconcept/CIK0001038143/
ifrs-full/Goodwill.json", # Goodwill Orange
158.         "https://data.sec.gov/api/xbrl/companyconcept/CIK0000924613/
ifrs-full/Goodwill.json", # Goodwill Nokia
159.         "https://data.sec.gov/api/xbrl/companyconcept/CIK0000717826/
ifrs-full/Goodwill.json", # Goodwill Ericsson
160.         "https://data.sec.gov/api/xbrl/companyconcept/CIK0000839923/
ifrs-full/Goodwill.json", # Goodwill Vodafone
161.     ]
162.
163.     # URLs de ingresos
164.     urls_ingresos= [
```



```
165.     "https://data.sec.gov/api/xbrl/companyconcept/CIK0000789019/
us-gaap/Revenues.json", # Revenue Microsoft
166.     "https://data.sec.gov/api/xbrl/companyconcept/CIK0001652044/
us-gaap/Revenues.json", # Revenue Alphabet!
167.     "https://data.sec.gov/api/xbrl/companyconcept/CIK0000814052/
ifrs-full/Revenue.json", #Revenue Telefónica
168.     "https://data.sec.gov/api/xbrl/companyconcept/CIK0001038143/
ifrs-full/Revenue.json", # Revenue Orange
169.     "https://data.sec.gov/api/xbrl/companyconcept/CIK0000924613/
ifrs-full/Revenue.json", # Revenue Nokia
170.     "https://data.sec.gov/api/xbrl/companyconcept/CIK0000717826/
ifrs-full/Revenue.json", # Revenue Ericsson
171.     "https://data.sec.gov/api/xbrl/companyconcept/CIK0000839923/
ifrs-full/Revenue.json", # Revenue Vodafone
172. ]
173.
174.     # URLs de deuda a corto plazo
175.     urls_deuda_corto_plazo= [
176.         "https://data.sec.gov/api/xbrl/companyconcept/CIK0000789019/
us-gaap/ShortTermBorrowings.json", #ShortTermBorrowings Microsoft
177.         "", # ShortTermBorrowingsAlphabet
178.         "", # ShortTermBorrowingsTelefonica
179.         "", # ShortTermBorrowings Orange
180.         "", # ShortTermBorrowings Nokia
181.         "https://data.sec.gov/api/xbrl/companyconcept/CIK0000717826/
ifrs-full/ShorttermBorrowings.json", #ShortTermBorrowings Ericsson
182.         "https://data.sec.gov/api/xbrl/companyconcept/CIK0000839923/
ifrs-full/ShorttermBorrowings.json", #ShortTermBorrowings Vodafone
183.     ]
184.
185.     # URLs de deuda a largo plazo
186.     urls_deuda_largo_plazo= [
187.         "https://data.sec.gov/api/xbrl/companyconcept/CIK0000789019/
us-gaap/LongTermDebt.json", #LongTermDebt Microsoft
188.         "https://data.sec.gov/api/xbrl/companyconcept/CIK0001652044/
us-gaap/LongTermDebt.json", #LongTermDebt Alphabet
189.         "https://data.sec.gov/api/xbrl/companyconcept/CIK0000814052/
ifrs-full/LongtermBorrowings.json", #LongtermBorrowings Telefónica
190.         "https://data.sec.gov/api/xbrl/companyconcept/CIK0001038143/
ifrs-full/LongtermBorrowings.json", #LongtermBorrowings Orange
191.         "https://data.sec.gov/api/xbrl/companyconcept/CIK0000924613/
ifrs-full/LongtermBorrowings.json", #LongtermBorrowings Nokia
192.         "https://data.sec.gov/api/xbrl/companyconcept/CIK0000717826/
ifrs-full/LongtermBorrowings.json", #LongtermBorrowings Ericsson
193.         "https://data.sec.gov/api/xbrl/companyconcept/CIK0000839923/
ifrs-full/LongtermBorrowings.json", #LongtermBorrowings Vodafone
194.     ]
```

```
195.
196.     # URLs de patrimonio neto
197.     urls_patrimonio_netos= [
198.         "https://data.sec.gov/api/xbrl/companyconcept/CIK0000789019/
us-gaap/StockholdersEquity.json", # PN Microsoft
199.         "https://data.sec.gov/api/xbrl/companyconcept/CIK0001652044/
us-gaap/StockholdersEquity.json", # PN Alphabet
200.         "https://data.sec.gov/api/xbrl/companyconcept/CIK0000814052/
ifrs-full/Equity.json", # PN Telefónica
201.         "https://data.sec.gov/api/xbrl/companyconcept/CIK0001038143/
ifrs-full/Equity.json", # PN Orange
202.         "https://data.sec.gov/api/xbrl/companyconcept/CIK0000924613/
ifrs-full/Equity.json", # PN Nokia
203.         "https://data.sec.gov/api/xbrl/companyconcept/CIK0000717826/
ifrs-full/Equity.json", # PN Ericsson
204.         "https://data.sec.gov/api/xbrl/companyconcept/CIK0000839923/
ifrs-full/Equity.json", # PN Vodafone
205.     ]
206.
207.     # URLs de resultado neto
208.     urls_resultado_netos= [
209.         "https://data.sec.gov/api/xbrl/companyconcept/CIK0000789019/
us-gaap/NetIncomeLoss.json", # Rdo Neto Microsoft
210.         "https://data.sec.gov/api/xbrl/companyconcept/CIK0001652044/
us-gaap/NetIncomeLoss.json", # Rdo Neto Alphabet
211.         "https://data.sec.gov/api/xbrl/companyconcept/CIK0000814052/
ifrs-full/ProfitLoss.json", # Rdo Neto Telefónica
212.         "https://data.sec.gov/api/xbrl/companyconcept/CIK0001038143/
ifrs-full/ProfitLoss.json", # Rdo Neto Orange
213.         "https://data.sec.gov/api/xbrl/companyconcept/CIK0000924613/
ifrs-full/ProfitLoss.json", # Rdo Neto Nokia
214.         "https://data.sec.gov/api/xbrl/companyconcept/CIK0000717826/
ifrs-full/ProfitLoss.json", # Rdo Neto Ericsson
215.         "https://data.sec.gov/api/xbrl/companyconcept/CIK0000839923/
ifrs-full/ProfitLoss.json", # Rdo Neto Vodafone
216.     ]
217.
218.     # URLs de resultado operativo
219.     urls_resultado_operativo= [
220.         "https://data.sec.gov/api/xbrl/companyconcept/CIK0000789019/
us-gaap/OperatingIncomeLoss.json", # Rdo operativo Microsoft
221.         "https://data.sec.gov/api/xbrl/companyconcept/CIK0001652044/
us-gaap/OperatingIncomeLoss.json", # Rdo operativo Alphabet
222.         "https://data.sec.gov/api/xbrl/companyconcept/CIK0000814052/
ifrs-full/ProfitLossFromOperatingActivities.json", # Rdo operativo
Telefónica
```

```
223.         "https://data.sec.gov/api/xbrl/companyconcept/CIK0001038143/  
ifrs-full/ProfitLossFromOperatingActivities.json", # Rdo operativo  
Orange  
224.         "https://data.sec.gov/api/xbrl/companyconcept/CIK0000924613/  
ifrs-full/ProfitLossFromOperatingActivities.json", # Rdo operativo  
Nokia  
225.         "https://data.sec.gov/api/xbrl/companyconcept/CIK0000717826/  
ifrs-full/ProfitLossFromOperatingActivities.json", # Rdo operativo  
Ericsson  
226.         "https://data.sec.gov/api/xbrl/companyconcept/CIK0000839923/  
ifrs-full/ProfitLossFromOperatingActivities.json", # Rdo operativo  
Vodafone  
227.     ]  
228.  
229.     # Obtener los datos para todas las empresas existentes  
230.     data_empresas_assets= [get_sec_data(url, urls_empresas_assets)  
for url in urls_empresas_assets]  
231.  
232.     # Obtener todas las fechas  
233.     all_dates_years=sorted(set(pd.concat([data['data']['filed']  
fordata in data_empresas_assets]).dt.year), reverse=True)  
234.  
235.     # Definir las opciones para el selector de fechas en la interfaz  
de usuario  
236.     fecha_selector_options= [{'label': 'No mostrar', 'value':  
'none'}, {'label': 'Ver todos los años', 'value': 'all'}] + [{'label':  
str(value), 'value': str(value)} for value in all_dates_years]  
237.  
238.     # Mapear nombres de empresas a nombres más legibles y uniformes  
239.     nombre_empresas= {  
240.         "Nokia Corp": "Nokia",  
241.         "Alphabet Inc.": "Alphabet",  
242.         "ORANGE": "Orange",  
243.         "MICROSOFT CORPORATION": "Microsoft",  
244.         "ERICSSON LM TELEPHONE CO": "Ericsson",  
245.         "TELEFONICA, S.A.": "Telefonica",  
246.         "VODAFONE GROUP PUBLIC LTD CO": "Vodafone"  
247.     }  
248.  
249.     # Actualizar las opciones del desplegable para seleccionar  
empresas  
250.     empresa_options= [  
251.         {'label': nombre_empresas.get(data['name'], data['name']),  
'value': f"assets_{i}"}  
252.         for i, data in enumerate(data_empresas_assets)  
253.     ]  
254.
```

```
255.     # Inicializar la estructura principal de la aplicación Dash
256.     app.layout=html.Div([
257.         # Contenedor para la fila superior que incluye selecciones y
           títulos
258.         html.Div([
259.             # Contenedor para la columna de selección con menús
           desplegados
260.             html.Div([
261.                 # Etiqueta y menú desplegable para seleccionar
           empresas
262.                 html.Label("Seleccione las empresas:"),
263.                 dcc.Dropdown(
264.                     id='empresa-selector', # Identificador para el
           selector de empresas
265.                     options=empresa_options, # Opciones del menú
           desplegable, definidas previamente
266.                     value='none', # Valor inicial seleccionado
267.                     multi=True# Permitir selecciones múltiples
268.                 ),
269.                 # Etiqueta y menú desplegable para seleccionar
           fechas
270.                 html.Label("Seleccione la fecha:"),
271.                 dcc.Dropdown(
272.                     id='fecha-selector', # Identificador para el
           selector de fecha
273.                     options=fecha_selector_options, # Opciones de
           fecha
274.                     value='none', # Valor inicial seleccionado
275.                     multi=True# Permitir selecciones múltiples
276.                 ),
277.                 # Etiqueta y menú desplegable para seleccionar el
           tipo de dato a visualizar
278.                 html.Label("Seleccione el tipo de dato:"),
279.                 dcc.Dropdown(
280.                     id='tag-selector', # Identificador para el
           selector de tipo de dato
281.                     options=[# Opciones para el tipo de dato,
           incluyendo varios indicadores financieros
282.                               {'label': 'No mostrar', 'value': 'none'},
283.                               {'label': 'Activos', 'value': 'assets'},
284.                               {'label': 'Activos Corrientes', 'value':
           'currentassets'},
285.                               {'label': 'Activos No Corrientes', 'value':
           'noncurrentassets'},
286.                               {'label': 'Propiedad, Planta y Equipo',
           'value': 'propertyplantandequipmentgross'},
```

```
287.         {'label': 'Intangibles', 'value':
    'intangibles'},
288.         {'label': 'Fondo de Comercio', 'value':
    'goodwill'},
289.         {'label': 'Ingresos', 'value': 'revenue'},
290.         {'label': 'Deuda Corto Plazo', 'value':
    'ShortTermBorrowings'},
291.         {'label': 'Deuda Largo Plazo', 'value':
    'LongTermDebt'},
292.         {'label': 'Patrimonio Neto', 'value':
    'Equity'},
293.         {'label': 'Resultado Neto', 'value':
    'ProfitLoss'},
294.         {'label': 'Resultado Operativo', 'value':
    'ProfitLossFromOperatingActivities'},
295.         {'label': 'ROA', 'value': 'ROA'},
296.         {'label': 'ROE', 'value': 'ROE'},
297.         {'label': 'Activos Corrientes / Activos
    Totales', 'value': 'currentassets_totalassets_ratio'},
298.         {'label': 'Activos No Corrientes / Activos
    Totales', 'value': 'noncurrentassets_totalassets_ratio'},
299.     ],
300.     value='none', # Valor inicial seleccionado
301.     multi=False# No permitir selecciones múltiples
    para este selector
302. ),
303.     # Etiqueta y menú desplegable para seleccionar el
    tipo de gráfico
304.     html.Label("Seleccione el gráfico:"),
305.     dcc.Dropdown(
306.         id='graph-type-selector', # Identificador para
    el selector de tipo de gráfico
307.         options=[# Opciones de tipo de gráfico (líneas o
    barras)
308.                 {'label': 'Gráfico de líneas', 'value':
    'line'},
309.                 {'label': 'Gráfico de barras', 'value':
    'bar'},
310.            ],
311.         value='line', # Valor inicial seleccionado
312.         multi=False# No permitir selecciones múltiples
    para este selector
313.     ),
314.
315.     ], style={'display': 'inline-block', 'width': '30%',
    'vertical-align': 'top', 'margin-right': '20px'}),
316.
```

```
317.         # Contenedor para la columna de texto, incluyendo
           categorías de empresas
318.         html.Div([
319.             # Sección de texto para "Operadoras" con la lista de
           empresas correspondiente
320.             html.Div([
321.                 html.H4("OPERADORAS:"),
322.                 html.Ul([
323.                     html.Li("Telefónica"),
324.                     html.Li("Orange"),
325.                     html.Li("Vodafone"),
326.                 ]),
327.             ], style={'display': 'inline-block', 'vertical-
           align': 'top', 'margin-right': '20px'}),
328.             # Sección de texto para "Fabricantes" con la lista
           de empresas correspondiente
329.             html.Div([
330.                 html.H4("FABRICANTES:"),
331.                 html.Ul([
332.                     html.Li("Nokia"),
333.                     html.Li("Ericsson"),
334.                 ]),
335.             ], style={'display': 'inline-block', 'vertical-
           align': 'top', 'margin-right': '20px'}),
336.             # Sección de texto para "Prestadores de Servicios"
           con la lista de empresas correspondiente
337.             html.Div([
338.                 html.H4(["PRESTADORES", html.Br(), "DE
           SERVICIOS:"]),
339.                 html.Ul([
340.                     html.Li("Alphabet"),
341.                     html.Li("Microsoft"),
342.                 ]),
343.             ], style={'display': 'inline-block', 'vertical-
           align': 'top'}),
344.             ], style={'display': 'inline-block', 'width': '70%',
           'vertical-align': 'top'}),
345.         ], style={'display': 'flex', 'width': '90%', 'margin': '0
           auto', 'margin-bottom': '20px'}),
346.
347.         # Selección de texto con mi nombre en la esquina inferior
           derecha
348.         html.Div('BLANCA JIMÉNEZ PÉREZ', style={
349.             'position': 'fixed',
350.             'bottom': '10px',
351.             'right': '10px',
352.             'z-index': '1000'
```

```
353.         }),
354.
355.         # Contenedor para el gráfico generado basado en las
           selecciones del usuario
356.         html.Div([
357.             dcc.Graph(id='time-series-plot') # Componente para
           visualizar el gráfico
358.         ], style={'width': '100%', 'display': 'block'}),
359.     ])
360.
361.     # Decorador para definir una función de callback en Dash. Esta
           función se activa con cambios en los inputs especificados.
362.     @app.callback(
363.         Output('time-series-plot', 'figure'), # Define el componente
           y la propiedad que se actualizarán con el callback.
364.         [Input('empresa-selector', 'value'), # Inputs de los menús
           desplegados que activan el callbac
365.          Input('fecha-selector', 'value'),
366.          Input('tag-selector', 'value'),
367.          Input('graph-type-selector', 'value')]
368.     )
369.
370.     # Definición de la función que se llama cuando los inputs del
           callback cambian.
371.     # Recibe las selecciones actuales de los menús desplegados como
           argumentos.
372.     defupdate_plot(selected_empresas, selected_filed_values,
           selected_tag, graph_type):
373.         try:
374.             traces= [] # Inicializar una lista para almacenar los
           datos de los trazos del gráfico.
375.             unique_traces_check=set() # Conjunto para verificar
           trazos únicos y evitar duplicados.
376.
377.             # Iterar sobre cada empresa seleccionada para procesar
           sus datos.
378.             forselected_empresainselected_empresas:
379.                 data_info=None# Inicializar variable para almacenar
           los datos obtenidos de la API.
380.                 index=int(selected_empresa.split('_')[1]) # Extraer
           el índice de la empresa basado en su identificador.
381.
382.                 # Lógica para obtener y procesar datos específicos
           basados en el tipo de cálculo seleccionado (p. ej., ROE, ROA).
383.                 # Se hace uso de funciones para obtener datos de la
           API y se realizan cálculos específicos.
```

```
384.         # Se crea un trazo para el gráfico basado en los
           datos procesados.
385.
386.         # Condición para manejar diferentes tipos de tags y
           calcular ratios específicos como ROE o ROA.
387.         ifselected_tagin ['ROE', 'ROA']: # Se realizan
           cálculos específicos para ROE y ROA usando los datos obtenidos.
388.             data_resultado_operativo=get_sec_data(urls_resul
           tado_operativo[index], urls_resultado_operativo)
389.             data_patrimonio_netto=get_sec_data(urls_patrimoni
           o_netto[index], urls_patrimonio_netto)
390.             data_activos=get_sec_data(urls_empresas_assets[i
           ndex], urls_empresas_assets)
391.
392.         # Calcular ROE o ROA con Resultado Operativo
393.         ifdata_resultado_operativoanddata_patrimonio_net
           oanddata_activos:
394.             ro_data=data_resultado_operativo['data']
395.             pn_data=data_patrimonio_netto['data']
396.             a_data=data_activos['data']
397.
398.             # Unir los DataFrames por la fecha
399.             merged_data=ro_data.merge(pn_data,
           on='filed', suffixes=('_ro', '_pn'))
400.             merged_data=merged_data.merge(a_data.rename(
           columns={'val': 'val_a'}), on='filed')
401.
402.             # Calcular ROE y ROA
403.             ifselected_tag=='ROE':
404.                 merged_data['val']
           =merged_data['val_ro'] /merged_data['val_pn']
405.             else: # ROA
406.                 merged_data['val']
           =merged_data['val_ro'] /merged_data['val_a']
407.
408.             # Asegurarse de que solo se tome el último
           valor de cada año
409.             merged_data['year']
           =merged_data['filed'].dt.year
410.             merged_data=merged_data.sort_values(by='file
           d').drop_duplicates(subset='year', keep='last')
411.
412.             # Filtrar por fechas seleccionadas
413.             if'all'notinselected_filed_values:
414.                 valid_dates_set=set(int(date)
           fordateinselected_filed_values)
```



```
415.         merged_data=merged_data[merged_data['yea
r'].isin(valid_dates_set)]
416.
417.         # Añadir trazos a la gráfica
418.         ifnotmerged_data.empty:
419.             trace_type='line'ifgraph_type=='line'els
e'bar'
420.             trace= {
421.                 'x':
merged_data['filed'].dt.yearifgraph_type=='bar'elsemerged_data['filed'
],
422.                 'y': merged_data['val'],
423.                 'type': trace_type,
424.                 'mode':
'lines+markers'ifgraph_type=='line'elseNone,
425.                 'marker': {'size': 10}
ifgraph_type=='line'else {},
426.                 'name':
data_resultado_operativo["name"]
427.             }
428.             traces.append(trace)
429.
430.         # Condición para calcular ratios de activos y
añadirlos a los trazos del gráfico.
431.         elifselected_tagin
['currentassets_totalassets_ratio',
'noncurrentassets_totalassets_ratio']:
432.             data_total=get_sec_data(urls_empresas_assets[ind
ex], urls_empresas_assets)
433.             data_specific=None
434.             valid_dates_set=set() # Inicializamos
valid_dates_set aquí para garantizar que siempre tenga un valor.
435.
436.             ifselected_tag=='currentassets_totalassets_ratio
':
437.                 data_specific=get_sec_data(urls_empresas_cur
rentassets[index], urls_empresas_currentassets)
438.                 elifselected_tag=='noncurrentassets_totalassets_
ratio':
439.                     data_specific=get_sec_data(urls_empresas_non
currentassets[index], urls_empresas_noncurrentassets)
440.
441.             ifdata_specificanddata_total:
442.                 specific_data=data_specific['data']
443.                 total_data=data_total['data']
444.
445.         # Convertimos las fechas a datetime
```

```
446.         specific_data['filed']
         =pd.to_datetime(specific_data['filed'])
447.         total_data['filed']
         =pd.to_datetime(total_data['filed'])
448.
449.         # Hacemos un merge para calcular el ratio
450.         merged_data=pd.merge(specific_data,
         total_data, on='filed', suffixes=('_specific', '_total'))
451.         merged_data['ratio'] =
         (merged_data['val_specific'] /merged_data['val_total']) *100
452.         merged_data['year']
         =merged_data['filed'].dt.year
453.         merged_data=merged_data.sort_values(by='file
         d').drop_duplicates(subset='year', keep='last')
454.
455.         # Filtrar por fechas seleccionadas
456.         filtered_data=pd.DataFrame() #
         Inicializamos filtered_data como un DataFrame vacío.
457.         if'all'inselected_filed_values:
458.             filtered_data=merged_data
459.         elif'none'notinselected_filed_values:
460.             valid_dates_set=set(int(date)
         fordateinselected_filed_valuesifdate.isdigit())
461.             filtered_data=merged_data[merged_data['y
         ear'].isin(valid_dates_set)]
462.
463.         # Ahora verificamos si filtered_data está
         vacío antes de proceder
464.         ifnotfiltered_data.empty:
465.             # Determinar si se muestra un punto o
         una línea/barra
466.             trace_mode='markers'iflen(valid_dates_se
         t) ==1else'lines+markers'ifgraph_type=='line'elseNone
467.             marker_size=10iflen(valid_dates_set)
         ==1else6
468.             trace_type='line'ifgraph_type=='line'els
         e'bar'
469.             trace= {
470.                 'x': filtered_data['filed']
         ifgraph_type=='line'elsefiltered_data['year'],
471.                 'y': filtered_data['ratio'],
472.                 'type': trace_type,
473.                 'mode': trace_mode,
474.                 'marker': {'size': marker_size},
475.                 'name':
         f"{'AC/AT'ifselected_tag=='currentassets_totalassets_ratio'else'ANC/AT
         '}"
```

```
476.         }
477.         traces.append(trace)
478.
479.         # Esta condición verifica si el tipo de dato
         seleccionado no es uno de los cálculos ya realizados
480.         ifselected_tagnotin ['ROE', 'ROA',
         'currentassets_totalassets_ratio',
         'noncurrentassets_totalassets_ratio']:
481.
482.         # Aquí se seleccionan diferentes conjuntos de
         datos basados en el tag seleccionado por el usuario.
483.         # Cada bloque condicional invoca la función
         `get_sec_data` con URLs específicas para cada tipo de dato.
484.         # Si la información es recuperada exitosamente,
         se procesa y se prepara para la visualización.
485.
486.         ifselected_tag=='assets':
487.             data_info=get_sec_data(urls_empresas_assets[
         index], urls_empresas_assets)
488.         elifselected_tag=='currentassets':
489.             data_info=get_sec_data(urls_empresas_current
         assets[index], urls_empresas_currentassets)
490.         elifselected_tag=='noncurrentassets':
491.             data_info=get_sec_data(urls_empresas_noncurr
         entassets[index], urls_empresas_noncurrentassets)
492.         elifselected_tag=='propertyplantandequipmentgros
         s':
493.             data_info=get_sec_data(urls_nuevas_ppae[inde
         x], urls_nuevas_ppae)
494.         elifselected_tag=='intangibles':
495.             data_info=get_sec_data(urls_intangibles[inde
         x], urls_intangibles)
496.         elifselected_tag=='goodwill':
497.             data_info=get_sec_data(urls_goodwill[index],
         urls_goodwill)
498.         elifselected_tag=='revenue':
499.             data_info=get_sec_data(urls_ingresos[index],
         urls_ingresos)
500.         elifselected_tag=='ShortTermBorrowings':
501.             data_info=get_sec_data(urls_deuda_corto_plazo[
         index], urls_deuda_corto_plazo)
502.         elifselected_tag=='LongTermDebt':
503.             data_info=get_sec_data(urls_deuda_largo_plazo[
         index], urls_deuda_largo_plazo)
504.         elifselected_tag=='Equity':
505.             data_info=get_sec_data(urls_patrimonio_neto[
         index], urls_patrimonio_neto)
```



```
537.         year=unit_data['year'].iloc[
    0]
538.
539.         # Se aplica la tasa de
    conversión correspondiente a los valores de los datos.
540.         conversion_factor=CONVERSION
    _RATES.get(unit_key, {}).get(year, 1.0)
541.
542.         unit_data['val']
    =unit_data['val'] *conversion_factor
543.
544.         # Se decide el tipo de trazo
    (línea o barra) basado en la selección del usuario.
545.         trace_type='line'ifgraph_typ
    e=='line'else'bar'
546.
547.         # Se crea el trazo con los
    datos procesados y se añade a la lista de trazos.
548.         trace= {
549.             'x': unit_data['filed'],
550.             'y': unit_data['val'],
551.             'type': trace_type,
552.             'mode':
    'lines+markers'ifgraph_type=='line'elseNone,
553.             'marker': {'size': 10}
    ifgraph_type=='line'else {},
554.             'name': entity_name,
555.         }
556.         traces.append(trace)
557.
558.         # Modificar el título del gráfico basado en el tipo de
    dato seleccionado para reflejar el contenido visualizado.
559.         tag_label_map={ # Mapeo de tags a etiquetas legibles
    para el usuario.
560.             'assets': 'activos',
561.             'currentassets': 'activos corrientes',
562.             'noncurrentassets': 'activos no corrientes',
563.             'propertyplantandequipmentgross': 'propiedad, planta
    y equipo',
564.             'intangibles': 'intangibles',
565.             'goodwill': 'fondo de comercio',
566.             'revenue': 'ingresos',
567.             'ShortTermBorrowings': 'deuda corto plazo',
568.             'LongTermDebt': 'deuda largo plazo',
569.             'Equity': 'patrimonio neto',
570.             'ProfitLoss': 'resultado neto',
```

```
571.         'ProfitLossFromOperatingActivities': 'resultado
operativo',
572.         'ROA': 'ROA',
573.         'ROE': 'ROE',
574.         'currentassets_totalassets_ratio': 'Activos
Corrientes / Activos Totales',
575.         'noncurrentassets_totalassets_ratio': 'Activos No
Corrientes / Activos Totales'
576.     }
577.     tag_label=tag_label_map.get(selected_tag, "")
578.     figure_title=f"Serie temporal de {tag_label} de las
empresas" if tag_label else "Serie temporal de empresas"
579.
580.     # Configuración común de estilo de fuente para el
gráfico.
581.     common_font_style= {'family': 'Arial, sans-serif',
'size': 12}
582.
583.     # Si el tag seleccionado no es uno de los ratios
específicos, se ajustan los valores de y y se configura el eje y.
584.     if selected_tag not in ['ROE', 'ROA',
'currentassets_totalassets_ratio',
'noncurrentassets_totalassets_ratio']:
585.
586.         # Dividir los valores de y por 1e9 (mil millones) y
ajustar el trazado
587.         for trace in traces:
588.             # Ajuste de los valores de y para representarlos
en mil millones y mejorar la legibilidad del gráfico.
589.             trace['y'] = [y/1e9 for y in trace['y']]
590.
591.         # Configuración adicional del eje y para usar
notación en millones.
592.         yaxis_config= {
593.             'title': 'Valor (mil millones)',
594.             'titlefont': common_font_style,
595.             'tickfont': common_font_style,
596.             'tickformat': '.0f',
597.         }
598.
599.         # Devolución de la configuración completa del
gráfico para ser renderizado por Dash.
600.         return {
601.             'data': traces,
602.             'layout': {
603.                 # Configuraciones del layout, incluyendo
títulos de ejes, estilo de fuente y modo de selección.
```

```
604.         'xaxis': {
605.             'title': 'Fecha',
606.             'titlefont': common_font_style,
607.             'tickfont': common_font_style,
608.         },
609.         'yaxis': yaxis_config,
610.         'title': {
611.             'text': figure_title,
612.             'font': {'family': 'Arial, sans-serif',
        'size': 14},
613.         },
614.         'legend': {
615.             'traceorder': 'reversed',
616.             'font': common_font_style,
617.         },
618.         'font': common_font_style,
619.         'clickmode': 'event+select',
620.     }
621. }
622. else:
623.     # Configuración alternativa del gráfico para los
        casos de cálculo de ratios específicos.
624.     return {
625.         'data': traces,
626.         'layout': {
627.             # Configuraciones del layout específicas
        para ratios.
628.             'xaxis': {
629.                 'title': 'Fecha',
630.                 'titlefont': common_font_style,
631.                 'tickfont': common_font_style,
632.             },
633.             'yaxis': {
634.                 'title': 'Valor
        (%)'if'ratio'inselected_tagelse'Valor',
635.                 'titlefont': common_font_style,
636.                 'tickfont': common_font_style,
637.             },
638.             'title': {
639.                 'text': figure_title,
640.                 'font': {'family': 'Arial, sans-serif',
        'size': 14},
641.             },
642.             'legend': {
643.                 'traceorder': 'reversed',
644.                 'font': common_font_style,
645.             },
```

```
646.         'font': common_font_style,
647.         'clickmode': 'event+select',
648.     }
649. }
650.
651.     except Exception as e: # En caso de error, devolver una figura
        vacía o con configuración básica.
652.         print(f"Error en la actualización de la gráfica:
        {str(e)}")
653.         return {
654.             'data': [],
655.             'layout': {
656.                 'xaxis': {'title': 'Fecha'},
657.                 'yaxis': {'title': 'Valor
        (%)' if 'ratio' in selected_tag else 'Valor'},
658.                 'title': 'Serie Temporal de Empresas',
659.                 'legend': {'traceorder': 'reversed'},
660.                 'clickmode': 'event+select',
661.             }
662.         }
663.
664.     # Comando para ejecutar la aplicación Dash si este script es el
        punto de entrada principal.
665.     if __name__ == '__main__':
666.         app.run_server(debug=True) # Ejecutar el servidor de Dash
        con modo debug activado.
```