Agencia Espacial
Bases de Datos
TP6

CURSO 2023/2024

— MEMORIA —

Pablo Larraz (841983)
Óscar Brizuela (820773)
Alejandro Benedí (843826)

Índice

Introducción

En este trabajo se aborda la necesidad de seleccionar y evaluar distintos Sistemas Gestores de Bases de Datos (SGBDs) en el contexto específico de la Agencia Espacial Europea (*ESA*). La *ESA*, como una de las principales organizaciones dedicadas a la exploración espacial, enfrenta el desafío de gestionar y almacenar vastas cantidades de información relacionada con sus diversas misiones y proyectos espaciales. Gracias a los conocimientos adquiridos en la asignatura y una ardua investigación se pretende realizar un análisis para identificar los SGBD más adecuados que se ajusten a las necesidades específicas de información de la *ESA*. Este proceso implica no solo considerar las ventajas y desventajas de cada SGBD, sino también evaluar su capacidad para manejar datos complejos, garantizar la integridad y seguridad de la información, y facilitar la colaboración y el acceso a los datos dentro de la organización.

Contexto de la aplicación

La Agencia Espacial Europea (*ESA*, por sus siglas en inglés), es una organización internacional fundada en 1975 y formada por 22 estados que se dedica a la exploración y al vuelo espacial. Cuenta con un presupuesto de 7790 millones de euros para el año 2024 y, a pesar de que tiene su sede principal en París, sus estructuras están muy descentralizadas por toda Europa. España se unió a esta organización en 1979.

Debido a que se trata de una organización en la que intervienen tantos gobiernos, las restricciones de presupuesto no deberían ser un gran problema *a priori*. Es evidente que también será un factor a tener en cuenta pero, como se comentará más adelante, los principales factores para elegir un SGBD adecuado serán principalmente la precisión, la integridad, la disponibilidad, la seguridad y la velocidad de transmisión con la que dicho SGBD gestiona la información.

Los programas desarrollados por la *ESA* son muy variados, pudiéndose clasificar por áreas presupuestarias. Entre estas áreas podemos destacar programas dedicados a la observación de la Tierra, lanzamiento de cohetes, vuelos espaciales humanos, telecomunicaciones y navegación, seguridad espacial o diversas misiones de sondas y satélites, entre otros. Además, la *ESA* también difunde periódicamente prototipos, conceptos y diseños en forma de artículos con el objetivo de que sirvan de base para misiones futuras.

Por tanto, la *ESA* se enfrenta a la compleja tarea de gestionar y almacenar una gran cantidad de información relacionada con sus diversas misiones y proyectos espaciales. Esta información abarca desde datos científicos recopilados durante misiones de exploración hasta registros de ingeniería y logística necesarios para el funcionamiento de las operaciones espaciales. Algunos ejemplos de datos recopilados, almacenados y gestionados por la *ESA* son contenido multimedia (principalmente imágenes, aunque también pueden almacenar audio y vídeo), datos de telemetría, datos orbitales, características y magnitudes físicas de cuerpos espaciales, datos geoespaciales y características de naves, cohetes y sondas.

Es importante tener en cuenta que la mayoría de esta información responde a operaciones de lectura, pues los datos son sensorizados de algún modo (dependiendo de la misión o el proyecto) y enviados a un centro donde se analizarán y se almacenarán para su estudio. No obstante, sí que puede haber operaciones de escritura en aquellas misiones en las que haya que, por ejemplo, ajustar las órbitas de sondas a distancia, arreglar fallos técnicos o manipular el movimiento de *robots*, además de la propia escritura de información en los servidores y almacenes de la *ESA*. Es muy frecuente también la adición de metadatos para procurar la preservación en el largo plazo, además de para hacer cierta información pública a través de portales en internet.

La *ESA* opera en un entorno altamente dinámico y exigente, donde la precisión, la fiabilidad y la disponibilidad de los datos son críticas para el éxito de sus misiones. Además, la colaboración entre equipos distribuidos en diferentes ubicaciones geográficas y la necesidad de compartir datos de manera segura y eficiente dentro de la organización son aspectos fundamentales para garantizar el progreso y la eficacia en sus proyectos.
Dada la complejidad y la magnitud de los datos que maneja la *ESA*, se hace evidente la necesidad de contar con SGBDs que no solo sean capaces de manejar grandes volúmenes de datos, sino también de garantizar la integridad, la seguridad y la accesibilidad de la información. Además, la capacidad de adaptarse a diferentes tipos de datos, desde telemétricos hasta geoespaciales o científicos, y de proporcionar herramientas para análisis y colaboración, es esencial para optimizar las operaciones de la agencia.

Evaluación de los SGBDs

A continuación, se procede a analizar una serie de SGBDs en función de los parámetros antes mencionados, con el objetivo de decidir cuál es el mejor para almacenar, recopilar y gestionar la información con la que tiene que tratar la *ESA* en sus diversos programas. Es importante mencionar que no todos los programas gestionan el mismo tipo de información: algunos, como el diseño de nuevas naves espaciales o la medición de datos científicos, pueden contar con más información tabular; por el contrario, programas relacionados con el análisis de imágenes espaciales, por ejemplo, pueden tener más contenido multimedia. Por tanto, el análisis de cada SGBD se llevará a cabo teniendo en cuenta toda la posible información a gestionar en el conjunto de programas.
De esta manera, los criterios que se van a valorar para cada SGBD van a ser la escalabilidad, el rendimiento y la seguridad del SGBD, así como la fiabilidad, la disponibilidad, la integración de los datos y la flexibilidad de los datos a gestionar.

Oracle

Este SGBD es líder en el mercado, reconocido por su robustez, escalabilidad y capacidades de gestión de datos empresariales. *Oracle* destaca por su capacidad para manejar grandes volúmenes de datos y su soporte para transacciones complejas, entre otros.

Oracle también cuenta con altas medidas de rendimiento, lo que permitiría analizar grandes conjuntos de datos de manera rápida. Esto es particularmente importante en la investigación científica, sobre todo en asuntos de toma de decisiones en el caso de, por ejemplo, ajuste de órbitas de vehículos espaciales.

En cuanto a la fiabilidad y disponibilidad de los datos, *Oracle* ofrece métodos de replicación de información y tolerancia a fallos, lo que es crucial en este contexto. Por ejemplo, cuenta con la tecnología *Flashback*, que se asemeja a un botón de “Deshacer” una transacción específica en caso de errores humanos o corrupciones lógicas en vez de la necesidad de realizar *backups* constantemente. Relacionado con la disponibilidad, a través de sistemas como *RAC* (*Oracle Real Application Clusters*) o *ASM* (*Automatic Storage Management*) se lleva a cabo una asignación de recursos optimizada.

En cuanto a la integración de información, *Oracle* soporta gran interoperabilidad que le permite recopilar datos de varias fuentes además de llevar a cabo procesos *ETL*. Además, gracias a paquetes como el *ORD_DOC PL/SQL*, *Oracle* permite almacenar contenido multimedia en bases de datos orientadas a columnas mediante tipos de datos como los *BLOBS,* los *BFILES* o los tipos *ORDDoc*. Sucede lo mismo para datos de audio con el paquete *ORD_AUDIO PL/SQL*. Por otro lado, debido a la diversidad de aplicaciones que tiene la compañía, *Oracle* permite la sincronización entre varios sistemas.

Relacionado con la seguridad, *Oracle* cuenta con medidas de encriptación, autenticación, control de acceso en función de roles y usuarios y técnicas para evitar la manipulación de información por parte de, por ejemplo, agencias o gobiernos externos. Entre estas técnicas destacan las restricciones de tipos de datos o las auditorías que monitorizan las operaciones de las bases de datos.

Por último, en relación a la flexibilidad, *Oracle* cuenta con una plataforma altamente flexible que se puede acomodar a diversos tipos de datos y estructuras. Esto es crucial debido al carácter internacional de la agencia, que trabaja con investigadores e ingenieros de toda Europa de forma interdisciplinar.

Sin embargo, *Oracle* cuenta con ciertas características que presentan dudas para nuestra elección, como el hecho de ser un entorno relativamente cerrado y estar obligado a colaborar con distintos sistemas de la misma compañía. Otro punto en contra es el precio de sus ediciones a pesar de que, como ya hemos mencionado, el presupuesto de la *ESA* es muy alto, lo que le permitiría no escatimar en gastos. Según el artículo 1, la *NASA* malgastó cerca de 20 millones en la *suite* de *Oracle*.

Si se eligiera este SGBD estaríamos obligados a realizar un preciso estudio de nuestras necesidades para ajustar en lo máximo posible el gasto.

MySQL

MySQL es un SGBD relacional de código abierto, ampliamente utilizado en aplicaciones web y empresariales debido a su facilidad de uso, escalabilidad, ligereza, fiabilidad y que cuenta con una sólida comunidad de usuarios y desarrolladores dispuestos a colaborar.

El rendimiento de *MySQL* puede variar dependiendo de varios factores, como la configuración del servidor, la eficiencia del diseño de la base de datos, la optimización de consultas y el volumen de datos. En general, *MySQL* ofrece un rendimiento rápido para aplicaciones de tamaño medio o pequeño cuando el volumen de datos no es muy grande. Sin embargo, si el volumen de los datos aumenta el rendimiento se ve mermado considerablemente. *MySQL* permite implementar técnicas de indexaciones, así como particiones u otras técnicas para aumentar el rendimiento de las búsquedas que permite manejar cargas de trabajo más pesadas de manera más eficiente y proporcionar tiempos de respuesta más rápidos. Cabe destacar que *MySQL* cuenta con una extensión para gestionar datos espaciales sin perder su esencia en cuanto a ligereza del sistema (2). Sin embargo, este es más limitado que otras opciones de la competencia.

En relación con la fiabilidad y disponibilidad de los datos, *MySQL* incorpora un sistema básico de transacciones *ACID* que aporta cierta durabilidad y consistencia a la información almacenada. Además, incorpora un sistema de replicación de datos en el cual un servidor maestro replica su información a uno o a más servidores esclavo, lo que permite un mejor tiempo de respuesta y una mejor tolerancia a fallos. Esto puede ser de gran ayuda en esta caso ya que permitiría repartir la carga de trabajo en diferentes servidores entre los diferentes centros de la *ESA*.

Si nos centramos en la seguridad, este SGBD ofrece una autenticación robusta con diferentes métodos de autenticación (contraseña, certificado *SSL*, …), y provee también sistemas de encriptación de datos, (*aes_encrypt* y *aes_decrypt*) que pueden ser de gran ayuda para aumentar la seguridad de aquellos proyectos confidenciales.

Aunque este SGBD presenta grandes ventajas en cuanto a la rápida configuración para proyectos medianos, se percibe limitado para nuestros requerimientos. Carece de ciertas características avanzadas presentes en otros sistemas, su gestión de transacciones puede ser menos robusta y puede experimentar limitaciones en rendimiento con cargas extremadamente pesadas. Además, su escalabilidad vertical es limitada en configuraciones estándar, y puede requerir optimización manual para obtener el máximo rendimiento.

PostgreSQL

PostgreSQL es un SGBD de código abierto conocido por su robustez y funcionalidad avanzada. Es el segundo SGBD más utilizado del mundo, después de *Oracle*. Destaca por su capacidad para manejar grandes volúmenes de datos y cargas de trabajo complejas. Ofrece características como transacciones *ACID*, rendimiento sólido y escalabilidad, tanto horizontal como vertical, mediante replicación y particionamiento gracias a herramientas como *pgpool-II* o *CitusDB*.

En cuanto al rendimiento, las consultas analíticas complejas cuentan con buenos tiempos de respuesta debido a técnicas de indexación y otras estrategias basadas en el procesamiento paralelo. Además, la arquitectura *MVCC* (*Multiversion Concurrency Control*) permite lidiar con asuntos de concurrencia y lectura consistente sin sacrificar rendimiento. De hecho, está demostrado que el rendimiento de este SGBD es mayor cuanto mayor sea el volumen de datos, cosa que para nosotros juega a favor.

En relación a la disponibilidad y fiabilidad de los datos, *PostgreSQL* también cuenta con mecanismos de replicación para la tolerancia a fallos de nodos. *Patroni* es un buen ejemplo de extensión que, mediante el uso de *Python*, garantiza la alta disponibilidad de los datos en entornos distribuidos. Por otro lado, debido a que *PostgreSQL* soporta transacciones *ACID*, la durabilidad y la consistencia de la información está garantizada.

Si nos fijamos en la integración de la información, encontramos que *PostgreSQL* puede ser utilizado por multitud de lenguajes de programación a través de sus *drivers* nativos. Cuenta con soporte para multitud de tipos de datos diferentes. Además, existe una versión *NoSQL* de este SGBD para manejo de bases de datos clave/valor y documentales, lo que podría ser eficiente a la hora de lectura de datos sensorizados desde el espacio o para almacenar todos los documentos correspondientes a los proyectos de la *ESA*, por ejemplo.

Por otro lado, *PostgreSQL*, cuenta con una seguridad avanzada y una comunidad activa de usuarios y desarrolladores. Cuenta con mecanismos de seguridad como *Kerberos* y protocolos *SSL/TLS* de encriptación, además de sistemas *RBAC* (*Role-Based Access Control*).

Por último, este SGBD también ofrece flexibilidad a la hora de definir los esquemas, soportando tipos de datos para datos estructurados, semi-estructurados y no estructurados. Con él también se pueden definir funciones por parte del usuario (*UDFs*).

Por último cabe mencionar que, gracias a su comunidad, *PostgreSQL* se ha convertido en un opción por la que se decantan la mayoría de los proyectos. La gran integración con sistemas heterogéneos es un gran punto a su favor. Además, *PostgreSQL* proporciona una versión destinada a los datos geoespaciales con soporte para archivos *GIS* proveyendo funciones de análisis, transformación, importación y exportación de datos, así como una gran velocidad de procesamiento por los índices creados (3).
Como punto en contra se podría destacar la falta de una gran corporación detrás, como sí existe en el caso de *Oracle*, proporcionando un servicio técnico completo constante.

DB2

DB2, desarrollado por *IBM*, es un SGBD relacional que se destaca por su robustez y amplia adopción en el mundo empresarial. Sigue una arquitectura de modelo relacional. Con una capacidad excepcional para manejar grandes volúmenes de datos, *DB2* ofrece un rendimiento notable incluso en entornos de carga intensiva. *DB2* es ampliamente escalable tanto horizontal como verticalmente, lo que es importante a la hora de procesar grandes cantidades de datos. También ofrece sólidos mecanismos de seguridad para proteger los datos sensibles, con control de acceso a nivel de usuario y cifrado para proteger la información sensible de la *ESA*.

Las herramientas de administración incluidas facilitan tareas como configuración, monitorización, respaldo y recuperación de la base de datos. Además, al ser un producto de *IBM*, *DB2* se integra fácilmente con otras tecnologías y productos de la empresa, como *WebSphere*, *Cognos* y *Rational*, lo que contribuye a la creación de un ecosistema tecnológico coherente. Aunque es un producto propietario de *IBM*, *DB2* también ofrece compatibilidad con estándares de la industria, como *SQL* y *JDBC*, lo que facilita su interoperabilidad con otras aplicaciones y herramientas del mercado.

La integración de datos centraliza información de diversas fuentes para mejorar la coherencia y accesibilidad. Herramientas como *IBM Data Replication* permiten replicar los datos en tiempo real a diferentes servidores. Además, *DB2 Federation* y *IBM InfoSphere Federation Server* permiten consultar y combinar datos de múltiples fuentes heterogéneas como si estuvieran en una única base de datos (bases de datos federadas). Tecnologías como *Change Data Capture (CDC)* y *IBM MQ* garantizan actualizaciones en tiempo real, mejorando la toma de decisiones y la visualización de estos

Sin embargo, este SGBD presenta desventajas remarcables, como pueden ser la gran curva de aprendizaje (al igual que *Oracle*) o los requerimientos de *hardware* previos a la instalación. Para sacar el mayor beneficio a DB2 sería necesario utilizar varios sistemas y tecnologías de la compañía, lo cual añade complejidad extra si se decide integrarse con otros sistemas que no sean de la compañía. *IBM* contiene una extensión, *DB2 Spatial Extender*, que permite almacenar, gestionar y analizar datos espaciales, como coordenadas geográficas, polígonos y líneas, facilitando consultas avanzadas sobre datos geoespaciales para aplicaciones como sistemas de información geográfica, planificación urbana y logística (4).

Microsoft SQL Server

Microsoft SQL Server (o, simplemente, *SQL Server*) es un SGBD desarrollado por *Microsoft* que se utiliza ampliamente en entornos empresariales para almacenar, administrar y recuperar datos de manera eficiente. Una de las características destacadas de *SQL Server* es su escalabilidad, lo que significa que puede adaptarse tanto a pequeñas empresas como a grandes corporaciones. Ofrece opciones de escalabilidad vertical, permitiendo aumentar la capacidad de procesamiento y almacenamiento en un único servidor, así como escalabilidad horizontal, que implica distribuir la carga de trabajo entre varios servidores.

SQL Server proporciona sólidos mecanismos de seguridad para proteger los datos sensibles, incluyendo control de acceso a nivel de usuario, cifrado de datos y auditoría de actividades. Además, ofrece un conjunto completo de herramientas de administración que facilitan tareas como la configuración, monitorización, respaldo y recuperación de la base de datos.

Esta plataforma también es conocida por su integración con otras tecnologías de *Microsoft*, como el entorno de desarrollo *.NET* y el sistema operativo *Windows Server*. Esto permite una integración sin problemas con otras aplicaciones y herramientas de *Microsoft*, así como con servicios en la nube como *Azure*. Este sistema no proporciona ningún soporte especializado más que un tipo de datos para representar información sobre la ubicación física y la forma de objetos geográficos (5).

Microsoft Access

Microsoft Access es una solución de bases de datos desarrollada por Microsoft, dirigida a usuarios individuales o pequeñas empresas. Destaca por su enfoque en bases de datos de escritorio, lo que significa que está diseñado para ejecutarse en un solo equipo sin necesidad de un servidor dedicado. Su interfaz gráfica intuitiva permite a los usuarios crear y gestionar bases de datos sin requerir conocimientos avanzados de programación. Utiliza un modelo relacional para organizar los datos, pero su capacidad de manejo de grandes volúmenes de datos es limitada en comparación con sistemas más robustos como *SQL Server*. Además, carece de soporte específico para datos geoespaciales.

Su escalabilidad es limitada, ya que no está diseñado para manejar grandes cantidades de datos ni múltiples usuarios concurrentes de manera eficiente. Esto puede llevar a problemas de rendimiento y estabilidad a medida que la aplicación crece en tamaño o se utiliza por varios usuarios simultáneamente.

Otra desventaja es su limitada seguridad. Aunque ofrece características básicas de seguridad como contraseñas de base de datos y permisos de usuario, estas pueden no ser suficientes para proteger datos sensibles en entornos empresariales. Además, *Access* carece de muchas características avanzadas presentes en otros SGBDs, como la capacidad de programar procedimientos almacenados o el soporte para datos geoespaciales.

Finalmente, *Microsoft Access* está estrechamente integrado con el ecosistema *Windows*, lo que limita su portabilidad y capacidad de integración con otros sistemas operativos y tecnologías. Esto puede representar un inconveniente para organizaciones que necesitan una solución más flexible y compatible con múltiples plataformas.

Caché

Caché es reconocido por ser un SGBD altamente escalable y versátil, diseñado para aplicaciones de misión crítica que exigen un rendimiento excepcional y una disponibilidad continua. Su capacidad para manejar grandes volúmenes de datos en tiempo real, procesamiento de transacciones y análisis avanzado lo convierten en una opción popular en diversas industrias, como salud, finanzas, logística y telecomunicaciones. Al combinar tecnologías de bases de datos relacionales y orientadas a objetos, así como una arquitectura en memoria que optimiza el acceso a los datos, Caché ofrece una sólida base para aplicaciones exigentes.

Sin embargo, a pesar de sus numerosas fortalezas, también existen algunas desventajas asociadas con el uso de Caché. El costo de las licencias de Caché puede representar un desafío financiero significativo para algunas organizaciones, especialmente para aquellas con presupuestos ajustados. La adquisición de licencias puede ser prohibitiva y requerir una inversión considerable.

La utilización de un modelo de datos multidimensional y un lenguaje de programación propio, *ObjectScript*, puede generar una curva de aprendizaje pronunciada para los desarrolladores. *Caché* es desarrollado y mantenido por *InterSystems*, lo que implica una dependencia significativa de un único proveedor para soporte, actualizaciones y desarrollo futuro. Aunque Caché es una solución sólida para aplicaciones de misión crítica, puede tener dificultades para integrarse con tecnologías y herramientas más modernas y populares.

VoltDB

VoltDB es un SGBD diseñado específicamente para aplicaciones que requieren un alto rendimiento, baja latencia y escalabilidad horizontal. Utiliza una arquitectura de base de datos en memoria para ofrecer un procesamiento de transacciones extremadamente rápido y consistente. *VoltDB* se destaca por su capacidad para manejar cargas de trabajo intensivas en datos y transacciones, como aplicaciones de telecomunicaciones, servicios financieros y análisis en tiempo real. Su enfoque en la velocidad y la disponibilidad lo hace adecuado para aplicaciones de alta velocidad y baja latencia, donde la respuesta rápida es crucial.

Además, en cuanto a la disponibilidad de los datos, la compañía está anunciando grandes mejoras y un soporte continuo como se puede ver en el artículo 6 del anexo, en el que se describe el lanzamiento de la replicación de centros de datos.

En términos de seguridad, *VoltDB* implementa múltiples capas de protección para garantizar la integridad y confidencialidad de los datos. Esto incluye el cifrado de datos en reposo y en tránsito, la autenticación de usuarios y la autorización de acceso basada en roles. Además, *VoltDB* ofrece auditoría de actividades para el seguimiento y la revisión de eventos de

seguridad. Estas medidas combinadas fortalecen la seguridad de los datos y protegen contra amenazas externas e internas.

Este sistema también facilita la integración gracias a su soporte para múltiples lenguajes de programación, protocolos de comunicación estándar y conectores integrados para sistemas externos, además de ofrecer *APIs* robustas y herramientas de integración que simplifican el desarrollo de soluciones interoperables y la conexión con otros sistemas y servicios.

Apache Cassandra

Apache Cassandra es un SGBD distribuido altamente escalable diseñado para manejar grandes volúmenes de datos en un entorno distribuido con tolerancia a fallos. Utiliza una arquitectura descentralizada basada en el modelo de almacenamiento de columnas para distribuir los datos de manera uniforme a través de múltiples nodos en un clúster. Por tanto, la adición de más nodos al clúster utilizado hace que se consiga una rápida escalabilidad horizontal.

En cuanto a rendimiento, *Cassandra* es capaz de trabajar con cargas de cargas de trabajo intensivas sobre todo en escritura. Además, su motor de almacenamiento basado en un sistema de *logs* estructurado le permite alcanzar una latencia muy baja, lo que es crucial para el contexto en el que nos encontramos. Sin embargo, el uso de su lenguaje de consultas (*CQL*) no es muy flexible ni rápido, además de no permitir consultas complejas, por lo que sería necesaria una integración con *Spark*.

En relación a la disponibilidad y fiabilidad, estrechamente ligado a la forma en la que *Cassandra* escala, este SGBD es perfecto para asegurar que los datos son fiables gracias a la replicación de la información en sus nodos. Esto, a su vez, permite que los datos estén disponibles mucho más rápido y elimina la posibilidad de pérdida de datos en caso de que, como en otros SGBD, los datos estuvieran en un solo nodo. Métodos como *hites handoff* o reparaciones de lectura y escritura permiten gestionar las inconsistencias y las discrepancias entre los datos. Sin embargo, *Cassandra* no soporta transacciones *ACID*, por lo que pueden existir problemas de concurrencia y sincronización.

Cassandra cuenta con varias herramientas de integración a través de bibliotecas, *APIs* y conectores con diversos lenguajes de programación. Los tipos de datos almacenados también pueden ser muy variados, aunque no es ideal para el almacenamiento de contenido multimedia. A pesar de que sí puede almacenar tipos de datos como *BLOBS*, el tamaño de estos, por ejemplo, está limitado a 2 *GB*.

Relacionado con la seguridad, *Cassandra* ofrece mecanismos y protocolos de autenticación y autorización a través de encriptación, como prácticamente todos los SGBDs. Además, también asegura la integridad de los datos durante el tránsito a través de los nodos del clúster, evitando fugas de información en caso de atacantes en la red del clúster.

Por último, *Cassandra* ofrece mucha flexibilidad en el diseño de los esquemas y modelos de datos, permitiendo a los investigadores adaptarse a los distintos patrones y requisitos de rendimiento.

MongoDB

MongoDB es un SGBD *NoSQL* que se destaca por su flexibilidad, escalabilidad y capacidad para manejar datos semi-estructurados y no estructurados. Utiliza un modelo de documentos *JSON* (*BSON* en realidad) para almacenar los datos, lo que permite una fácil escalabilidad horizontal y una rápida iteración de desarrollo. *MongoDB* es conocido por su capacidad para manejar grandes volúmenes de datos y su capacidad de consulta flexible.

Este sistema garantiza la fiabilidad y disponibilidad de los datos en la nube. Gracias a su arquitectura distribuida y a la replicación automatizada de datos, *Atlas* ofrece una tolerancia a errores distribuida y recuperación de datos sin intervención manual siguiendo un modelo de consistencia eventual. Con clústeres configurados para mantener réplicas en múltiples zonas de disponibilidad dentro de una región, se asegura la disponibilidad constante de los datos, incluso ante interrupciones regionales o en la nube. Además, la conmutación por error automática permite una rápida recuperación en caso de fallos, con la posibilidad de configurar nodos secundarios en múltiples regiones para una protección adicional. *Atlas* también ofrece la conmutación por error entre varias nubes, asegurando la disponibilidad incluso en situaciones extremas. Respaldado por un acuerdo de nivel de servicio líder en la industria, los clústeres de *MongoDB Atlas* ofrecen una alta disponibilidad del 99.995%. Con características como copias de seguridad gestionadas, recuperación incremental de datos y movilidad de datos en la nube, *Atlas* proporciona una solución completa y confiable para las aplicaciones más importantes de las empresas.

Además, ofrece una variedad de soluciones para abordar los desafíos de seguridad y privacidad de manera efectiva. Con *Queryable Encryption*, los datos cifrados dentro de *MongoDB* permanecen seguros, incluso al compartirse con terceros. El cifrado integral cubre los datos en reposo y en tránsito, garantizando la protección de los datos en todo su ciclo de vida. Este sistema contiene una gran comunidad que ha ayudado en la integración con distintos lenguajes de programación y tecnologías. Ha facilitado enormemente su integración en distintos sistemas.

HBase

Apache HBase es una base de datos *NoSQL* distribuida y escalable que está diseñada para manejar grandes volúmenes de datos estructurados. Se ejecuta sobre *Hadoop Distributed File System (HDFS)*. *HBase* es ideal para aplicaciones que requieren acceso aleatorio a grandes conjuntos de datos, como almacenamiento de datos en tiempo real, análisis de registros y aplicaciones de *Big Data*. Ofrece alta disponibilidad y rendimiento al tiempo que garantiza la tolerancia a fallos en entornos distribuidos. Este, por sí mismo, no proporciona funcionalidades específicas para datos geoespaciales de manera nativa, ya que está

diseñado principalmente para manejar grandes volúmenes de datos estructurados de manera distribuida y escalable. Sin embargo, *HBase* puede integrarse con otras tecnologías y herramientas para trabajar con datos geoespaciales.

Por ejemplo, se puede utilizar *HBase* junto con bibliotecas o *frameworks* especializados en análisis geoespacial, como *Apache Hadoop GIS* o *GeoMesa*, para almacenar y consultar datos geoespaciales de manera eficiente. Estas herramientas permiten indexar y realizar consultas espaciales sobre datos almacenados en *HBase*, lo que facilita el análisis y la visualización de información geográfica en entornos distribuidos y de gran escala.

Solución adoptada

Tras analizar detenidamente las diferentes opciones, se ha considerado que *PostgreSQL* e *IBM DB2* son las opciones más adecuadas para las necesidades específicas de la Agencia Espacial Europea (*ESA*).

Se han descartado los siguientes SGBDs debido, principalmente, a la falta de prestaciones en determinados ámbitos:

- *Microsoft Access:* cuenta con grandes limitaciones en cuanto a escalabilidad y concurrencia, además de que tiene una gran dependencia con la plataforma *Windows*. En el caso de que los sistemas de la ESA fueran diferentes y variados, no se podría utilizar este SGBD para almacenar los datos.

- *MySQL:* a pesar de su capacidad para trabajar con grandes volúmenes de datos y del soporte para datos geoespaciales, la pérdida de rendimiento a medida que aumenta la cantidad de estos datos dificulta el análisis a través de consultas.

- *HBase*: requiere de una complejidad en cuanto al diseño y mantenimiento del sistema que, junto con su limitación en la escalabilidad, nos hace descartarlo.

- *Caché*: su prolongada curva de aprendizaje e insuficiente soporte a nivel de comunidad y garantía de uso nos proponen un inconveniente en su uso. Cuenta también con falta de integración con lenguajes de programación y bajo rendimiento al trabajar con grandes volúmenes de datos.

- *VoltDB:* el poco uso en la industria de este SGBD para aplicaciones que requieren de análisis complejo hace que *VoltDB* sea descartado por cuestiones de falta de documentación y confianza para este contexto.

- *Cassandra*: al ser *NoSQL*, su consistencia eventual y rendimiento para algunas consultas complejas, a la vez que su complejidad de gestión de datos críticos, son un obstáculo para su elección.

- *Oracle:* a pesar de que cuenta con un grandísimo soporte de la comunidad, además de integración con una cantidad ingente de tipos de datos y medidas de seguridad, el hecho de que sea cerrado (y que, por tanto, depende exclusivamente de productos de su misma *suite*) ha hecho que se descarte este SGBD. Por otro lado, el alto coste del sistema también ha influido en esta decisión, a pesar de que la ESA cuente con un gran presupuesto. Otros SGBDs de su mismo nivel (como *PostgreSQL*) cuentan con mayores opciones de personalización e integración con sistemas externos.

- *MongoDB:* a pesar de que las BBDD no relacionales suelen ser utilizadas principalmente en contextos en los que se requiere más lectura que escritura (como puede ser el caso de la ESA), el hecho de que haya que almacenar datos complejos y altamente estructurados ha provocado el descarte de esta opción.

Por tanto, los SGBDs que han sido finalmente seleccionados han sido *PostgreSQL* e *IBM DB2*.

PostgreSQL ofrece una combinación de robustez, rendimiento, seguridad y flexibilidad que lo convierte en una opción más que válida para satisfacer las necesidades de gestión de datos de la Agencia Espacial Europea. De la misma manera, la escalabilidad, facilidad de configuración y personalización de *DB2* también hace de este sistema una opción prácticamente idónea para el contexto en el que nos encontramos.

En resumen, ambos sistemas cumplen con creces los requisitos para los parámetros analizados, además de contar con soporte para datos geoespaciales, por lo que la ESA podría decantarse por uno de ellos para almacenar y analizar toda la información que gestiona.

Esfuerzos invertidos

Tareas / nombre	Pablo Larraz (841983)	Oscar Brizuela (820773)	Alejandro Benedi (843826)
Investigación sobre la ESA y su contexto	1 hora	3 horas	1 hora
Análisis de SGBDs	5 horas		
Solución adoptada	1,5 horas		
Realización de presentación	2 horas	0,5 horas	2 horas
Horas totales	**9,5 horas**	**10 horas**	**9,5 horas**

Anexo

1. https://www.ciospain.es/liderazgo--gestion-ti/la-nasa-malgasto-15-millones-de-dolares-en-licencias-de-oracle-no-utilizadas
2. https://mappinggis.com/2019/09/mysql-y-gis-usa-mysql-como-una-base-de-datos-espacial/
3. https://geoinnova.org/blog-territorio/postgresql-ypostgis-que-son-y-como-se-relacionan/
4. https://www.ibm.com/docs/es/psfoa/1.0.0?topic=tasks-spatial-data-db2-spatial-extender
5. https://learn.microsoft.com/es-es/sql/relational-databases/spatial/spatial-data-sql-server?view=sql-server-ver16
6. https://www.prnewswire.com/il/news-releases/voltdb-lanza-la-replicacion-de-centros-de-datos-cruzados-sin-perdida-activa-n--843739429.html
7. https://blogs.oracle.com/maa/post/the-matchless-reliability-of-oracle-database-high-availability-part-2
8. https://docs.oracle.com/en/database/oracle/oracle-database/12.2/imurg/introduction-to-oracle-multimedia.html
9. https://stackoverflow.com/questions/47161185/storing-media-files-in-cassandra
10. https://docs.datastax.com/en/cassandra-oss/2.1/cassandra/dml/dml_about_hh_c.html
11. https://patroni.readthedocs.io/en/latest/
12. https://es.overleaf.com/articles/base-de-datos-nosql-caso-de-estudio-postgres-como-solucion-nosql/qjwggdysdbwv