



COMILLAS

UNIVERSIDAD PONTIFICIA

ICAI

GRADO EN INGENIERÍA EN TECNOLOGÍAS DE TELECOMUNICACIÓN

TRABAJO FIN DE GRADO INVESTIGACIÓN SOBRE LA ADAPTACIÓN MULTIGEOGRÁFICA DE MSCOPE

Autor: Eva Calvo-Sotelo Piera

Director: Rodrigo Sánchez

Madrid

Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título
INVESTIGACIÓN SOBRE LA ADAPTACIÓN MULTIGEOGRÁFICA DE MSCOPE
en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el
curso académico 2023/24 es de mi autoría, original e inédito y
no ha sido presentado con anterioridad a otros efectos.

El Proyecto no es plagio de otro, ni total ni parcialmente y la información que ha sido
tomada de otros documentos está debidamente referenciada.



Fdo.: Eva Calvo-Sotelo Piera

Fecha: 03/07/2024

Autorizada la entrega del proyecto

EL DIRECTOR DEL PROYECTO

Fdo.: Rodrigo Sánchez

Fecha: 03/07/2024





COMILLAS

UNIVERSIDAD PONTIFICIA

ICAI

GRADO EN INGENIERÍA EN TECNOLOGÍAS DE TELECOMUNICACIÓN

TRABAJO FIN DE GRADO INVESTIGACIÓN SOBRE LA ADAPTACIÓN MULTIGEOGRÁFICA DE MSCOPE

Autor: Eva Calvo-Sotelo Piera

Director: Rodrigo Sánchez

Madrid

Agradecimientos

A mis padres.

INVESTIGACIÓN SOBRE LA ADAPTACIÓN MULTIGEOGRÁFICA DE MSCOPE

Autor: Calvo-Sotelo Piera, Eva.

Director: Sánchez, Rodrigo.

Entidad Colaboradora: mscope.

RESUMEN DEL PROYECTO

El proyecto presente consiste en investigar las consecuencias de la expansión geográfica de la empresa mscope, para la cual llevo trabajando desde octubre de 2023 como ingeniera de software. Se me encomendó realizar un estudio preliminar sobre el impacto que tendría esta expansión geográfica, especialmente en términos de arquitectura de la herramienta y otros aspectos relevantes. Esta expansión comenzó en enero de 2024, con Reino Unido, lo que también me ha permitido reportar los cambios implementados hasta la fecha.

Palabras clave: expansión, escalabilidad, arquitectura, Big Data, microservicios.

1. Introducción

La compañía mscope actualmente distribuye una herramienta SaaS y Business Intelligence que opera exclusivamente en el territorio de España. Con la finalidad de expandir su alcance hacia nuevas regiones es necesario llevar a cabo una adaptación en la arquitectura de esta herramienta para habilitar su funcionamiento en un contexto multi geográfico. Mi proyecto consiste en una investigación profunda de los cambios inherentes a esta expansión.

2. Definición del Proyecto

El proyecto se puede dividir en 5 partes individuales, que, a pesar de estar relacionadas entre sí, trabajaré de forma independiente. Se llevarán a cabo investigaciones en las áreas siguientes:

- i. Datos: evaluación del almacenamiento de datos y su coste. Se explorarán métodos eficientes de almacenamiento y recuperación de datos utilizando tecnologías como Amazon S3 y RDS. Se investigará el origen de los datos y el mapeo necesario a realizar según cual sea este.

- ii. **Arquitectura:** diseño y optimización de la arquitectura del sistema para mejorar la escalabilidad y mantenibilidad. Esto incluye la implementación de una arquitectura modular y la introducción del proyecto One Intel Product (OIP) para separar las funcionalidades públicas y privadas.
- iii. **Backend:** Refactorización del código backend para mejorar su eficiencia y modularidad. Se utilizarán técnicas como la Programación Orientada a Aspectos (AOP) para manejar la lógica transversal y mejorar la gestión de conexiones a bases de datos.
- iv. **Frontend:** Implementación de mejoras en la interfaz de usuario, incluyendo la gestión dinámica de divisas y la optimización de la responsividad de la aplicación. Se abordarán problemas de hardcoding y se desarrollarán soluciones escalables para futuras expansiones.
- v. **Microservicios:** creación de microservicios desacoplados para funcionalidades específicas, permitiendo una mayor flexibilidad y reutilización del código. Se explorará la implementación de microservicios en diferentes tecnologías para optimizar el rendimiento y facilitar la integración en diversas partes del sistema.

3. Descripción de la herramienta

La herramienta que comercializa mscope es una página web donde los clientes pueden consultar aquellos datos financieros que sean de su interés. A continuación, se presenta la pestaña más importante de la aplicación: el Dashboard.

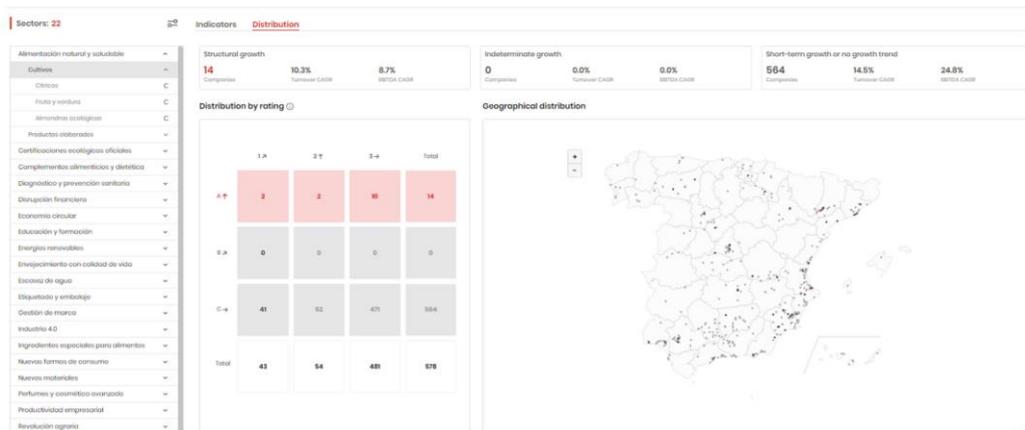


Ilustración 1 – Dashboard de la herramienta

La complejidad de la expansión radica en adaptar la herramienta a varios clientes de diferentes regiones manteniendo una uniformidad coherente. Cada región implica una serie de modificaciones en los cinco niveles mencionados anteriormente: datos, arquitectura, backend, frontend y microservicios.

4. Resultados

Una de las propuestas más importantes es la adopción de una arquitectura Big Data. Esta nueva arquitectura implica numerosos cambios que conllevan mejoras significativas en la gestión y procesamiento de grandes volúmenes de datos. La transición a Big Data no solo aumentaría la eficiencia y la velocidad de los cálculos y análisis, sino que también proporcionaría una base más robusta y escalable para el futuro crecimiento del proyecto.

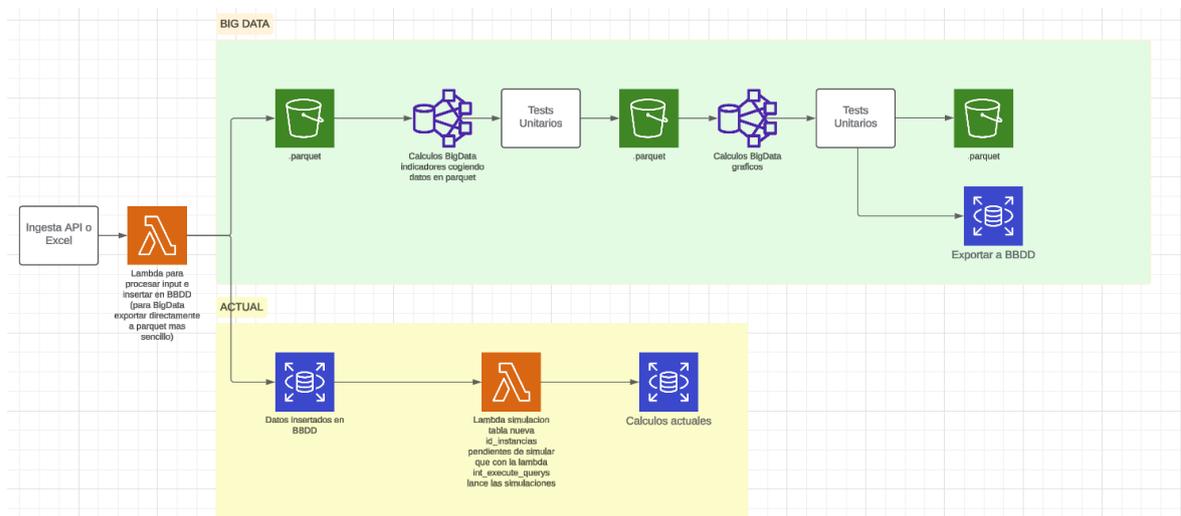


Ilustración 2 – Propuesta de arquitectura Big Data

5. Conclusiones

Este proyecto me ha enseñado que la clave del éxito en el desarrollo de software no reside solo en la implementación técnica, sino también en una planificación estratégica y una visión a largo plazo; la importancia de diseñar el sistema para que sea adaptable y escalable en múltiples dimensiones, en lugar de abordar problemas de manera aislada; y la priorización de la escalabilidad. Estas son lecciones que llevaré conmigo en futuros proyectos. Además, la experiencia de trabajar en un entorno profesional real ha reforzado la importancia de la comunicación, la colaboración y la adaptabilidad en el desarrollo de soluciones tecnológicas efectivas y sostenibles.

En resumen, este proyecto no solo ha sido una oportunidad para aplicar y ampliar mis conocimientos técnicos, sino también una valiosa lección sobre cómo abordar problemas

complejos de manera estratégica y sostenible. Estos aprendizajes me han preparado mejor para enfrentar futuros desafíos y contribuir de manera más efectiva al éxito de proyectos tecnológicos en el futuro.

6. Referencias

- [1] Amazon Web Services. (s.f.). *Amazon S3 Documentation*. Sacado de <https://docs.aws.amazon.com/s3/>
- [2] Amazon Web Services. (s.f.). *Amazon RDS Documentation*. Sacado de <https://aws.amazon.com/es/rds/>
- [3] Cerezo, F. (s.f.). *Lecciones de la asignatura de Datos Masivos / Big Data - Tema 5 - Map Reduce*. Escuela Técnica Superior de Ingeniería (ICAI).
- [4] Cerezo, F., & Vela, B. (s.f.). *Experience of the architectural evolution of a Big Data system*. VorTIC3 Research Group, Universidad Rey Juan Carlos (URJC).
- [5] Baeldung. (s.f.). *A Guide to Caching in Spring*. Sacado de <https://www.baeldung.com/spring-cache-tutorial>
- [6] Aprender Big Data. (s.f.). *Apache Parquet*. Sacado de <https://aprenderbigdata.com/apache-parquet/>
- [7] Arquitectura Java. (s.f.). *Spring AOP y Aspectos*. Sacado de <https://www.arquitecturajava.com/spring-aop-y-aspectos/>
- [8] Cincodías. (2023, 13 de noviembre). *Las pymes españolas aumentan su presencia internacional*. Sacado de https://cincodias.elpais.com/cincodias/2023/11/13/pyme/1699865705_396180.html#

Índice de la memoria

Capítulo 1. Introducción	4
1.1 Contexto de la empresa	4
1.2 contexto de la expansión geográfica de la empresa.....	6
Capítulo 2. Descripción de las Tecnologías.....	8
Capítulo 3. Estado de la Cuestión	10
Capítulo 4. Definición del Trabajo	12
4.1 Justificación.....	12
4.1.1 Fuente de datos de cada región.....	12
4.1.2 Optimización del rendimiento backend	12
4.1.3 Personalización del frontend.....	13
4.1.4 Implementación de microservicios	13
4.2 Objetivos y metodología	13
Capítulo 5. Investigación Desarrollada	14
5.1 Datos.....	14
5.1.1 Almacenamiento de datos y costes	14
5.1.2 Origen de los datos.....	19
5.1.3 Mapeo de los datos de Reino Unido.....	19
5.2 Arquitectura.....	22
5.2.1 Arquitectura actual.....	22
5.2.2 Mejora de la arquitectura con tecnología Big Data	31
5.3 Backend.....	38
5.3.1 One Intel Product.....	38
5.3.2 Estructura de carpetas del backend	41
5.3.3 Código del backend	46
5.3.4 Otras mejoras	47
5.4 Frontend	50
5.5 Microservicios.....	52
Capítulo 6. Análisis de Resultados.....	54

6.1	Resultados en Datos	54
6.2	Resultados en Arquitectura	54
6.3	Resultados en Backend.....	55
6.4	Resultados en Frontend	56
6.5	Resultados en Microservicios.....	56
Capítulo 7. Conclusiones.....		58
Capítulo 8. Bibliografía.....		60
ANEXO I: ALINEACIÓN DEL PROYECTO CON LOS ODS		61

Índice de ilustraciones

Ilustración 1 – Dashboard de la herramienta	9
Ilustración 2 – Propuesta de arquitectura Big Data	10
Ilustración 3. Gráfico de barras de los costes de RDS.....	15
Ilustración 4. Gráfico de barras de los costes de RDS en PRO	16
Ilustración 5. Pie Chart del disk usage del cliente en España.....	17
Ilustración 6. Pie Chart del disk usage del cliente en Reino Unido.....	18
Ilustración 7. Almacenamiento en Amazon S3	18
Ilustración 8. Mapeo de datos de UK	20
Ilustración 9. Arquitectura actual	23
Ilustración 10. Ejemplo simple de como funciona Map Reduce.....	24
Ilustración 11. Flujo de la AWS Step Function - cálculos producto final.....	29
Ilustración 12. Comparación de arquitecturas	32
Ilustración 13. Arquitectura OIP	39
Ilustración 14. Estructura de carpetas del backend.....	42
Ilustración 15. Visual auxiliar sobre la división del proyecto	44
Ilustración 16. Estructura de módulos	45
Ilustración 17. Inspector de web - Timing.....	47
Ilustración 18. Inspector de web - Headers	48
Ilustración 19. Modal multicliente	49

Capítulo 1. INTRODUCCIÓN

En octubre de 2023, comencé a mis prácticas en mscope como desarrolladora de software. En esta empresa, se me brindó la oportunidad de realizar mi proyecto de fin de grado, el cual consiste en investigar las consecuencias, a nivel tecnológico, que supone la expansión geográfica que ha querido llevar a cabo la misma. Antes de empezar con dicho proyecto de investigación, me gustaría hacer una breve introducción de mscope y las razones de su expansión, así como las tecnologías que utilizan.

1.1 CONTEXTO DE LA EMPRESA

Mscope es una startup en etapa temprana que se dedica a aplicar inteligencia artificial (IA) en los campos de la microeconomía y las finanzas con el objetivo de ayudar a los inversores a identificar pequeñas y medianas empresas (pymes) altamente prometedoras para así acelerar su crecimiento. La misión de mscope es transformar el panorama de las inversiones mediante la aplicación de tecnología avanzada y análisis de datos detallados.

El equipo de mscope está compuesto por profesionales altamente cualificados con títulos de doctorado (PhD), MBA y antecedentes en universidades y empresas de primer nivel. Este equipo multidisciplinario abarca áreas como producto (*Product*), datos (*Data*) y tecnología (*Tech*), y está compuesto por 20 personas provenientes de ocho países diferentes distribuidos en cuatro continentes. La diversidad y el alto nivel de especialización del equipo permiten a mscope desarrollar soluciones de alta calidad que tienen un impacto significativo tanto en la economía como en la sociedad.

Inteligencia de mercado granular

Una de las principales fortalezas de mscope es su capacidad para elaborar inteligencia de mercado extremadamente granular. Esto se logra mediante la identificación, clasificación y calificación de pymes con alto potencial de crecimiento, utilizando una metodología

sofisticada y propietaria. Esta metodología permite a los inversores institucionales transformar la generación de oportunidades de inversión en un proceso radicalmente enfocado, exhaustivo y eficiente.

Soluciones modulares y automatización

Además de su enfoque en la inteligencia de mercado, mscope también se dedica a construir soluciones altamente modulares que permiten la digitalización y automatización de operaciones. Estas soluciones permiten a los inversores institucionales transformar la gestión de carteras en un proceso increíblemente intuitivo, colaborativo y transparente. La digitalización y automatización son esenciales para mejorar la eficiencia y la efectividad en la administración de inversiones, lo que resulta en una mejor toma de decisiones y una gestión más ágil de las carteras.

Definición y segmentación del mercado

Según una entrevista con el CEO de mscope, el producto principal de la empresa es la inteligencia de mercado dirigida a un sector muy específico: los inversores institucionales y el sector bancario. El producto se basa en datos recogidos de fuentes viables y fiables que no se alteran, sino que se categorizan y clasifican cuidadosamente. Los clientes de mscope son inversores institucionales, es decir, profesionales dedicados a la inversión, tales como fondos de equity, fondos de deuda, asesores financieros, bancos e incluso administraciones públicas.

mscope no se limita a proporcionar datos sin procesar; la empresa también segmenta estos datos de manera que agrega un valor significativo. A grandes rasgos, mscope se dedica a identificar, clasificar y calificar compañías. Uno de los aspectos más interesantes de su enfoque es la clasificación, ya que mscope es capaz de identificar lo que denomina microtendencias. Estas microtendencias son conjuntos o subconjuntos tan específicos que permiten a los inversores enfocarse en áreas muy concretas y detalladas.

Por ejemplo, si dijera "invierte en la industria", uno podría responder con escepticismo debido a la amplitud del término. Pero si específico "invierte en la industria automovilística"

y aún más detallado, "invierte en proveedores de coches de Renault y BMW", la información se vuelve mucho más útil. Pues mscope va incluso más allá y dice, "invierte en bioestimulantes", una microtendencia que contiene 27 empresas específicas. Esta capacidad para identificar microtendencias permite a los inversores no solo financiar o invertir en estas empresas, sino también segmentar sus estrategias comerciales.

Estrategias comerciales y de crecimiento

Las empresas pueden utilizar las microtendencias identificadas por mscope para definir a qué clientes dirigirse, qué clientes digitalizar, en qué clientes pueden colocar sus productos o a quién quieren vender. En este caso, la parte de clasificación resulta más relevante que la de calificación, ya que ayuda a diseñar estrategias comerciales o de crecimiento más precisas. Esto demuestra que, aunque mscope actualmente se enfoca en inversores institucionales, su alcance y potencial impacto pueden ser aún mayores.

En resumen, mscope se distingue no solo por su enfoque innovador y su equipo talentoso, sino también por su compromiso con la creación de herramientas que faciliten el crecimiento de las pymes y mejoren la eficiencia de los procesos de inversión institucional. Su uso de tecnologías avanzadas y su metodología propietaria son testimonio de su capacidad para liderar en el campo de la inteligencia de mercado y la gestión de inversiones. Con su habilidad para identificar microtendencias y proporcionar datos segmentados y categorizados, mscope no solo ayuda a los inversores a tomar decisiones más informadas, sino que también ofrece valor a las estrategias comerciales y de crecimiento de sus clientes.

1.2 CONTEXTO DE LA EXPANSIÓN GEOGRÁFICA DE LA EMPRESA

En enero de 2024, se adquiere un nuevo cliente en el Reino Unido, lo que marca el primer paso en la expansión internacional que la empresa aspira a lograr. Este nuevo cliente no solo representa una oportunidad de crecimiento, sino que también sirve como un catalizador para replantear y rediseñar la herramienta en su totalidad. El objetivo es asegurar que la herramienta esté preparada para una futura expansión a nuevas regiones geográficas.

En lugar de simplemente adaptar la herramienta para el cliente del Reino Unido, la estrategia consiste en diseñar un sistema multipaís desde el principio. Esto significa crear una infraestructura y un conjunto de funcionalidades que puedan manejar la incorporación de múltiples geografías sin necesidad de rediseñar cada vez que se añade un nuevo país.

Este enfoque no solo mejora la eficiencia en el desarrollo y despliegue, sino que también asegura que la herramienta sea escalable y adaptable a diversas necesidades regionales.

En resumen, la adquisición del cliente en el Reino Unido no solo impulsa la expansión inmediata, sino que también proporciona una oportunidad valiosa para rediseñar la herramienta con una visión a largo plazo, asegurando que esté preparada para soportar y facilitar el crecimiento futuro de la empresa en nuevas regiones geográficas.

Capítulo 2. DESCRIPCIÓN DE LAS TECNOLOGÍAS

Mscope ha desarrollado sus productos SaaS utilizando tecnología de punta. Estos productos incluyen servicios de Amazon Web Services (AWS), EMR Cluster para el manejo de Big Data, explotación de modelos de inteligencia artificial mediante procesos de Pyspark y entornos web que utilizan servidores Spring Boot con Java 11. Las interfaces de usuario están desarrolladas con React y Redux, lo que asegura una experiencia de usuario fluida y eficiente. La combinación de estas tecnologías avanzadas permite a mscope ofrecer soluciones robustas y escalables que satisfacen las necesidades de sus clientes institucionales.

- Servicios de Amazon Web Services (AWS)

AWS proporciona una infraestructura en la nube que sea segura, escalable y de alta disponibilidad. La empresa utiliza estos servicios para alojar y ejecutar sus aplicaciones. Los servicios de AWS permiten a la empresa escalar sus recursos de manera flexible según la demanda, asegurando que las aplicaciones se mantengan rápidas y fiables.

- EMR Cluster para Big Data

El uso de Amazon EMR (Elastic MapReduce) permite a mscope procesar grandes volúmenes de datos de manera eficiente. Este servicio facilita el análisis y la gestión de Big Data mediante el uso de marcos populares como Apache Hadoop, Apache Spark y Presto. Con EMR, mscope puede ejecutar tareas de análisis de datos complejas en escalas masivas, permitiendo una inteligencia de mercado precisa y detallada.

- Pyspark para Modelos de IA

Pyspark es la interfaz de Python para Apache Spark, y mscope lo utiliza para la explotación de modelos de inteligencia artificial. Esta combinación permite a los desarrolladores de mscope crear y desplegar modelos de IA de manera rápida y eficiente,

aprovechando las capacidades de procesamiento distribuido de Spark para manejar grandes conjuntos de datos.

- Entornos Web con Spring Boot y Java 11

Los entornos web de mscope están contruidos utilizando Spring Boot con Java 11. Spring Boot facilita el desarrollo de aplicaciones basadas en Java de manera rápida, proporcionando una configuración mínima y un conjunto robusto de herramientas para crear aplicaciones listas para producción. Java 11, con su estabilidad y características mejoradas, asegura que las aplicaciones sean seguras, eficientes y fáciles de mantener. La base de datos está gestionada con SQLServer, una potente solución de Microsoft que ofrece robustez, escalabilidad y un excelente rendimiento.

- Interfaces de Usuario con React y Redux

Las interfaces de usuario de mscope están desarrolladas con React y Redux. React es una biblioteca de JavaScript para construir interfaces de usuario interactivas y dinámicas, mientras que Redux es una herramienta para manejar el estado de la aplicación de manera predecible. Juntas, estas tecnologías aseguran que las interfaces sean rápidas, responsivas y fáciles de usar, ofreciendo a los usuarios una experiencia fluida y eficiente.

- SQLServer

SQLServer permite manejar grandes volúmenes de datos con rapidez y fiabilidad, además de ofrecer avanzadas capacidades de análisis y seguridad, lo que es crucial para satisfacer las necesidades de los clientes institucionales de mscope. La combinación de estas tecnologías avanzadas permite a mscope ofrecer soluciones robustas y escalables que satisfacen las necesidades de sus clientes institucionales.

La combinación de estas tecnologías avanzadas permite a mscope ofrecer soluciones robustas y escalables que satisfacen las necesidades de sus clientes institucionales.

Capítulo 3. ESTADO DE LA CUESTIÓN

Al abordar el proyecto de investigación sobre las consecuencias de la expansión geográfica de mscope, es crucial primero revisar las soluciones y trabajos existentes en el mercado que pueden ofrecer una perspectiva comparativa. Este capítulo se utilizará para analizar las iniciativas y herramientas similares, evaluando su relevancia y contribuciones, para proporcionar un contexto sólido que justifique la necesidad del presente proyecto.

El concepto de expansión no debería ser sorprendente para nadie, ya que existen varias empresas tecnológicas mundialmente conocidas como Google, Amazon y Microsoft, que en su día también tuvieron que realizar expansiones geográficas significativas para llegar a donde están hoy. Estas expansiones se caracterizan por la implementación de infraestructuras y servicios adaptados a mercados específicos. Sus estrategias incluyen ajustes en la arquitectura de sistemas, integración de datos locales y personalización de interfaces de usuario. Por ejemplo, Amazon Web Services (AWS) ha desarrollado centros de datos en diferentes regiones, optimizando la latencia y cumpliendo con las regulaciones locales. Estos casos ofrecen valiosos aprendizajes sobre la adaptación tecnológica necesaria para una expansión internacional efectiva.

Otra referencia para mscope, esta en el sector financiero, es Bloomberg. Esta destaca por su capacidad para ofrecer datos y análisis profundos de mercados financieros globales. Al igual que Bloomberg, mscope proporciona inteligencia de mercado, pero se especializa en pymes (Pequeñas y Medianas Empresas). Además, mscope tiene un valor agregado que son las microtendencias. Esto son conjuntos o subconjuntos de datos extremadamente específicos que permiten una inversión más precisa y segmentada del mercado.

De estos ejemplos podemos concluir, que donde hay que centrarse es en la adaptación de modelos de datos, la gestión de infraestructuras backend y la personalización de frontend para cumplir con las especificidades culturales y regulatorias de cada región.

Otra parte importante donde hay que poner el foco, es en la adopción de microservicios. Esto ha permitido a muchas empresas escalar sus aplicaciones de manera eficiente durante la expansión geográfica. Frameworks como Spring Boot facilitan la implementación de arquitecturas resilientes y escalables. Además, investigaciones sobre la arquitectura de microservicios destacan la importancia de la tolerancia a fallos, la baja latencia y la gestión eficiente de recursos, aspectos cruciales para soportar el crecimiento y la demanda en nuevas regiones.

El proyecto asignado en mscope se centra en estudiar el impacto de la expansión geográfica en términos de arquitectura y otros aspectos técnicos sobre la herramienta que la empresa distribuye. A pesar de que existen soluciones y estudios previos sobre la expansión geográfica de plataformas tecnológicas, el enfoque específico de mscope, presenta desafíos y necesidades únicos de la empresa que no han sido específicamente abordados por las soluciones existentes.

Capítulo 4. DEFINICIÓN DEL TRABAJO

4.1 JUSTIFICACIÓN

Este apartado tiene como objetivo destacar el valor del proyecto, mostrando de manera convincente su relevancia y atractivo para clientes, emprendedores e inversores. La justificación debe demostrar cómo este proyecto aborda problemas específicos y ofrece soluciones innovadoras que se diferencian de las opciones existentes en el mercado, subrayando su potencial para satisfacer las necesidades actuales y futuras del sector.

4.1.1 FUENTE DE DATOS DE CADA REGIÓN

Dado que la aplicación de mscope trabaja con indicadores financieros que pueden variar según el país, se requiere una investigación detallada sobre cómo realizar mapeos y procesos específicos para cada región. Este aspecto no está suficientemente cubierto por las soluciones genéricas de mercado, que suelen ofrecer enfoques más amplios y menos personalizados. Adaptar el formato de la fuente de datos de cada región garantiza que los análisis económicos sean precisos y relevantes, permitiendo a los clientes tomar decisiones informadas basadas en datos locales específicos.

4.1.2 OPTIMIZACIÓN DEL RENDIMIENTO BACKEND

La expansión geográfica conlleva un aumento en el volumen de solicitudes y la necesidad de gestionar eficientemente la memoria caché y el soporte multicliente. Evaluar y optimizar estos aspectos es crucial para asegurar un rendimiento robusto y escalable de la herramienta. Un backend optimizado no solo mejora la velocidad y la capacidad de respuesta de la aplicación, sino que también garantiza que los servicios sean fiables y puedan manejar el crecimiento en la carga de trabajo sin degradar la calidad del servicio.

4.1.3 PERSONALIZACIÓN DEL FRONTEND

Cada cliente de mscope tiene especificaciones visuales únicas que deben ser adaptadas a sus necesidades regionales. Este nivel de personalización requiere un enfoque detallado y específico que no siempre está contemplado en las soluciones de localización estándar disponibles en el mercado. Personalizar el frontend según las especificaciones de cada cliente mejora la experiencia del usuario, asegurando que la interfaz sea intuitiva y que los datos presentados sean fácilmente comprensibles y relevantes para los usuarios finales.

4.1.4 IMPLEMENTACIÓN DE MICROSERVICIOS

La utilización de microservicios y la necesidad de garantizar una arquitectura escalable, capaz de manejar la latencia y la tolerancia a fallos, son aspectos fundamentales para el éxito de la expansión. Este proyecto explorará cómo estos componentes pueden ser optimizados para soportar el crecimiento eficiente de mscope en nuevas regiones. Los microservicios permiten una mayor flexibilidad y modularidad, facilitando el mantenimiento y la actualización de la aplicación sin interrupciones significativas en el servicio.

4.2 OBJETIVOS Y METODOLOGÍA

El objetivo principal del proyecto es investigar y evaluar las opciones disponibles en cada uno de los cinco niveles críticos afectados por la expansión geográfica: datos, arquitectura, backend, frontend y microservicios. La meta es identificar y desarrollar las mejores soluciones para cada área, asegurando que la herramienta mscope esté preparada para adaptarse y crecer en nuevas regiones geográficas.

Para alcanzar estos objetivos, se seguirá un enfoque colaborativo y basado en la investigación. Se trabajará estrechamente con los responsables de cada departamento, combinando su experiencia y conocimientos con la información recopilada durante el proceso de investigación. Este enfoque permitirá identificar las soluciones más adecuadas y efectivas para cada uno de los cinco niveles, asegurando una implementación exitosa y sostenible que soporte la expansión futura de la herramienta.

Capítulo 5. INVESTIGACIÓN DESARROLLADA

En este capítulo, se describirá detalladamente el proyecto, enfocándose en las áreas más afectadas por la expansión a nuevas regiones geográficas. Cada subapartado del capítulo representa una de estas áreas. En cada uno se explorará los desafíos enfrentados, las soluciones implementadas y las mejoras logradas en cada área.

5.1 DATOS

En este apartado se investigarán los efectos que se espera que cause la expansión geográfica sobre los datos y la base de datos de la herramienta. Además, gracias al primer cliente en Reino Unido, también se podrán reportar algunos de los primeros pasos que ya se han tomado durante la primera expansión.

5.1.1 ALMACENAMIENTO DE DATOS Y COSTES

Para comenzar, se hará un análisis detallado los sistemas de almacenamiento y los costes asociados a estos que se tienen hasta la fecha. Mscope utiliza principalmente dos sistemas de almacenamiento para gestionar sus datos:

- Amazon S3: este servicio de Amazon Web Services (AWS) proporciona almacenamiento de objetos en la nube. Es altamente escalable y se utiliza para almacenar grandes volúmenes de datos de manera segura y eficiente. Los datos almacenados en S3 pueden incluir desde copias de seguridad hasta datos utilizados activamente en aplicaciones y análisis.
- Amazon RDS (Relational Database Service): este servicio gestionado de base de datos permite configurar, operar y escalar bases de datos relacionales en la nube. RDS es utilizado para almacenar datos estructurados y relacionales, cruciales para el funcionamiento de las aplicaciones de mscope.

Además de estos sistemas, mscope cuenta con un almacenamiento dedicado a la recuperación ante desastres, conocido como "disaster recovery". Este sistema es vital para garantizar que los datos puedan recuperarse en caso de una catástrofe, asegurando la continuidad del negocio y la integridad de la información. Aunque es de gran importancia, este no será relevante en nuestra investigación.

Para este análisis, también es importante conocer que la empresa opera sobre tres entornos de trabajo distintos: Dev (Desarrollo); Pre (Preproducción); y Pro (Producción).

Estos entornos permiten gestionar y desplegar aplicaciones de manera controlada, asegurando que cada cambio se prueba y valida antes de implementarse en producción. Dentro de estos entornos, mscope mantiene bases de datos específicas para sus clientes en España y en el Reino Unido (UK).

Conocida esta información, podemos pasar a analizar el impacto que supuso la entrada de Reino Unido como cliente a principios del 2024. Es evidente, que este nuevo cliente, al suponer una nueva base datos entera, ha tenido un impacto significativo en los costes del almacenamiento RDS. Este crecimiento se refleja en los gráficos proporcionados a continuación:

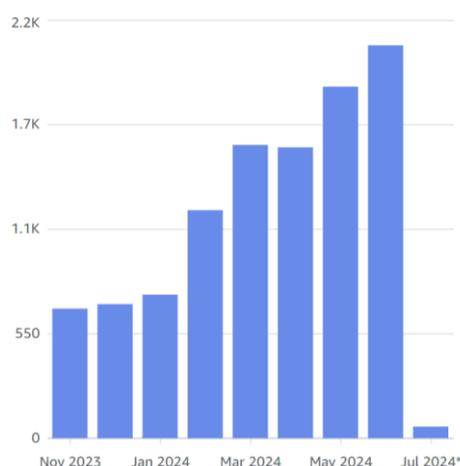


Ilustración 3. Gráfico de barras de los costes de RDS

Este primer gráfico muestra los costes totales de RDS desde noviembre de 2023 hasta julio de 2024. Se observa un incremento notable en los costes a partir de enero de 2024, coincidiendo con la entrada en operación de la base de datos del Reino Unido. Este incremento puede atribuirse a la necesidad de ampliar la capacidad de las máquinas para gestionar la nueva base de datos, lo que se traduce en un aumento del coste. En términos proporcionales, este crecimiento representa un 244% en costes adicionales.

De los tres entornos de trabajo, el de mayor coste es el de producción, como era de esperar. En el siguiente gráfico de barras, podemos observar los costes de RDS en tan solo dicho entorno:

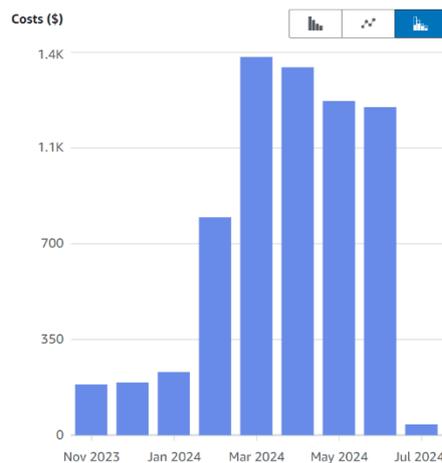


Ilustración 4. Gráfico de barras de los costes de RDS en PRO

La tendencia de los costes sigue una curva ascendente similar, destacando que el entorno de producción es el que más contribuye al incremento de los costes globales, con una participación de más del 60%.

Podemos concluir por tanto que la introducción de la base de datos del Reino Unido ha exigido una mayor capacidad de procesamiento y almacenamiento, incrementando los costes operativos en el entorno de producción de manera significativa. Realizar este análisis es crucial para entender las implicaciones financieras de la expansión y para planificar futuras estrategias de crecimiento y optimización de recursos.

Para entender el coste de almacenamiento, es crucial conocer la cantidad de espacio que estamos utilizando en gigabytes (GB). Esta información la he obtenido a través de la aplicación Microsoft SQL Server Management Studio, la cual permite visualizar el "Disk Usage" de cada base de datos.

Análisis de la base de datos de España (ESP)

En la base de datos de ESP, el total del espacio reservado es de 354,82 GB. Este espacio se distribuye de la siguiente manera:

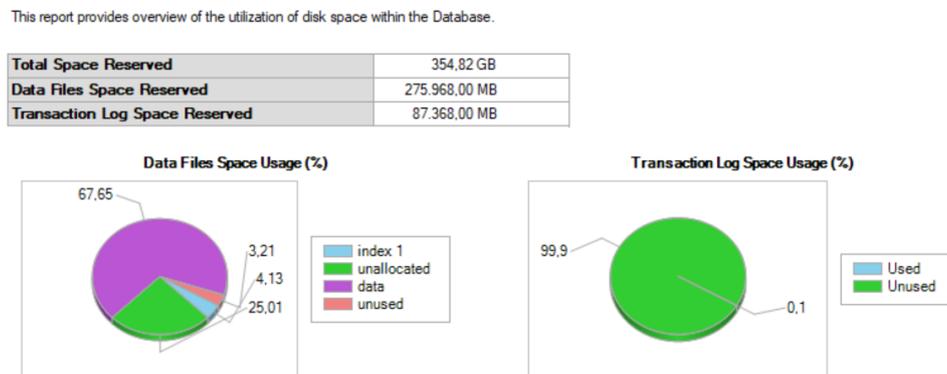


Ilustración 5. Pie Chart del disk usage del cliente en España

Nos interesa fijarnos en la distribución del gráfico de tarta en la esquina izquierda inferior de la *Ilustración 5*. En este se detallan las proporciones específicas de cada categoría de uso del espacio.

- Datos: 67,65% del espacio está dedicado a los datos almacenados.
- Índices: 25% del espacio se utiliza para los índices de las tablas.
- Datos no asignados: 4,13% del espacio corresponde a datos no asignados (unallocated).
- Datos sin usar: 3,21% del espacio está sin usar (unused).

Análisis de la base de datos del Reino Unido (UK)

En la base de datos de UK, el total del espacio reservado es de 134,78 GB. La distribución del espacio es la siguiente:

This report provides overview of the utilization of disk space within the Database.

Total Space Reserved	134,78 GB
Data Files Space Reserved	82.416,19 MB
Transaction Log Space Reserved	55.599,94 MB



Ilustración 6. Pie Chart del disk usage del cliente en Reino Unido

De nuevo, nos interesa fijarnos en el mismo la distribución del mismo gráfico de tarta:

- Datos: 74,14% del espacio está dedicado a los datos almacenados.
- Datos no asignados: 24,84% del espacio se destina a datos no asignados (unallocated).
- Índices: 0,55% del espacio se utiliza para los índices.
- Datos sin usar: 0,47% del espacio está sin usar (unused).



Ilustración 7. Almacenamiento en Amazon S3

Además está el almacenamiento en Amazon S3, el cual ocupa 6,3 TB. Este espacio se utiliza para almacenar diversos "buckets", que contienen datos esenciales para la operación y expansión de la empresa. La *Ilustración 7* muestra un resumen de este almacenamiento, incluyendo la cantidad de objetos almacenados (122,2 millones) y el tamaño promedio de los objetos (54,9 KB).

5.1.2 ORIGEN DE LOS DATOS

Otro factor de gran importancia es la fuente de origen de nuestros datos, ya que esta afectará en la forma de que estos tienes, y por tanto, en la manera en la que los tendremos que trabajar. Actualmente mscope trabaja con una fuente diferente para cada país.

Para los datos de España, se utiliza Orbis, una herramienta esencial para el análisis de información financiera y la captación de clientes a nivel internacional. Orbis es una base de datos que proporciona información detallada sobre millones de empresas en todo el mundo, con un enfoque particular en empresas europeas. Esta se caracteriza por incluir indicadores de solidez financiera, por permitir el geoposicionamiento de estas en mapas, facilitar la creación de análisis financieros detallados y gráficos para ilustrar las finanzas, entre otros.

Para los datos del Reino Unido, mscope utiliza CreditSafe, una herramienta de crédito empresarial que ofrece informes detallados sobre la solvencia y el riesgo de empresas en el Reino Unido y en otros mercados. CreditSafe proporciona información crucial para evaluar la salud financiera de los clientes y proveedores, facilitando decisiones informadas y estratégicas para la expansión y operación de mscope en el Reino Unido.

Para garantizar la uniformidad de los datos entre las diferentes regiones en las que opera mscope, se ha implementado un procedimiento automatizado y eficiente para el mapeo de datos. Este proceso se utiliza específicamente para los datos del Reino Unido, con el objetivo de que tengan la misma estructura y formato que los datos de España, que fue el primer formato que se implementó.

5.1.3 MAPEO DE LOS DATOS DE REINO UNIDO

El flujo del procedimiento de mapeo de datos para el Reino Unido se visualiza en la imagen proporcionada y se detalla a continuación:

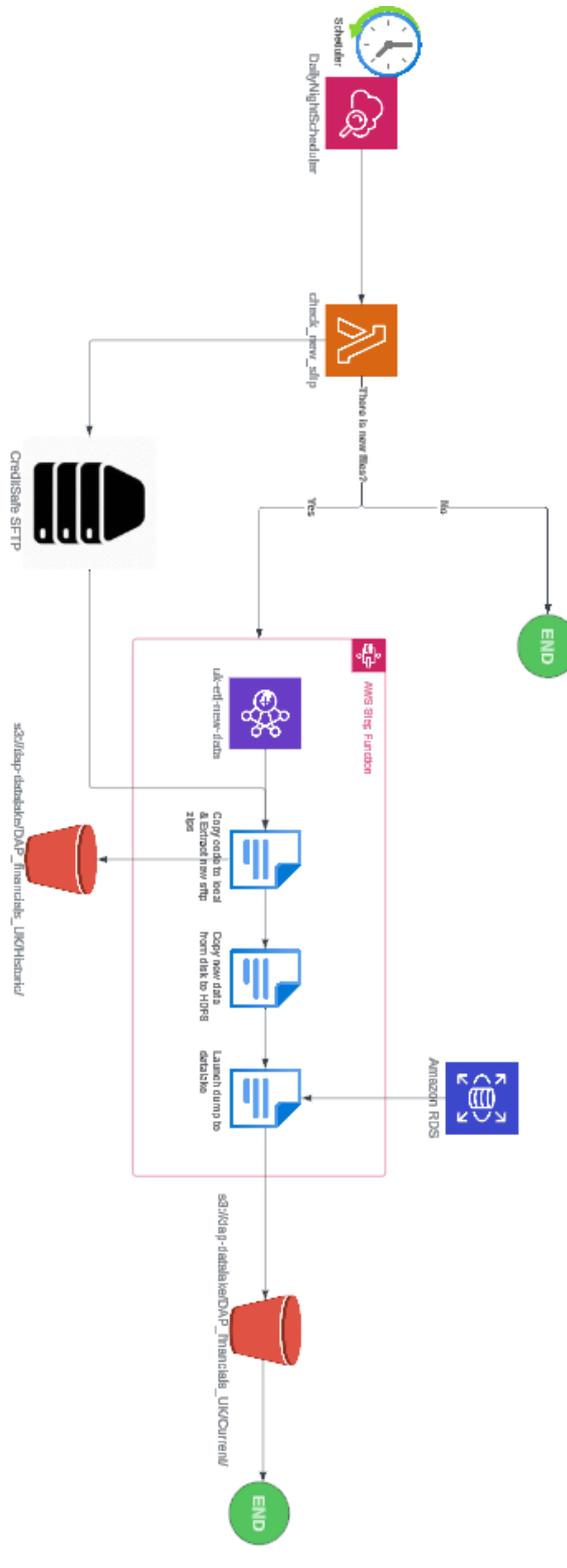


Ilustración 8. Mapeo de datos de UK

Explicación del flujo de izquierda a derecha:

1. Scheduler

- Descripción: un programador (Scheduler) que se ejecuta una vez al día.
- Función: invoca la función Lambda que comprueba si hay nuevos datos disponibles en el SFTP de CreditSafe.

2. Lambda Function

- Descripción: función Lambda en AWS.
- Función: verifica en el SFTP de CreditSafe si hay nuevos datos disponibles.
- Condición:
 - Si no hay nuevos datos: El proceso termina (END).
 - Si hay nuevos datos: Se ejecuta la Step Function en AWS (3).

3. Step Function:

- Descripción: un flujo de trabajo en AWS Step Functions.
- Función: levanta un clúster EMR (Elastic MapReduce) que ejecuta varios pasos para procesar los nuevos datos.
 - Copia el código local y extrae los nuevos archivos zip del SFTP.
 - Copia los nuevos datos del disco al sistema de archivos HDFS.
 - Realiza un volcado de los datos procesados a la base de datos.

4. Transformación y Almacenamiento:

- Descripción: los datos se transforman para obtener el mismo formato de Excel que se utiliza para los datos de España (provenientes de Orbis).
- Función: estas transformaciones aseguran que los datos tengan uniformidad, independientemente de su fuente original.
- Almacenamiento: los archivos Excel resultantes se almacenan en Amazon S3.

Gracias a estas transformaciones específicas, mscope puede mantener una uniformidad en la estructura y formato de sus datos independientemente de su origen. Este proceso, conocido como "mapeo de datos", es crucial para la integración de los datos del Reino Unido con los datos de España. Al tener una estructura uniforme, se facilita el análisis comparativo y la integración de datos en los sistemas de información de mscope, permitiendo una mejor toma de decisiones y una gestión más eficiente.

5.2 ARQUITECTURA

Considerando los puntos discutidos anteriormente, resulta evidente que a nivel de arquitectura es necesario replantear y rediseñar nuestra infraestructura para abordar los problemas actuales y mejorar el rendimiento y la escalabilidad del sistema.

5.2.1 ARQUITECTURA ACTUAL

Hasta ahora solo se ha visto cómo se obtienen los excels, pero nuestro objetivo final es trabajar con tablas de bases de datos. El flujo de procesamiento actual de los datos de mscope se basa en la transformación de archivos Excel provenientes de diferentes regiones (Reino Unido y España) y su integración en bases de datos relacionales. A continuación, se detalla el proceso paso a paso, basado en la ilustración proporcionada:

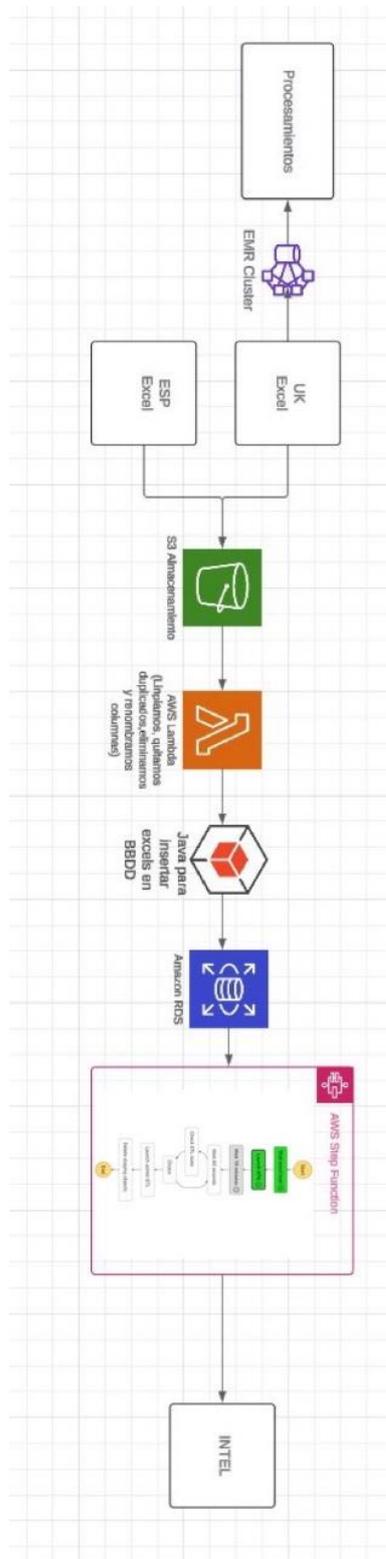


Ilustración 9. Arquitectura actual

El flujo comienza con los datos de UK y España en formato Excel, los cuales se almacenan en Amazon S3, el servicio de almacenamiento en la nube de AWS. Se puede observar en la Ilustración 9, que previo al Excel de Reino Unido, están todos los procesamientos que se han explicado en el apartado anterior. Estos son donde se toman los datos crudos de la fuente de origen y se procesan, mediante un cluster EMR (Elastic MapReduce), para generar un Excel con la misma estructura que los datos de España.

Amazon EMR (Elastic MapReduce) es una de las tecnologías descritas que utiliza mscope. Es un servicio web que facilita el procesamiento y análisis de grandes volúmenes de datos. Con EMR, los usuarios pueden procesar grandes cantidades de datos de manera eficiente y económica al aprovisionar clústeres escalables en la nube de AWS.

Este servicio se basa en el paradigma MapReduce, un modelo de programación diseñado para procesar y generar grandes conjuntos de datos mediante un modelo de pares clave-valor. Este paradigma se compone de dos etapas principales: Map y Reduce, con una etapa intermedia de Shuffle. A continuación, se explica el funcionamiento de este modelo utilizando un ejemplo (mostrado en la *Ilustración 10*) en el que el objetivo es sumar el número total de elementos agrupándolos por los valores "rojos" y "azules". Este proceso es equivalente a una consulta de agrupación de datos en una base de datos.

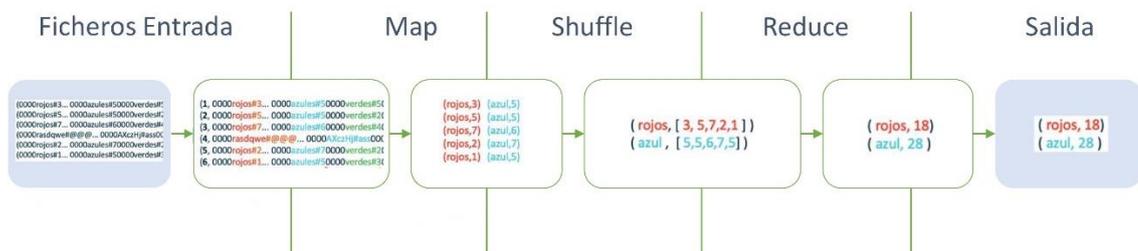


Ilustración 10. Ejemplo simple de como funciona Map Reduce

En la etapa Map, el objetivo es extraer solo los registros relevantes que contienen los valores "rojos" y "azules", junto con sus contadores asociados. En la entrada se reciben varios ficheros que contienen registros con los valores "rojos" y "azules". En el proces map cada registro es leído y procesado para extraer el valor (rojos o azules) y su contador asociado. Por ejemplo, de un fichero con múltiples registros, se extraen pares clave-valor como (rojos, 3), (azules, 5), etc.

En la etapa Shuffle, los pares clave-valor generados por la etapa Map se reorganizan y se envían a diferentes hilos de procesamiento según su clave. Este proceso asegura que todos los registros con la misma clave se envíen al mismo hilo de procesamiento. En este ejemplo específico todos los pares con la clave "rojos" se envían al mismo hilo de procesamiento y todos los pares con la clave "azules" se envían a otro hilo de procesamiento diferente.

En la etapa Reduce, los datos se agrupan y se procesan para sumar los contadores asociados a cada valor. Este paso finaliza el procesamiento de los datos, produciendo la salida final. Para cada grupo de pares clave-valor recibidos, se suman los contadores. Por ejemplo, todos los pares (rojos, [3, 5, 7, 2, 1]) se agrupan y se suman para obtener (rojos, 18). De manera similar, los pares (azules, [5, 5, 6, 7, 5]) se agrupan y se suman para obtener (azules, 28).

El resultado final del procesamiento MapReduce es un conjunto de pares clave-valor donde cada clave tiene asociado el total sumado de sus contadores. En nuestro ejemplo, la salida final es: (rojos, 18); (azules, 28).

Una de las principales ventajas del modelo MapReduce es su capacidad de escalar horizontalmente. Si se dispone de más ficheros de entrada, se pueden añadir más tareas Map que se ejecutan en paralelo, y las tareas Reduce se pueden realizar en varios pasos consecutivos si es necesario. Este esquema de procesamiento paralelo permite manejar grandes volúmenes de datos de manera eficiente y escalable, aprovechando la capacidad de procesamiento distribuido.

Habiendo entendido el ejemplo, podemos ahora definir cinco características clave que explican por qué este servicio es tan ampliamente utilizado.

Características

1. Escalabilidad: EMR permite la creación de clústeres escalables que pueden ajustarse automáticamente en tamaño para adaptarse a la carga de trabajo, proporcionando el rendimiento necesario sin sobredimensionar recursos.
2. Pago por uso: este modelo de pago permite a los usuarios pagar solo por los recursos que consumen, reduciendo significativamente los costos en comparación con la gestión de infraestructura propia.
3. Integración con AWS: EMR se integra perfectamente con otros servicios de AWS, como S3, facilitando el almacenamiento, procesamiento y análisis de datos de manera integrada.
4. Flexibilidad de herramientas: Soporta una amplia variedad de herramientas y frameworks de procesamiento de datos, lo que permite a los usuarios elegir la mejor tecnología para su caso de uso específico.
5. EMR gestiona automáticamente muchas de las tareas complejas asociadas con el despliegue, configuración y ajuste de clústeres de procesamiento de datos, simplificando significativamente la administración de grandes entornos de datos.

En el contexto de nuestro proyecto, estos clústeres EMR se utilizan para procesar datos crudos provenientes de su fuente de origen y transformarlos en un formato uniforme, generando archivos Excel con la misma estructura que los datos de España. Este procesamiento asegura la consistencia y calidad de la información antes de su posterior almacenamiento y análisis.

Una vez obtenidos y almacenados ambos documentos Excel, se realiza una limpieza y preparación de los datos a través de una AWS Lambda.

AWS Lambda es un servicio informático proporcionado por Amazon Web Services (AWS) que permite ejecutar código sin necesidad de aprovisionar o gestionar servidores. Con AWS Lambda, puedes ejecutar tu código en respuesta a eventos específicos, como cambios en los datos, solicitudes HTTP, actividades de streaming, entre otros.

Características

1. **Serverless:** AWS Lambda es un servicio completamente administrado, lo que significa que AWS se encarga de la infraestructura subyacente, permitiendo a los desarrolladores centrarse únicamente en el código.
2. **Escalabilidad:** Lambda escala automáticamente en respuesta al volumen de eventos entrantes. Puede manejar desde unas pocas solicitudes por día hasta miles por segundo.
3. **Pago por uso:** solo se paga por el tiempo de cómputo que utiliza el código, calculado en milisegundos, y por el número de solicitudes realizadas. No hay costos asociados a la inactividad.
4. **Integración con otros servicios AWS:** Lambda se integra perfectamente con otros servicios de AWS, como el S3 que ya conocemos, facilitando la creación de aplicaciones robustas y escalables.
5. **Flexibilidad de lenguajes:** soporta una amplia gama de lenguajes de programación, incluyendo Node.js, Python, Java y C#, entre otros. Esta diversidad ofrece una notable flexibilidad a los desarrolladores, ya que no están restringidos a utilizar únicamente el lenguaje predominante en su stack actual. En su lugar, pueden optar por el lenguaje y las bibliotecas que sean más ventajosos para la tarea específica que desean realizar. Por ejemplo, una función que genera un mapa puede requerir miles de líneas de código en Java, mientras que la misma funcionalidad podría lograrse con solo unas pocas líneas en Python. Esta capacidad de elegir el mejor lenguaje para cada situación no solo optimiza el desarrollo, sino que también mejora la eficiencia y la productividad.

AWS Lambda se utiliza comúnmente para una variedad de tareas críticas en la computación moderna. Entre sus usos más destacados se encuentra el procesamiento de datos en tiempo real y la automatización de infraestructura, ejecutando scripts que responden automáticamente a cambios en la infraestructura, lo cual optimiza la gestión de recursos y la eficiencia operativa.

En el desarrollo de aplicaciones web y móviles, Lambda permite implementar lógica empresarial sin la necesidad de gestionar servidores (ya que es un servicio *serverless*), facilitando la creación de aplicaciones escalables y eficientes. También es ampliamente utilizado para la integración y automatización de servicios, automatizando tareas de integración y procesos de negocio que mejorarían la operatividad y reducirían el esfuerzo manual.

En nuestro caso específico, hemos utilizado AWS Lambda para optimizar el procesamiento de datos almacenados en Amazon S3. Las funciones Lambda desarrolladas (en Python) permiten limpiar los datos eliminando duplicados, ajustando columnas y renombrándolas según sea necesario, garantizando así que los datos estén en el formato adecuado antes de ser insertados en nuestras bases de datos. Esta automatización no solo mejora la calidad de los datos, sino que también agiliza significativamente el flujo de trabajo.

Una vez limpios, los datos se insertan en las bases de datos mediante funciones lambda, esta vez escritas en Java. Los datos se almacenan en tablas con el prefijo ETL (por ejemplo, ETL_instancia_columnas, ETL_instancia_operaciones, etc.), que se guardan en Amazon RDS.

Finalmente, se corre una step function en AWS que coordina la ejecución de varios procesos de base de datos, asegurando que los datos estén correctamente procesados y listos para ser utilizados en la herramienta final. El flujo de dicha step function es el mostrado en la siguiente ilustración (*Ilustración 11*).

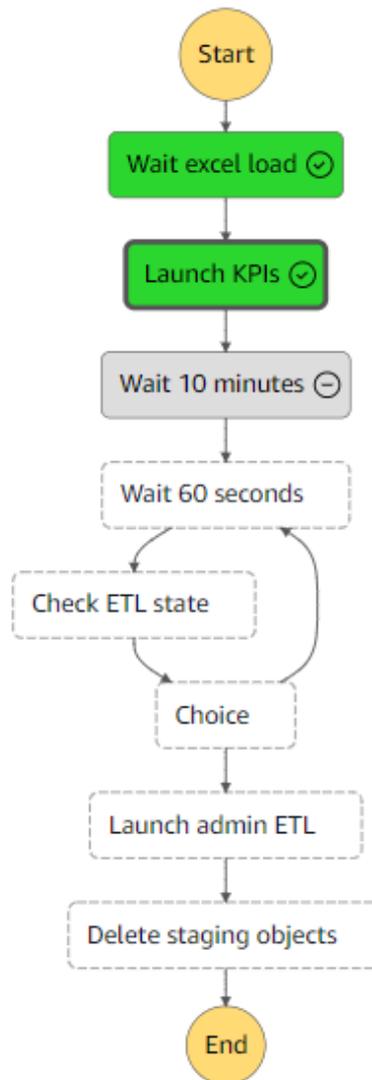


Ilustración 11. Flujo de la AWS Step Function - cálculos producto final

Arriba se describe el flujo de trabajo automatizado que se utiliza para ejecutar varios procesos de base de datos. En otras palabras, las tablas obtenidas hasta ahora, no son las utilizadas en el producto final. Esta función es la que realiza los cálculos sobre los datos, para ya hacer las inserciones en las tablas que sí se utilizarán para producción. Aquí está el desglose paso a paso de lo que hace cada etapa en el flujo:

1. Start: el flujo de trabajo comienza.
2. Wait excel load: espera a que se cargue un archivo de Excel. Este paso es esencial para asegurar que los datos necesarios estén disponibles antes de proceder.
3. Launch KPIs: después de cargar el archivo de Excel, se lanzan los indicadores clave de rendimiento (KPIs). Este paso puede implicar la ejecución de cálculos o la actualización de métricas clave basadas en los datos del Excel.
4. Wait: tiempo para que los procesos de KPIs se completen completamente o para sincronizarlos con otros procesos.
5. Check ETL state: comprueba el estado del proceso ETL (Extract, Transform, Load). Este paso verifica si el proceso ETL ha terminado o está en una etapa específica.
6. Choice: basado en el estado del ETL, toma una decisión:
 - Si el proceso ETL está completo o en el estado deseado, procede al siguiente paso.
 - Si no, podría repetir la espera y la verificación del estado hasta que se cumpla la condición necesaria.
7. Launch admin ETL: una vez que el ETL está en el estado adecuado, se lanza un proceso ETL administrativo.
8. Delete staging objects: finalmente, elimina los objetos de staging. Los objetos de staging son temporales y se usan para almacenar datos intermedios durante los procesos ETL. Este paso asegura que no queden datos temporales que puedan afectar futuros procesos o consumir espacio innecesariamente.
9. End: el flujo de trabajo termina.

Como ya se ha mencionado, el output de esta función son los datos que se muestran en la aplicación, la cual corresponde al bloque llamado “INTEL” en la *Ilustración 9*.

5.2.2 MEJORA DE LA ARQUITECTURA CON TECNOLOGÍA BIG DATA

La arquitectura actual, aunque funcional, presenta ciertos desafíos y limitaciones que pueden ser optimizados mediante la implementación de tecnologías Big Data. El esquema de la *Ilustración 12* compara ambas arquitecturas, la que se tiene actualmente y la que se quiere implementar.

Antes de adentrarnos en esta propuesta, hay otra que se ha planteado para hacer una mejora sobre la fuente de datos. Con esta se pretende eliminar el formato de archivos excel y trabajar directamente con APIs proporcionadas por Orbis. Esto permitirá una integración mucho más eficiente y reducirá los pasos necesarios para la transformación de datos. Estos datos provenientes de la API se procesan directamente mediante funciones lambda, que insertan los datos en las bases de datos sin necesidad de múltiples pasos intermedios. Esto simplifica el flujo de trabajo y reduce el tiempo de procesamiento.

Ahora sí, con respecto a la arquitectura Big Data, la primera gran diferencia con respecto a la tecnología actual es el formato de almacenamiento de los datos, ya que con esta nueva tecnología, se utilizaría el formato parquet.

Apache Parquet es un formato de almacenamiento de datos columnar, optimizado para el procesamiento y la consulta de grandes volúmenes de datos. Fue diseñado específicamente para trabajar con sistemas de Big Data y proporciona una forma eficiente de almacenar y acceder a datos en un entorno distribuido. Las ventajas de este formato son las siguientes:

1. Eficiencia en el almacenamiento:
 - Compresión superior: Parquet utiliza técnicas de compresión avanzadas, lo que reduce significativamente el tamaño del archivo almacenado. A diferencia de los formatos fila-orientados como formatos más tradicionales como el CSV, donde la compresión es menos efectiva debido a la estructura del archivo, Parquet puede comprimir columnas enteras de datos similares, logrando una mayor reducción del tamaño.

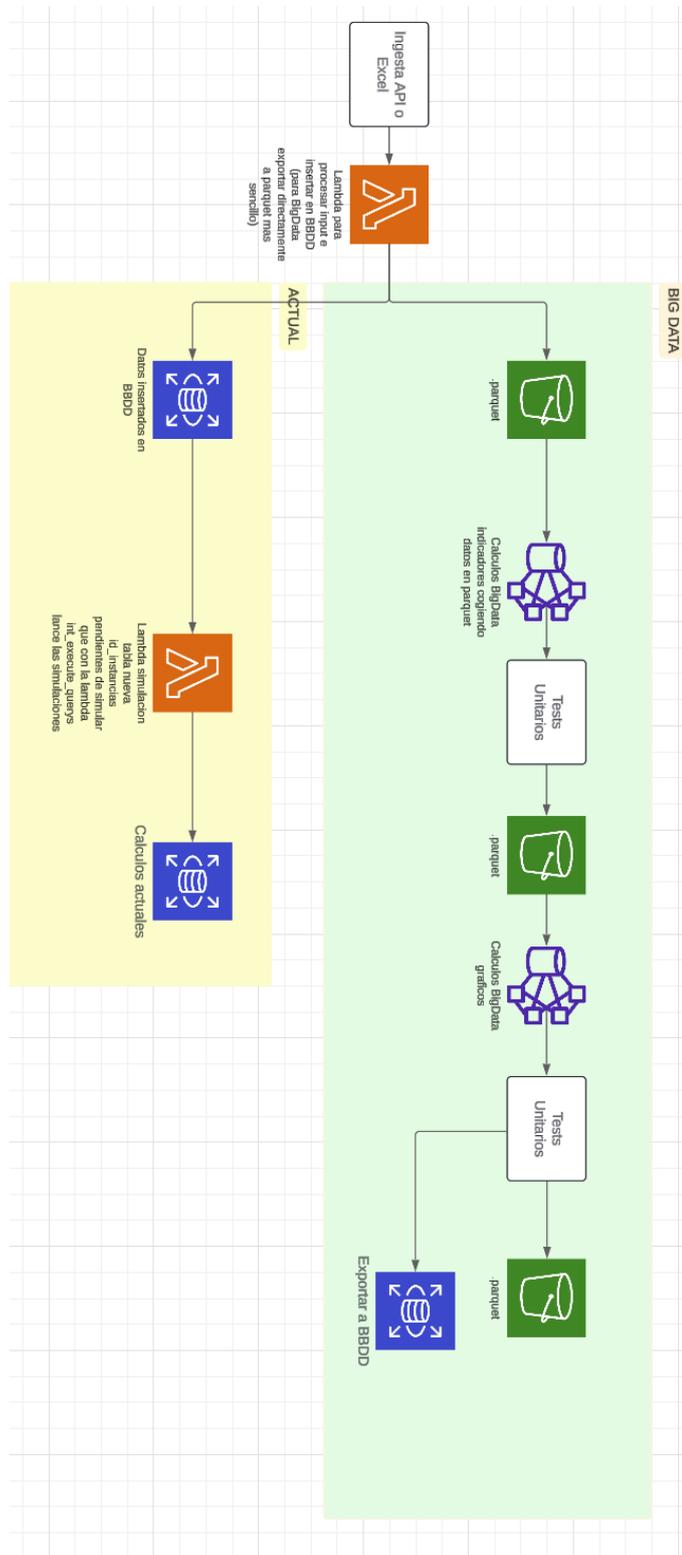


Ilustración 12. Comparación de arquitecturas

2. Mejora en el acceso a los datos:

- Acceso rápido: debido a su naturaleza columnar, Parquet permite acceder y procesar solo las columnas necesarias para una consulta específica, en lugar de leer todo el archivo como sucede con CSV o JSON. Esto mejora considerablemente la velocidad de las consultas y reduce el tiempo de procesamiento.
- Lectura selectiva: las aplicaciones pueden leer únicamente las columnas relevantes, lo que resulta en un menor uso de ancho de banda y memoria, optimizando así el rendimiento.

3. Optimización para consultas analíticas:

- Procesamiento eficiente: este formato está diseñado para integrarse con herramientas de procesamiento de Big Data como Apache Spark, Hive y Impala. Estas herramientas pueden explotar la estructura columnar de Parquet para realizar consultas analíticas complejas de manera más eficiente.
- Almacenamiento de tipos de datos complejos: el formato Parquet soporta tipos de datos complejos, como estructuras anidadas y arreglos, proporcionando flexibilidad para almacenar datos ricos y variados de manera estructurada.

4. Compatibilidad y Ecosistema:

- Integración con herramientas de Big Data: Parquet es ampliamente compatible con muchas herramientas y frameworks de Big Data. Esto facilita su adopción y uso en diferentes plataformas y aplicaciones, asegurando que los datos almacenados en este formato puedan ser utilizados y procesados por una variedad de sistemas.

- Estandarización: este formato se ha convertido en un estándar de facto para el almacenamiento de datos en entornos de Big Data, lo que garantiza la interoperabilidad y la facilidad de uso a largo plazo.

Además, este formato, comparándolo con otros formatos más tradicionales, presenta muchos beneficios. Repasemos los inconvenientes que suponen estos otros formatos:

- CSV

CSV almacena datos en un formato fila-orientado, donde cada registro es una línea separada por comas. Esto puede resultar en archivos de gran tamaño y menos eficiencia en la compresión. Leer un archivo CSV completo para acceder a una sola columna puede ser ineficiente, especialmente con grandes volúmenes de datos.

- JSON

JSON almacena datos en un formato de texto que representa estructuras jerárquicas. Aunque es flexible y legible, puede ser ineficiente en términos de espacio y velocidad de acceso. La estructura de JSON introduce una sobrecarga adicional debido a la sintaxis y los delimitadores, lo que aumenta el tamaño del archivo y reduce la eficiencia de la compresión.

En conclusión, el uso del formato Parquet para el almacenamiento de datos en mscope permitiría una mayor eficiencia en el almacenamiento y acceso a los datos, optimizando los procesos de Big Data. La estructura columnar de Parquet, junto con sus capacidades de compresión y procesamiento rápido, lo convierte en la opción ideal para manejar grandes volúmenes de datos y realizar análisis complejos de manera eficiente. Esta mejora es crucial para garantizar la consistencia y calidad de la información frente a los grandes volúmenes de datos que se esperan obtener con la expansión geográfica.

Una vez adaptado este formato, y habiendo almacenado los datos en Parquet, estos serán procesados en clusters EMR. Esto permite realizar cálculos complejos y análisis de grandes volúmenes de datos de manera eficiente y escalable.

Antes de exportar los resultados a la base de datos, se realizarán tests unitarios para asegurar la calidad y precisión de los datos procesados. Esta práctica es esencial para evitar errores en el producto final que puedan afectar la experiencia del usuario. Una vez hechos los cálculos, los tests unitarios verificarán que todo ha ido correctamente.

Implementar tests unitarios es una buena práctica no solo en el ámbito del Big Data, sino en cualquier tipo de tecnología. Actualmente, en nuestro proceso, no se realizan estos tests, lo que ha causado numerosos problemas en el entorno de producción (PRO). Los errores que no se detectan durante el procesamiento se trasladan al producto final, afectando directamente al consumidor. Este tipo de fallos ha provocado que los clientes sufran inconvenientes que podrían haberse evitado con una adecuada implementación de tests.

La falta de tests unitarios ha generado situaciones en las que los errores pasaron desapercibidos, impactando negativamente en la calidad del producto y en la confianza de los usuarios. Para solucionar esto, con la nueva tecnología Big Data, planeamos implementar un riguroso sistema de tests unitarios que verifiquen cada paso del procesamiento de datos.

Esta implementación permitirá identificar y corregir errores de manera proactiva, garantizando que los datos exportados a las bases de datos sean precisos y fiables. De este modo, se mejorará significativamente la calidad del producto final y se evitarán problemas que puedan afectar al consumidor, asegurando una experiencia de usuario óptima y libre de fallos.

Una vez verificados, los datos se exportarán a las bases de datos, listos para ser utilizados en la aplicación final. Se puede observar en el diagrama de Big Data de la Ilustración 12 que los cálculos de los datos se separan según la naturaleza de estos: datos de indicadores y datos para pintar gráficos. Esto se haría así para optimizar el proceso, ya que en caso de necesitar recalcular tan solo un tipo de dato, no se tienen que volver a hacer cálculos sobre otros datos de distinta naturaleza.

Lo más importante a destacar es que esta nueva arquitectura de Big Data solucionaría uno de los problemas más críticos a los que actualmente se enfrenta la aplicación: tener todos

los cálculos acoplados a la base de datos. Por ejemplo, cuando se realizan cálculos de rating de empresas, se produce inevitablemente un downtime en la web. Esto ocurre porque el uso de la CPU de la base de datos aumenta significativamente, y debido a que la aplicación está acoplada a la base de datos, el sistema no puede soportar la carga y se cae. Al desacoplar los cálculos de la base de datos utilizando la infraestructura de Big Data, evitaríamos estos problemas, mejorando significativamente la estabilidad y el rendimiento del sistema.

Aunque la arquitectura de Big Data ofrece una solución robusta, no es la única técnica de desacoplamiento disponible. Una alternativa es la técnica de Read Replica. Esta consiste en crear réplicas de lectura de la base de datos principal. Estas réplicas pueden manejar solicitudes de lectura, distribuyendo la carga de trabajo y reduciendo el impacto en la base de datos principal. Las Read Replicas se utilizan comúnmente para escalar aplicaciones de lectura intensiva y mejorar el rendimiento sin afectar las operaciones de escritura en la base de datos principal.

Las ventajas de Read Replica incluyen la distribución de la carga de trabajo al permitir que múltiples réplicas manejen las consultas de lectura, lo que reduce la contención de recursos en la base de datos principal. Además, mejora el rendimiento de las operaciones de lectura, permitiendo que estas se realicen de manera más eficiente. Sin embargo, presenta algunas limitaciones, como la consistencia eventual, ya que las réplicas de lectura pueden no estar completamente actualizadas con los últimos datos de la base de datos principal. Además, no reduce la carga de las operaciones de escritura en la base de datos principal, lo que puede seguir siendo un cuello de botella en aplicaciones con muchas escrituras.

Y bien, aunque la técnica de Read Replica ofrece ciertas ventajas, la arquitectura de Big Data con EMR y Hadoop proporciona una solución más completa y eficiente para nuestro caso específico por varias razones.

Primero, la infraestructura de Big Data está diseñada para escalar horizontalmente de manera eficiente. Podemos agregar o reducir recursos en los clústeres de EMR según las necesidades, lo que facilita la gestión del crecimiento en volumen de datos y usuarios que implicaría la expansión geográfica a la que la empresa se quiere someter. Por otro lado,

aunque las Read Replicas escalan las operaciones de lectura, no proporcionan una solución para las operaciones de escritura intensivas ni ofrecen la misma flexibilidad en la gestión de recursos.

Además, la arquitectura de Big Data desacopla completamente los cálculos de la base de datos, evitando que los problemas en un proceso afecten a otros. Esto mejora significativamente la fiabilidad y disponibilidad del sistema. En contraste, aunque las Read Replicas ayudan a separar las lecturas, las operaciones de escritura y otros procesos críticos aún pueden impactar la base de datos principal.

Por último, la arquitectura de Big Data es ideal para una organización en expansión geográfica. Podemos ajustar fácilmente los recursos necesarios para soportar nuevas regiones sin complicaciones, simplemente escalando el clúster de EMR según sea necesario. Las Read Replicas, sin embargo, no proporcionan la misma facilidad para adaptarse a cambios en la escala y en la arquitectura geográfica.

En conclusión, la arquitectura de Big Data con EMR y Hadoop es la opción más adecuada para desacoplar los cálculos de la base de datos en nuestra infraestructura, solucionando problemas críticos de downtime y contención de recursos al realizar cálculos de rating. Al desacoplar los cálculos y realizarlos en clústeres EMR, se evita el downtime en la plataforma, mejorando la disponibilidad y la experiencia del usuario. Además, esta arquitectura elimina la necesidad de múltiples tablas intermedias, reduciendo el espacio de almacenamiento utilizado y simplificando el esquema de la base de datos. La tecnología Big Data es altamente escalable, permitiendo procesar grandes volúmenes de datos en paralelo; actualmente, el departamento de datos tarda dos semanas en simular todas las empresas (alrededor de 38.000) porque solo puede procesar 400 a la vez, pero con Big Data, este proceso se reduciría a solo 2-3 horas. Finalmente, con la nueva arquitectura, será posible calcular y analizar datos de empresas de diferentes regiones simultáneamente, algo que la infraestructura actual no permite. Esto proporciona una plataforma escalable y adaptable para el crecimiento futuro de la empresa, garantizando una mejor gestión y procesamiento de datos a medida que la organización se expande.

5.3 BACKEND

Con la entrada del cliente del Reino Unido, surgió un problema significativo con la estructura de carpetas del backend: la diferenciación entre los distintos clientes. La solución rápida, aunque no del todo convencional, y para anda escalable, fue la utilización de condicionales if-else que leían el ID del cliente que se quería utilizar. Por ejemplo, si un cliente no quería ver el mapa en el dashboard, se ocultaba específicamente para ese cliente utilizando un if-else. Esta práctica, aunque permitió tener el producto listo para el cliente en el corto plazo, resultó ser ineficiente y poco manejable a largo plazo.

Estas malas prácticas son comunes en entornos laborales donde el cliente demanda una solución para una fecha determinada. Aunque proporcionar una solución rápida puede parecer beneficioso inicialmente, es contraproducente a largo plazo, ya que posteriormente se deben corregir estas malas prácticas y reemplazarlas con soluciones robustas y escalables. Idealmente, se debe buscar flexibilidad con el cliente para entregar un producto final de calidad, tanto visualmente como en su arquitectura interna, lo que asegura menos problemas futuros.

5.3.1 ONE INTEL PRODUCT

La necesidad de mejorar estas prácticas condujo a la implementación del proyecto One Intel Product (OIP). Este proyecto es similar en concepto a los microservicios y tiene como objetivo separar las funcionalidades en partes más pequeñas, haciendo que sean más manejables y reutilizables. La arquitectura OIP busca separar las funcionalidades privadas y públicas, creando un módulo específico para clientes privados.

La principal razón para implementar OIP es mejorar la modularidad y reutilización del código. Al dividir las funcionalidades en módulos más pequeños, se facilita su mantenimiento y escalabilidad. Esto permite adaptar y personalizar el producto según las necesidades específicas de cada cliente sin introducir malas prácticas en el código.

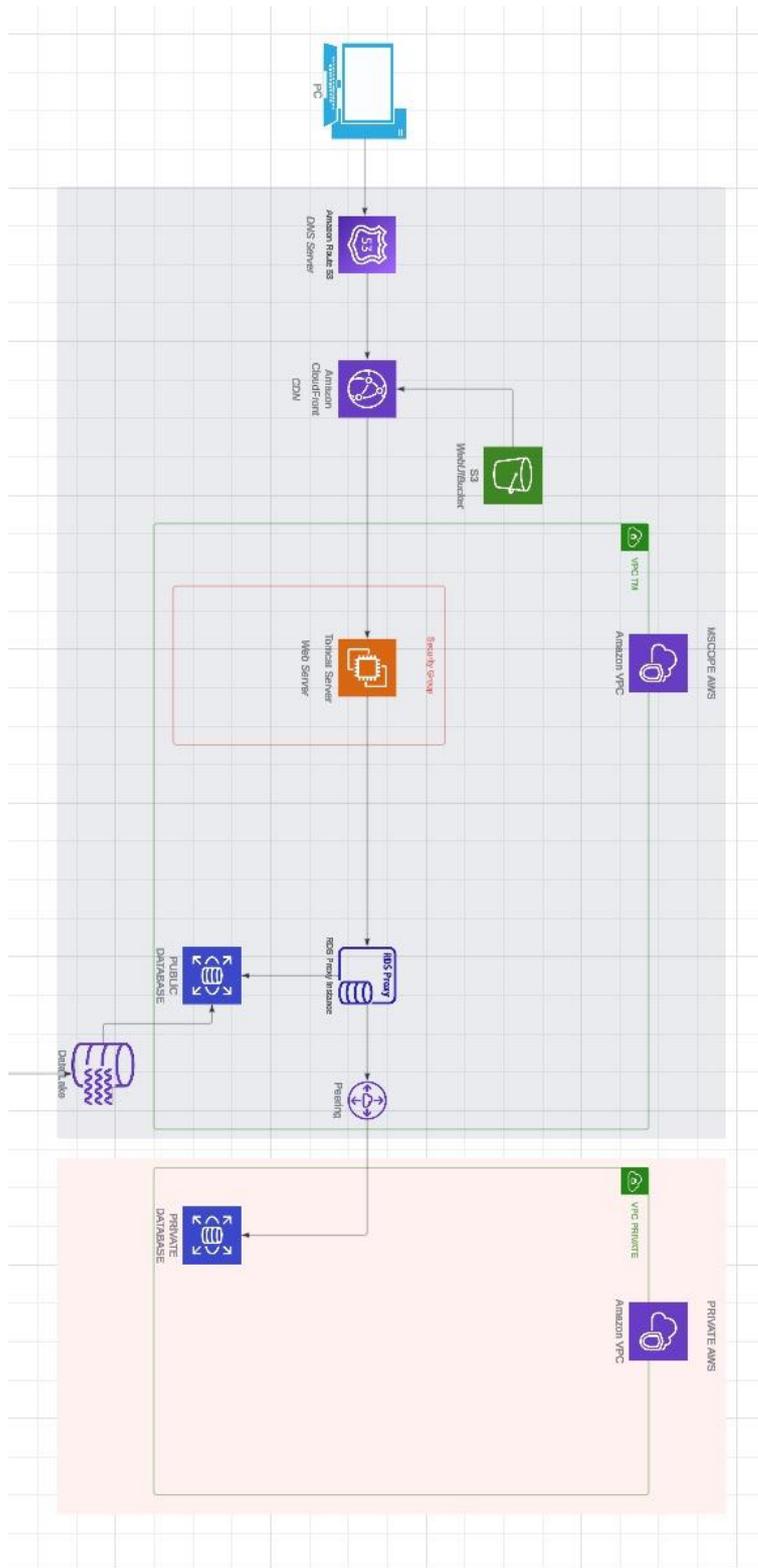


Ilustración 13. Arquitectura OIP

Antes de la expansión a Reino Unido, nuestra arquitectura solo comprendía la parte en gris mostrada en la Ilustración 13. Con la expansión a UK, aparece la parte en rojo del esquema, representando los nuevos requerimientos y funcionalidades de este cliente. El OIP introduce el concepto de "peering", que facilita la integración eficiente y sin conflictos entre las diferentes partes de la arquitectura. Peering es una técnica que conecta dos VPCs (Virtual Private Clouds), permitiendo que se comuniquen directamente sin pasar por la red pública de Internet, lo que mejora la seguridad y la latencia.

A continuación, se describe la arquitectura componente a componente de izquierda a derecha:

- PC del Usuario: Punto de entrada para los usuarios que acceden a la aplicación.
- Amazon Route 53: Servidor DNS que dirige el tráfico al servidor adecuado.
- Amazon CloudFront: Red de entrega de contenido (CDN) que distribuye el contenido desde S3.
- Amazon S3: Almacenamiento para archivos estáticos y contenido del sitio web.
- Servidor Tomcat: Servidor web que maneja las solicitudes y sirve el contenido dinámico.
- Amazon RDS: Base de datos relacional que almacena los datos de la aplicación.
- Data Lake: Almacenamiento de datos a gran escala para análisis y procesamiento de Big Data.
- Peering: Conexión entre VPCs para asegurar la comunicación entre las diferentes partes de la arquitectura.
- VPCs (Virtual Private Clouds): Redes privadas que contienen y gestionan los recursos de la aplicación, separando la parte pública y privada.

El flujo comienza cuando un usuario accede a la aplicación desde su PC. La solicitud del usuario es dirigida por Amazon Route 53, que actúa como servidor DNS, hacia el servidor adecuado. Amazon CloudFront, la red de entrega de contenido (CDN), distribuye el contenido estático almacenado en Amazon S3, asegurando una entrega rápida y eficiente. El Servidor Tomcat maneja las solicitudes dinámicas, interactuando con Amazon RDS para

acceder a la base de datos relacional donde se almacenan los datos de la aplicación. Para análisis y procesamiento de grandes volúmenes de datos, se utiliza el Data Lake. La comunicación entre las diferentes partes de la arquitectura se facilita mediante Peering, que conecta las VPCs públicas y privadas, asegurando una transferencia de datos segura y eficiente. Las VPCs contienen y gestionan todos los recursos de la aplicación, separando las funciones públicas y privadas para mejorar la seguridad y la organización de la infraestructura.

En resumen, la implementación del OIP y la evolución de nuestra arquitectura han permitido una mejor gestión y escalabilidad de nuestras aplicaciones, asegurando que podamos satisfacer las necesidades de nuestros clientes de manera eficiente y con buenas prácticas de desarrollo.

5.3.2 ESTRUCTURA DE CARPETAS DEL BACKEND

El OIP, como es evidente, también afecta a la estructura de carpetas del backend. Anteriormente, la estructura de carpetas del backend no estaba modularizada y solo contenía el módulo público (que no se llamaba así, ya que aún no existían módulos diferenciados). Sin embargo, con la implementación del Proyecto One Intel Product (OIP), se adopta una estructura modular como se muestra en la Ilustración 14. Esta estructura mejora significativamente las prácticas de desarrollo al separar las lógicas como se explicara a continuación.

El módulo público contiene toda la lógica de negocio que es común para todos los clientes. Esto incluye:

- Fórmulas económicas e indicadores: incluye cálculos y algoritmos financieros, además de métricas estándar (como el EBITDA) que son relevantes para todos los clientes.
- Gráficas: visualizaciones de datos que representan información económica y financiera de manera uniforme para todos los usuarios.



Ilustración 14. Estructura de carpetas del backend

El módulo privado, por otro lado, está diseñado para soportar la lógica específica de cada cliente. Esto incluye variaciones en los requisitos de negocio que no deberían impactar la lógica pública. Ejemplos de lógica específica pueden incluir:

- Especificaciones del cliente: diferentes clientes pueden tener requisitos específicos que varían considerablemente. Por ejemplo, mientras que un cliente puede querer visualizar una gráfica, otro puede preferir ver los mismos datos en formato de tabla. En términos de lógica de backend, esto puede traducirse en diferentes métodos de cálculo o procesamiento de datos según las necesidades del cliente.
- Lógica de negocio específica por país: algunos indicadores o cálculos pueden ser más relevantes en ciertos países. Por ejemplo, España puede tener un interés particular en un indicador específico que no es tan relevante en el Reino Unido. Del mismo modo, los métodos de cálculo pueden variar entre países, o pueden ser necesarios nuevos cálculos que sean específicos para un cliente en particular.

La separación en módulos públicos y privados permite mantener una base de código limpia y manejable. Las especificaciones privadas se manejan de manera aislada, asegurando que no interfieran con la lógica común que es utilizada por todos los clientes. Esta modularidad facilita la adición de nuevas características y personalizaciones específicas para clientes sin comprometer la estabilidad y el rendimiento de la aplicación global.

En resumen, la estructura modular introducida por el OIP no solo mejora las prácticas de desarrollo, sino que también garantiza que las personalizaciones específicas de los clientes no impacten negativamente en la lógica compartida. Esto permite una mayor flexibilidad y adaptabilidad en la implementación y gestión de los requisitos de los clientes.

Para terminar de entender este concepto de modularización, se podría decir hemos dividido el proyecto en tres partes: Admin App (o Admin Logics), Shared Library (o Common Code), y Customer App (o Customer Logics). Esta separación facilita la modularización del código y asegura una mejor mantenibilidad y escalabilidad del proyecto.

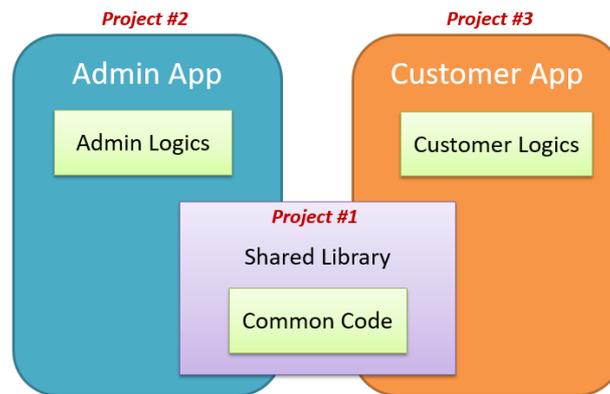


Ilustración 15. Visual auxiliar sobre la división del proyecto

La Admin App contiene la lógica pública común a todos los clientes, actuando como la base del proyecto. Esta lógica incluye todas las funcionalidades y cálculos que son universales y no dependen de ningún cliente específico. La Admin App es fundamental ya que proporciona las operaciones estándar y las funcionalidades básicas que utilizan todos los módulos privados y públicos.

La Shared Library contiene el fichero necesario para lanzar, ejecutar y arrancar la aplicación. Incluye todas las librerías necesarias para todos los proyectos, asegurando que cada módulo pueda acceder a las dependencias comunes sin duplicar código. Esta biblioteca compartida actúa como un núcleo centralizado de recursos y configuraciones.

La Customer App maneja las especificaciones de cada cliente. Esta parte del proyecto incluye la lógica específica que varía de un cliente a otro. Por ejemplo, diferentes métodos de cálculo o procesamiento de datos que se adaptan a las necesidades particulares de cada cliente, asegurando que las personalizaciones no interfieran con la lógica común.

Cada módulo en nuestra estructura tiene un archivo pom.xml, que es un fichero de configuración de construcción para Maven. Maven es una herramienta de gestión de proyectos y comprensión que proporciona una forma estándar de construir aplicaciones Java. Maven gestiona las dependencias, la compilación y los ciclos de vida del proyecto, facilitando la integración continua y el despliegue automatizado. Además, Maven puede gestionar proyectos desde la construcción inicial hasta la publicación, incluyendo la documentación, la generación de informes y la gestión de la configuración del proyecto.

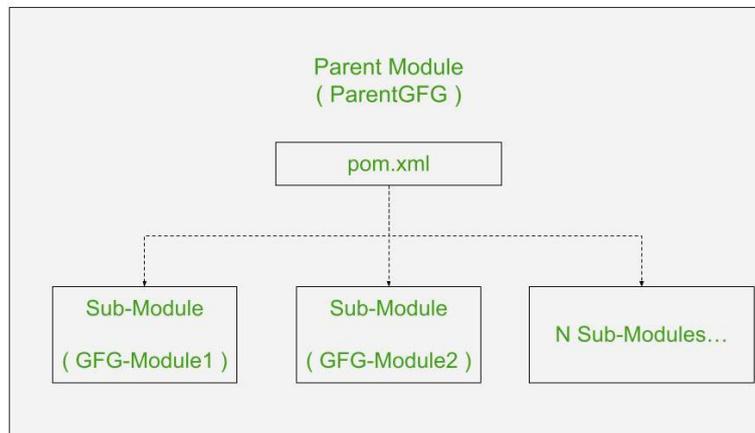


Ilustración 16. Estructura de módulos

Esta visualización (*Ilustración 16*) detalla la estructura modular utilizando Maven. El módulo principal (Parent Module) contiene submódulos (Sub-Modules) como el common module private, module private, y el module public. Cada submódulo tiene su propio archivo pom.xml para manejar las dependencias específicas.

La adopción de una estructura modular facilita la gestión y escalabilidad del proyecto. La separación en Admin App, Shared Library y Customer App asegura que el código sea más manejable y adaptable a las necesidades específicas de cada cliente sin comprometer la lógica común. Utilizar Maven para gestionar las dependencias y la construcción del proyecto refuerza esta modularidad, permitiendo una integración y despliegue más eficiente.

5.3.3 CÓDIGO DEL BACKEND

Hasta ahora hemos discutido la estructura modular del proyecto. Ahora, centrémonos en cómo se gestionan las conexiones y la lógica específica de cada cliente a nivel de código en el backend utilizando AOP (Programación Orientada a Aspectos).

AOP, o Programación Orientada a Aspectos, es una metodología complementaria a la Programación Orientada a Objetos (OOP). AOP permite separar las preocupaciones transversales del código principal de la aplicación, como la gestión de transacciones, la seguridad, y en nuestro caso, la lógica de enrutamiento de bases de datos. AOP facilita la modularización de estas preocupaciones transversales, mejorando la mantenibilidad y escalabilidad del código.

En el contexto de Spring Framework, AOP es una funcionalidad poderosa que permite interceptar las peticiones antes de que lleguen a la base de datos. Esto es especialmente útil para determinar a qué base de datos debe dirigirse una solicitud, basándose en ciertas condiciones o parámetros.

La Programación Orientada a Aspectos (AOP) ofrece varias ventajas significativas. Primero, permite la modularización de la lógica transversal, separando claramente las preocupaciones transversales del código de negocio principal. Esto no solo organiza mejor el código sino que también mejora su legibilidad. En segundo lugar, AOP reduce la duplicación de código, evitando la necesidad de manejar la lógica de enrutamiento de bases de datos en múltiples lugares. Al centralizar esta lógica en aspectos, se elimina el código repetitivo y se promueve la reutilización de código. Por último, AOP mejora la mantenibilidad del código, ya que facilita la modificación y mantenimiento del mismo. Al tener la lógica transversal centralizada en aspectos, cualquier cambio necesario se realiza en un solo lugar, simplificando las actualizaciones y reduciendo el riesgo de errores.

5.3.4 OTRAS MEJORAS

5.3.4.1 Optimización: Memoria Caché y Compresión

En el backend, utilizamos el sistema de caché que proporciona por defecto el framework de Spring Boot. Este sistema asigna una anotación `@Cacheable` a cada método que requiere mucho tiempo. Dicha anotación verifica si la petición está devolviendo los mismos datos. Si es así, no vuelve a llamar a la base de datos, sino que entrega lo que tiene guardado en la memoria caché. Esta técnica mejora significativamente el rendimiento al reducir el número de llamadas a la base de datos.

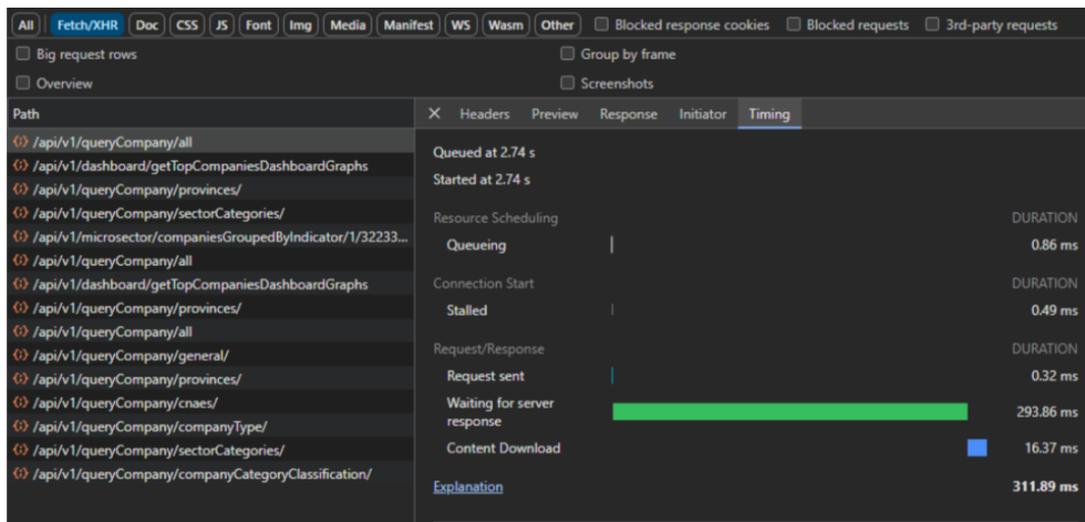


Ilustración 17. Inspector de web - Timing

Detectamos la necesidad de optimización de caché al observar el rendimiento en el inspector de la web. En la la figura se puede observar una barra verde que representa el tiempo de espera de la respuesta del servidor (waiting for server response) y una barra azul que muestra el tiempo de descarga del contenido (content download). Cuando la barra verde supera los 2-3 segundos, se considera una alarma roja, indicando que el usuario experimentará una latencia notable. Esta observación nos llevó a implementar la caché para mejorar la velocidad de respuesta del sistema.

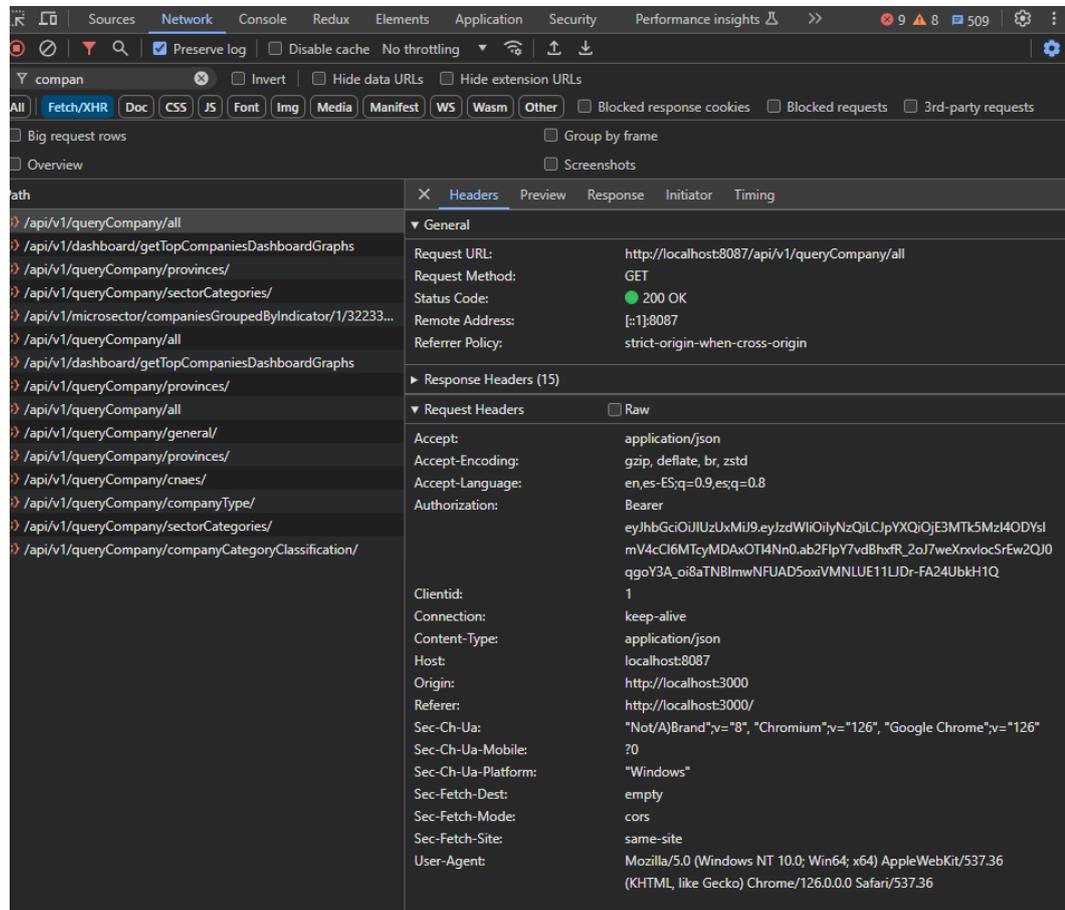


Ilustración 18. Inspector de web - Headers

Dentro de este mismo estudio, pero esta vez revisando los headers de respuesta (como se muestra en la figura superior), también descubrimos que el backend no estaba comprimiendo las respuestas. Esto provocaba que las peticiones que involucraban grandes volúmenes de datos tardaran mucho más debido al tamaño de los datos transferidos. Implementamos la compresión gzip en el servidor Apache del backend, lo que redujo significativamente el tamaño de las respuestas y mejoró los tiempos de carga.

5.3.4.2 Multicliente

El objetivo de esta mejora es permitir que un usuario pueda gestionar varios clientes sin necesidad de cerrar sesión. Anteriormente, cada usuario solo podía acceder a un cliente a la vez, lo que resultaba tedioso ya que requería cerrar sesión y volver a iniciar sesión para cambiar de cliente. Con la nueva funcionalidad multicliente, un usuario puede acceder a todos sus clientes con un solo inicio de sesión, utilizando un modal para cambiar de perfil de cliente fácilmente.

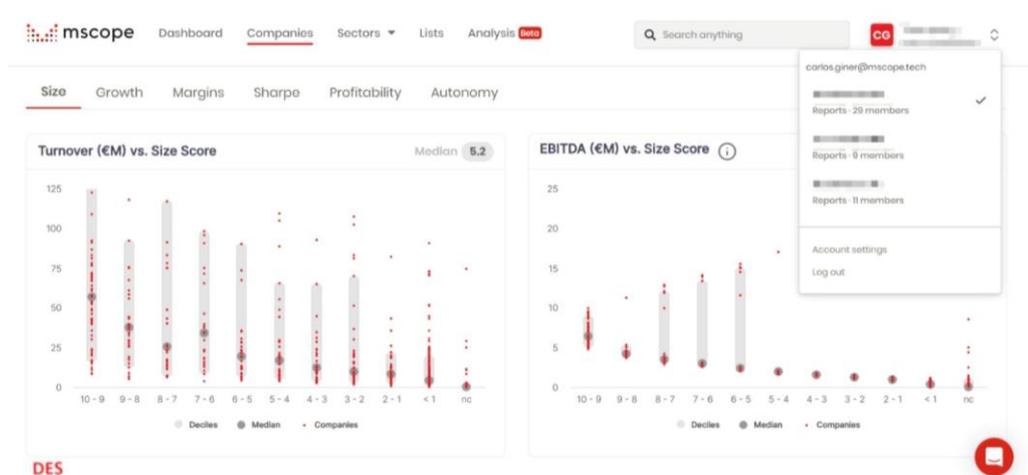


Ilustración 19. Modal multicliente

Esta mejora se observa en la figura superior, donde se muestra cómo un usuario puede iniciar sesión una sola vez y tener acceso a múltiples clientes desde un modal intuitivo y fácil de usar. Esto no solo mejora la experiencia del usuario, sino que también facilita la gestión de múltiples clientes bajo un solo correo electrónico. Esta funcionalidad no solo incrementa la eficiencia del usuario, sino que también hace que la herramienta sea más flexible y adaptable a las necesidades de los clientes que manejan múltiples cuentas.

En conjunto, estas mejoras han sido cruciales para optimizar el rendimiento del sistema y mejorar la experiencia del usuario, asegurando que la plataforma sea más eficiente, flexible y fácil de usar.

5.4 FRONTEND

Con la expansión del proyecto al Reino Unido, surgieron diversos desafíos que requirieron adaptaciones tanto en el backend como en el frontend de nuestra aplicación. Una tarea específica que abordé fue la gestión de la responsividad de la divisa utilizada en la herramienta financiera. Dado que muchos gráficos, títulos, tablas y datos están acompañados por el símbolo del euro (€), el código original tenía estos símbolos incrustados (hardcodeados), lo que generó problemas al incorporar datos del Reino Unido, cuya divisa es la libra esterlina (£ o pound).

El principal problema era que, aunque los datos numéricos se calculaban correctamente (como ya se ha explicado en el apartado de datos), la divisa no se ajustaba automáticamente cuando se cambiaba al cliente de Reino Unido. Esto se debía a que los símbolos de divisa estaban hardcodeados en múltiples partes del código. Por lo tanto, al introducir nuevos datos de UK, faltaba ajustar manualmente la divisa en cada lugar relevante, lo que no era escalable ni mantenible.

Para resolver este problema, llevé a cabo una serie de pasos meticulosos para asegurar que la divisa pudiera cambiarse dinámicamente sin necesidad de modificaciones manuales repetitivas en el futuro.

1. Identificación de Ubicaciones:

- Primero, realicé una auditoría exhaustiva de todas las páginas de la web, incluidas las pestañas, tablas y gráficos, para identificar dónde se utilizaban símbolos de divisa. Esto implicó revisar tanto el código en back como en front, así como en los archivos JSON o CSS asociados. También había algún símbolo de divisa que provenía directamente de las tablas en base de datos.

2. Creación de una función dinámica:

- Desarrollé una función que centraliza la gestión de las divisas. Esta función toma un parámetro que especifica la divisa (por ejemplo, EUR para euros, GBP para libras) y devuelve el símbolo correspondiente.

3. Refactorización del código:

- Sustituí todos los símbolos de divisa hardcoded por llamadas a esta nueva función. Esto garantizó que cualquier actualización o adición de una nueva divisa solo requeriría modificar la función en un solo lugar.

4. Pruebas:

- Realicé pruebas exhaustivas para asegurar que la función se comportara correctamente en todas las partes de la aplicación y que todas las tablas, gráficos y títulos mostraran la divisa correcta basada en el cliente activo.

Esta solución dinámica no solo resolvió el problema de las libras, sino que también preparó la aplicación para futuras expansiones a otros mercados con diferentes divisas. Ahora, cualquier nueva divisa puede ser añadida simplemente actualizando la función, sin necesidad de buscar y modificar manualmente múltiples partes del código.

La tarea de gestionar dinámicamente las divisas en la herramienta financiera, aunque una tarea pequeña, fue un paso importante para preparar la aplicación para su búsqueda expansión internacional. A través de la identificación meticulosa de los problemas, la creación de una solución centralizada y la refactorización del código, logramos una solución que no solo resuelve los problemas actuales, sino que también facilita futuras expansiones y mejoras. Esta experiencia subraya la importancia de diseñar soluciones con la escalabilidad y la mantenibilidad en mente desde el principio.

5.5 MICROSERVICIOS

La creación de microservicios para determinadas funcionalidades se plantea como una solución eficiente y escalable cuando se considera la expansión y evolución de un proyecto. Los microservicios permiten desarrollar y desplegar partes específicas de una aplicación de manera independiente, ofreciendo varios beneficios clave.

Pensar en microservicios tiene sentido por varias razones. En primer lugar, al desacoplar funcionalidades específicas en microservicios, se logra una mayor modularidad y flexibilidad en el desarrollo. Esto permite que cada microservicio se desarrolle, despliegue y escale de forma independiente, sin afectar el resto del sistema. Además, los microservicios facilitan el uso de diferentes tecnologías y lenguajes de programación para distintas funcionalidades, lo que permite aprovechar las mejores herramientas disponibles para cada tarea. Por ejemplo, una funcionalidad específica puede ser más eficiente o tener mejores bibliotecas disponibles en Python, mientras que el resto del sistema está desarrollado en Java.

Un ejemplo muy concreto, pero que justifica perfectamente la creación de un microservicio, fue la necesidad de obtener latitud y longitud para pintar las empresas en un mapa para el cliente del Reino Unido. Estos datos los teníamos de Orbis para España, sin embargo, con la fuente de Reino Unido, no disponíamos de ellos. En vez de eso, teníamos las direcciones de estas empresas. La solución que se pensó fue crear un microservicio que, dada una dirección, genere las coordenadas de latitud y longitud correspondientes. Este microservicio desacoplado no solo será útil para el Reino Unido, sino que podrá aplicarse a cualquier geografía en el futuro. Además, si este microservicio se quiere utilizar para cualquier otra cosa que no sea la web, también será viable.

Implementar esta funcionalidad como un microservicio tiene varias ventajas. Primero, permite reutilizar el servicio en diferentes contextos y aplicaciones sin duplicar código. Segundo, facilita el uso de tecnologías específicas para la tarea. En este caso, existe una excelente biblioteca en Python que realiza la conversión de direcciones a coordenadas geográficas. Al implementar esta funcionalidad en una lambda de AWS, podemos

aprovechar dicha biblioteca y encapsular la lógica en un microservicio, evitando la necesidad de implementar esta funcionalidad en Java.

Cuando se escala un proyecto, hay funcionalidades que pueden ser útiles más allá de su contexto inicial. Al diseñar estas funcionalidades como microservicios, se asegura que sean fácilmente integrables y reutilizables en otros módulos o aplicaciones. Además, los microservicios permiten una mayor flexibilidad en el stack tecnológico, ya que cada microservicio puede desarrollarse con la tecnología que mejor se adapte a sus necesidades específicas.

En resumen, la implementación de microservicios para determinadas funcionalidades tiene sentido por su modularidad, flexibilidad y capacidad para aprovechar tecnologías específicas. En el caso concreto de la geolocalización para el mapa, crear un microservicio en Python encapsulado en una lambda de AWS permite reutilizar esta funcionalidad en diferentes geografías y contextos, mejorando la eficiencia y escalabilidad del proyecto. Este enfoque ya ha sido aplicado en diversos apartados anteriores, donde se han visto ejemplos de microservicios que aportan ventajas significativas en términos de mantenimiento y evolución del sistema.

Capítulo 6. ANÁLISIS DE RESULTADOS

En este capítulo, se destacarán los resultados más relevantes obtenidos durante el desarrollo del proyecto en cada una de las áreas clave: datos, arquitectura, backend, frontend y microservicios. Además, se realizará un análisis crítico de estos resultados, evaluando tanto los éxitos como las áreas que requieren mejoras.

6.1 RESULTADOS EN DATOS

Durante el proyecto, se realizó una adaptación de los formatos de datos para ajustarse a las necesidades específicas de cada región, como se observó con la incorporación del cliente en el Reino Unido. Esta adaptación ha permitido una integración más precisa y relevante de los datos económicos y financieros, mejorando la calidad del análisis y asegurando que los datos sean comparables y útiles para la toma de decisiones.

La adaptación de datos ha sido exitosa en términos de precisión y relevancia. Sin embargo, el proceso ha sido laborioso y manual. Para mejorar, se recomienda desarrollar scripts automatizados para la adaptación de formatos de datos en futuros proyectos, lo que aumentaría la eficiencia y reduciría el margen de error.

6.2 RESULTADOS EN ARQUITECTURA

Aunque la implementación completa de la arquitectura Big Data aún está en fase de propuesta, se han identificado los beneficios potenciales de esta transición. La adopción de tecnologías como Hadoop y EMR promete mejorar la capacidad de procesamiento y almacenamiento de datos, permitiendo manejar grandes volúmenes de datos de manera más eficiente.

La propuesta de la arquitectura Big Data ha sido bien recibida y se espera que mejore significativamente el rendimiento y la escalabilidad del sistema. Sin embargo, la

implementación de esta arquitectura requerirá una gestión y monitoreo constante, así como la formación del personal en nuevas tecnologías para maximizar los beneficios potenciales. La realización de pruebas piloto podría ser un paso intermedio útil antes de una implementación completa.

6.3 RESULTADOS EN BACKEND

Se implementaron mejoras significativas en el rendimiento del backend. La utilización del sistema de caché de Spring Boot, junto con la compresión gzip en el servidor Apache, ha reducido los tiempos de respuesta y mejorado la experiencia del usuario. Estas optimizaciones han disminuido la carga en la base de datos y reducido el tamaño de las respuestas enviadas, mejorando la eficiencia general del sistema.

Otra mejora clave en el backend fue la implementación del Proyecto One Intel Product (OIP). Esta iniciativa se centró en separar las funcionalidades públicas y privadas, creando módulos específicos para cada cliente. Esta modularización ha permitido una mejor gestión del código y una mayor flexibilidad para adaptar la herramienta a las necesidades específicas de cada cliente, sin afectar la lógica común.

Las optimizaciones implementadas en el backend han demostrado ser efectivas en la mejora del rendimiento. La separación de funcionalidades mediante OIP ha mejorado la mantenibilidad y flexibilidad del sistema. Sin embargo, es crucial continuar monitoreando el sistema para identificar posibles cuellos de botella y asegurarse de que la caché y la compresión se mantengan optimizadas. Además, se debería considerar la implementación de pruebas de rendimiento continuas para anticipar y mitigar problemas futuros. La estructura modular introducida por OIP también necesita una evaluación continua para asegurar que cumple con los objetivos de flexibilidad y escalabilidad a medida que se agregan nuevos clientes y funcionalidades.

6.4 RESULTADOS EN FRONTEND

Una tarea clave realizada en el frontend fue la implementación de una función para la gestión dinámica de divisas, reemplazando los símbolos de divisa hardcoded con una función que permite cambiar fácilmente entre diferentes monedas. Esta mejora ha preparado la herramienta para manejar múltiples geografías con distintas monedas, mejorando la flexibilidad y la usabilidad.

La gestión dinámica de divisas ha mejorado significativamente la capacidad de la herramienta para adaptarse a nuevas regiones. Sin embargo, se deben realizar pruebas exhaustivas para asegurarse de que todos los componentes del frontend se actualicen correctamente con la nueva divisa. Además, sería útil desarrollar una interfaz de configuración que permita a los administradores agregar y gestionar divisas fácilmente.

6.5 RESULTADOS EN MICROSERVICIOS

La propuesta de implementar microservicios para funcionalidades específicas, como la conversión de direcciones a coordenadas geográficas, ha sido un paso importante hacia la modularización y escalabilidad del sistema. Aunque estas implementaciones están en fase de planificación, se anticipa que proporcionarán una mayor flexibilidad y permitirán el uso de diferentes tecnologías para optimizar cada servicio.

La transición a una arquitectura de microservicios promete beneficios significativos, incluyendo la mejora de la modularidad y la facilidad de mantenimiento. Sin embargo, la implementación debe ser cuidadosamente planificada y ejecutada para evitar problemas de integración y asegurar que los microservicios funcionen de manera cohesiva. Se recomienda comenzar con un microservicio piloto para evaluar su impacto antes de una implementación más amplia.

En conclusión, el proyecto ha logrado avances significativos en la mejora de datos, backend y frontend, y ha establecido una base sólida para futuras implementaciones de arquitectura Big Data y microservicios. Aunque muchos de los cambios propuestos aún están en fase de planificación, los resultados obtenidos hasta ahora demuestran un claro camino hacia la optimización y escalabilidad del sistema. Continuar con esta dirección, realizando pruebas piloto y ajustando las estrategias según sea necesario, garantizará el éxito de la expansión geográfica y la mejora continua de la herramienta mscope. Además, es muy beneficioso que la empresa cuente con una documentación detallada sobre todos estos procesos y mejoras, la cual estoy desarrollando, para que esté accesible para todos los miembros del equipo. Esto no solo facilita la comprensión y adopción de las nuevas implementaciones, sino que también asegura la consistencia y continuidad del conocimiento dentro de la organización.

Capítulo 7. CONCLUSIONES

Trabajar en este proyecto ha sido una experiencia enriquecedora que ha proporcionado múltiples aprendizajes valiosos, tanto a nivel técnico como profesional. A lo largo de los meses en la empresa, he aprendido no solo sobre la implementación y adaptación de tecnologías avanzadas, sino también sobre la importancia de una planificación y análisis exhaustivos desde el principio. Estas conclusiones resumen los aprendizajes clave y reflexiones derivadas de este proyecto.

Uno de los principales aprendizajes ha sido la necesidad de dedicar una cantidad significativa de tiempo a la fase de análisis. Esta fase es crucial para comprender en profundidad los requerimientos, las limitaciones y las oportunidades del proyecto. Un análisis detallado no solo ayuda a identificar posibles problemas y desafíos desde el principio, sino que también permite diseñar soluciones más robustas y eficientes. Aunque es algo de sentido común, en la vida real y en el mundo laboral, esta práctica es difícil de implementar y a menudo se pasa por alto. Esto se debe a que vivimos en un entorno que se mueve muy rápido y parece que todo se hace a contrarreloj. En este proyecto, un análisis exhaustivo nos permitió identificar las debilidades en nuestra estructura de backend y frontend, y planificar mejoras significativas como la implementación de microservicios y la modularización del código.

Otra lección fundamental es la importancia de pensar en términos de "1 vs N". En otras palabras, en lugar de abordar problemas de manera aislada, se debe considerar cómo diseñar el sistema para que sea adaptable y escalable en múltiples dimensiones. Por ejemplo, en lugar de pensar en cómo añadir otro país al sistema, se debe considerar cómo crear un sistema multipaís que pueda manejar fácilmente nuevas incorporaciones. Este enfoque fue crucial en la implementación de funcionalidades como la gestión dinámica de divisas y la integración de nuevos datos geográficos, donde diseñar para la escalabilidad desde el principio evitó muchos problemas futuros.

Para escalar a otro país o añadir nuevas funcionalidades, es esencial tener muy claros los costes operativos. Esto no solo implica los costes financieros, sino también el esfuerzo requerido en términos de desarrollo, mantenimiento y soporte. Proporcionar al negocio una visión clara de estos costes y esfuerzos es fundamental para tomar decisiones informadas y estratégicas. En nuestro proyecto, calcular y comunicar estos costes permitió una planificación más efectiva y una mejor alineación con los objetivos de negocio.

La escalabilidad debe ser una prioridad desde el inicio del proyecto. Muchos de los problemas que enfrentamos surgieron debido a una falta de consideración de la escalabilidad en las primeras etapas del desarrollo. Al pensar en la escalabilidad desde el principio, se pueden diseñar sistemas que no solo manejen el crecimiento actual, sino que también estén preparados para futuros desafíos y expansiones. Esto incluye adoptar arquitecturas modulares, implementar microservicios, y utilizar tecnologías que permitan un fácil escalado horizontal y vertical.

Este proyecto me ha enseñado que la clave del éxito en el desarrollo de software no reside solo en la implementación técnica, sino también en una planificación estratégica y una visión a largo plazo. La importancia de un análisis exhaustivo, la consideración de problemas en términos de 1 vs N, la claridad en los costes operativos y la priorización de la escalabilidad son lecciones que llevaré conmigo en futuros proyectos. Además, la experiencia de trabajar en un entorno profesional real ha reforzado la importancia de la comunicación, la colaboración y la adaptabilidad en el desarrollo de soluciones tecnológicas efectivas y sostenibles.

En resumen, este proyecto no solo ha sido una oportunidad para aplicar y ampliar mis conocimientos técnicos, sino también una valiosa lección sobre cómo abordar problemas complejos de manera estratégica y sostenible. Estos aprendizajes me han preparado mejor para enfrentar futuros desafíos y contribuir de manera más efectiva al éxito de proyectos tecnológicos en el futuro.

Capítulo 8. BIBLIOGRAFÍA

- [9] Amazon Web Services. (s.f.). *Amazon S3 Documentation*. Sacado de <https://docs.aws.amazon.com/s3/>
- [10] Amazon Web Services. (s.f.). *Amazon RDS Documentation*. Sacado de <https://aws.amazon.com/es/rds/>
- [11] Cerezo, F. (s.f.). *Lecciones de la asignatura de Datos Masivos / Big Data - Tema 5 - Map Reduce*. Escuela Técnica Superior de Ingeniería (ICAI).
- [12] Cerezo, F., & Vela, B. (s.f.). *Experience of the architectural evolution of a Big Data system*. VorTIC3 Research Group, Universidad Rey Juan Carlos (URJC).
- [13] Baeldung. (s.f.). *A Guide to Caching in Spring*. Sacado de <https://www.baeldung.com/spring-cache-tutorial>
- [14] Aprender Big Data. (s.f.). *Apache Parquet*. Sacado de <https://aprenderbigdata.com/apache-parquet/>
- [15] Arquitectura Java. (s.f.). *Spring AOP y Aspectos*. Sacado de <https://www.arquitecturajava.com/spring-aop-y-aspectos/>
- [16] Cincodías. (2023, 13 de noviembre). *Las pymes españolas aumentan su presencia internacional*. Sacado de https://cincodias.elpais.com/cincodias/2023/11/13/pyme/1699865705_396180.html#

ANEXO I: ALINEACIÓN DEL PROYECTO CON LOS ODS

Este proyecto se alinea con varios Objetivos de Desarrollo Sostenible (ODS).



Objetivo 4 - Educación de Calidad.

Se apoya a la educación de calidad al proporcionar a las empresas recursos para tomar decisiones más informadas y mejorar su desempeño.



Objetivo 8 - Trabajo Decente y Crecimiento Económico.

La expansión de esta herramienta en nuevas regiones puede impulsar el crecimiento económico, ya que ayudará a inversores de todo el mundo a identificar PYMEs altamente prometedoras y así, acelerar su crecimiento.



Objetivo 9 - Industria, Innovación e Infraestructura.

Al adaptar la arquitectura de esta herramienta para funcionar en múltiples regiones, se está promoviendo la innovación tecnológica y mejorando la infraestructura digital.



Objetivo 10 - Reducción de las Desigualdades.

Este proyecto podría ayudar a reducir las desigualdades en las nuevas regiones donde se aplique, ya que proporciona acceso a herramientas de inteligencia de negocios.



Objetivo 11 - Ciudades y Comunidades Sostenibles.

La implementación de esta herramienta en nuevas regiones puede contribuir a la creación de comunidades empresariales más sostenibles al permitir una toma de decisiones más eficiente y estratégica.



Objetivo 17 - Alianzas para Lograr los Objetivos.

Este proyecto puede fomentar alianzas y colaboración, ya que se requiere coordinación con diversas partes interesadas para expandir con éxito la herramienta en nuevas regiones.