

Trabajo Práctico 6

BASES DE DATOS 2

Hayk Kocharyan
757715@unizar.es

Pedro Tamargo Allué
758267@unizar.es

Jesús Villacampa Sagaste
755739@unizar.es

Juan José Tambo Tambo
755742@unizar.es

28 de mayo de 2020

Índice

1. Enunciado	2
2. SGBD	2
2.1. Oracle	2
2.2. MySQL	3
2.3. PostgreSQL	4
2.4. IBM DB2	5
2.5. Microsoft SQL Server	6
2.6. Microsoft Access	6
2.7. InterSystems Caché	7
2.8. VoltDB	8
2.9. Apache Cassandra	8
2.10. MongoDB	9
2.11. Apache HBase	10
3. Conclusiones	11
Referencias	12

1. Enunciado

Para la realización del trabajo se ha elegido el problema del *Gobierno de Aragón*. El problema proporcionado en el enunciado dice lo siguiente:

“El Gobierno de Aragón quiere gestionar información turística de la región para ofrecerla a través de un sitio web donde los usuarios podrán hacer comentarios y recomendaciones.”

Por lo tanto la expansión del enunciado es la siguiente:

Se pretende desarrollar una base de datos para almacenar la información turística de la región de Aragón. Se llevará acabo a través de un sitio web mediante la cual, los usuarios puedan consultar y realizar recomendaciones de puntos turísticos, así como comentar otras. También, este sistema deberá permitir almacenar usuarios. Ya que la página web permitirá realizar publicaciones, recomendaciones y comentarios se almacenarán en la base de datos. Se deberá garantizar el acceso concurrente a la base de datos debido a que múltiples usuarios pueden realizar publicaciones y comentarios de manera simultánea. Respecto a la garantía de la calidad del servicio primaremos la consistencia de los datos frente a la disponibilidad de la misma.

En relación a la estimación del trafico web, será de unos **10.000** usuarios a la semana, **15.000** en alta temporada.

Por lo mencionado anteriormente, este sistema gestor deberá garantizar una buena interoperabilidad con los lenguajes de programación utilizados en este proyecto.

Si fuese necesario se podría utilizar la infraestructura disponible en las ciudades de Huesca y Zaragoza.

Por ello, en base a estos requisitos buscaremos un sistema gestor *OLTP* (*Online Transaction Processing*) ya que un gestor OLAP está enfocado para en el análisis. Respecto al volumen de datos, el sistema no se prevee que almacene grandes volúmenes de datos, por lo tanto la escalabilidad no es un factor primordial.

Según los datos del presupuesto de turismo proporcionados por el presupuesto de la comunidad de Aragón del año 2020, turismo cuenta con una cantidad de 13.100 € (trece mil cien euros).[15]

Con estas condiciones suponemos que vamos a contar con un presupuesto similar.

Dado que el Gobierno de Aragón es una entidad pública se buscará el uso de un SGBD consolidado, y que ofrezca buen soporte en el tiempo.

2. SGBD

2.1. Oracle

Oracle es un sistema gestor de bases de datos multimodelo (soporta modelo relacional y modelo objeto relacional), perteneciente a la compañía *Oracle Corporation* [89] . En 1977 fue fundada la compañía *SDL* (*Software Development Laboratories*), este *SGBD* proviene de un proyecto financiado por la *CIA* para diseñar un sistema especial de bases de datos con código clave “*Oracle*” [88]. Los fundadores de *SDL* habían leído un artículo en la revista *IBM research* donde se describía una versión preliminar de *SQL* basado en el artículo de *E. F. Codd*¹, donde propone el *modelo relacional*. En 1984, *SDL* adopta el nombre de *Oracle Corporation*.

En su versión 19, la instalación de este *SGBD* se puede realizar *on-premise* (en los servidores de la empresa) o en el *cloud* de *Oracle*. Si elegimos la opción *on-premise*, los sistemas operativos en los que es posible instalar este *SGBD* son: *Windows*, *Linux* y *Solaris* [54].

Oracle es una de las bases de datos líderes del mercado en bases de datos operacionales [26]. Cuenta con mucha documentación disponible desde su página web² y además cuenta con una comunidad muy activa en la resolución de incidencias y dudas en páginas como *StackOverflow* [28].

El dominio de aplicación de este *SGBD* se encuentra desde el desarrollo de aplicaciones hasta servicios de almacenes de datos [53].

El sistema de licenciamiento de este *SGBD* depende del número de usuarios y el número de procesadores que tenga el servidor donde se ha de instalar. Para cada “*Processor License*” se aplica el “*Core factor*”³ y ese es el número de “*Processor Licenses*” necesarios. Para el cálculo de “*Named User Licenses*” se aplicará el máximo entre:

¹https://es.wikipedia.org/wiki/Edgar_Frank_Codd

²<https://docs.oracle.com/en/database/oracle/oracle-database/19/lnpls/index.html>

³<http://www.oracle.com/us/corporate/contracts/processor-core-factor-table-070634.pdf>

- $ProcessorLicenses * CoreFactor * NumUserMinimun$
- $NumUsuarios + NumDispositivos$

Donde, *NumUserMinimun* se corresponde con el número de usuarios mínimos por cada “*Processor License*” [56]. Los precios de “*Processor License*” y “*Named User Licenses*” en la versión *Enterprise Edition* se corresponden con: 47500\$ y 950\$ [57]. El resto de funcionalidades no añadidas, como las de almacenes de datos, se adquieren por separado.

En el tema de escalabilidad, *Oracle* cuenta con *RAC (Real Application Cluster)*, una tecnología que permite escalar una base de datos a un cluster de servidores ejecutando multiples instancias sobre una misma base de datos de manera transparente al usuario, aprovechando los recursos del clúster. [55]

El control de la concurrencia en este *SGBD* se realiza mediante los mecanismos de bloqueo del mismo para garantizar la consistencia de los datos entre transacciones. *Oracle* se encarga de gestionar los mecanismos de bloqueo automáticamente, de tal forma que no es necesario que el usuario escriba el bloqueo de forma explícita.[52] Los mecanismos de bloqueo utilizados por este *SGBD* se dividen en dos grupos, bloqueos exclusivos y bloqueos compartidos. [2]

Para el desarrollo con este *SGBD* se puede utilizar el driver *ODBC* (o *JDBC* si se usa *Java*) desde la propia página de la compañía.

El uso de herramientas de soporte complementarias como puede ser *DBever*⁴ y otras opciones de apoyo al desarrollo es posible configurando (si fuera necesario) el driver de conexión a la base de datos.

Las copias de seguridad se pueden realizar utilizando la herramienta *RMAN*. Esta herramienta es específica para las bases de datos *Oracle* y permite realizar copias de seguridad sobre las bases de datos y realizar una vuelta a una versión estable si disponemos de una copia de seguridad [51].

El soporte técnico de este *SGBD* se adquiere por separado, y se relaciona con cada uno de los productos adquiridos. [57]

2.2. MySQL

MySQL es un sistema gestor de bases de datos relaciones de código abierto. Es un software gratuito y libre bajo los términos de GNU⁵. MySQL también se encuentra bajo licencias cerradas como es la de Oracle, quien adquirió a Sun Microsystems, uno de los primeros propietarios de MySQL. Tras esta adquisición surgió MariaDB otro RDBMS⁶ que surgió debido a la privatización de MySQL.

MySQL es la base de datos de software libre más popular del mundo para distribuir con bajo coste económico aplicaciones web o e-commerce, procesamiento de transacción online y base de datos integrada confiables. Es una base de datos integrada, segura para transacción, de conformidad con las propiedades ACID, rollback, recuperación de falla a nivel de fila. MySQL proporciona facilidad de uso, posibilidad de dimensionamiento y alto desempeño, así como un paquete completo de drivers de base de datos y herramientas visuales para ayudarles a los desarrolladores y DBAs a crear y gestionar sus aplicaciones MySQL.

Entre las funcionalidades podemos destacar por ejemplo la capacidad de redimensionamiento para atender a las demandas de cargas de datos y usuarios en crecimiento exponencial. También se dispone de clusters de replicación con auto-recuperación para mejorar la escalabilidad, el desempeño y la disponibilidad.

Otra de las funcionalidades destacables es el poder cambiar el esquema online para satisfacer exigencias de negocio que se cambian constantemente. Este gestor también ofrece herramientas de monitorización (a nivel aplicación y consumo de recursos) y de autenticación mediante PAM y encriptación. Otra característica fundamental es el acceso tanto SQL como NoSQL para realizar consultar y operaciones simples y rápidas de Clave-Valor.

Cuando hablamos de escalabilidad MySQL ofrece esta solución a través de diferentes métodos. Permite replicación sobre diferentes nodos para distribuir la carga de trabajo. También trata con IDs en las transacciones para facilitar la conmutación por error y la creación de topologías de replicación en anillo. Respecto a los suscriptor, son multiprocesos para garantizar la disponibilidad, además que son cash-safe, es decir, permiten recuperarse ante fallos. Finalmente ofrece la herramienta de particionado MySQL Fabric para distribuir los datos en diferentes servidores.[49]

⁴<https://dbeaver.io/>

⁵General Public License

⁶Relational Database Manager System

Otro de los pilares fundamentales es la concurrencia. MySQL ofrece un mecanismo de hilos con un buen *throughput*. Provee un hilo por conexión y este hilo no se reasigna hasta que la conexión no acaba, además este método es escalable. Con este mecanismo es suficiente para la mayoría de webs, sin embargo, si se va a tratar con mucho tráfico es recomendable usar la versión enterprise que es ofertada con un *Pool de Threads*. Con esta última funcionalidad el gestor nos provee grupos de thread en función de la demanda, es útil una vez sobrepasamos las 256 conexiones simultáneas a la base de datos.

El control de la concurrencia se realiza en dos niveles, primero a nivel servidor y segundo a nivel almacenamiento. Cuenta con bloqueos de lectura y escritura con gran transparencia. Destacar que tiene bloqueos a nivel de tablas y de filas, esta última ofrece la mayor concurrencia posible, sin embargo, es la menos eficiente. [48]

MySQL está disponible en diferentes SO como son: Linux, Windows, Oracle Solaris, macOS, FreeBSD entre muchas otras. Además ofrece conectores para todos los lenguajes clave de desarrollo que incluyen PHP, Perl, Python, Java, C, C++, C#, Ruby, Node.js y Go. Cabe destacar que si optamos por la opción de MySQL enterprise podemos acceder a una amplia gama de herramientas de Oracle.

En cuanto a comunidad cuenta con un buen apoyo, en StackOverflow, existen mas de medio millón de consultas respecto a MySQL.

MySQL cuenta con el soporte de Oracle. Como MySQL está dedicado mas al ámbito de web es fundamental para Oracle para que pueda completar su catálogo, de hecho, el gestor aumentó su popularidad tras la adquisición y el mantenimiento de Oracle.

En cuanto a empresas que usand MySQL destacan los grande como Twitter , Facebook, YouTube, PayPal, Yahoo, Linkedin, Ticketmaster...

Para terminar vamos a hablar del coste. Si optamos por la versión estandar, el coste es de 2000 euros anuales, en cambio, si optamos por la versión enterprise este precio se eleva a 5000 euros anuales la versión de MySQL con hasta 4 sockets en el servidor.[50]

2.3. PostgreSQL

PostgreSQL, también llamado *Postgres*, es un sistema de gestión de bases de datos relacional orientado a objetos y de código abierto. Como muchos otros proyectos de código abierto, el desarrollo de *PostgreSQL* no es manejado por una empresa o persona, sino que es dirigido por una comunidad de desarrolladores que trabajan de forma desinteresada, altruista, libre o apoyados por organizaciones comerciales. Dicha comunidad es denominada *PGDG*⁷. [76]

PostgreSQL está publicado bajo *PostgreSQL License*, una licencia *Open Source* , similar a *BSD*⁸ o *MIT*⁹. [59]

Uno de los puntos fuertes de *Postgres* es el cumplimiento de las propiedades ACID¹⁰ y su gestión de la concurrencia. Aseguran que una lectura nunca bloquea escrituras y viceversa. Esto se consigue a través de un mecanismo llamado *MVCC*¹¹. Cada transacción en *Postgres* obtiene una ID de transacción llamada *XID*. Esto incluye transacciones de una sola declaración, como insertar, actualizar o eliminar, así como envolver explícitamente un grupo de declaraciones juntas a través de BEGIN-COMMIT. Cuando comienza una transacción, *Postgres* incrementa un *XID* y lo asigna a la transacción actual.[16]

Independientemente de la plataforma y la arquitectura que usemos, *PostgreSQL* está disponible para los diferentes SO, *Unix*, *Linux* y *Windows*, en 32 y 64 bits. Ésto hace de *PostgreSQL* un sistema multiplataforma y también hace que sea más rentable con instalaciones a gran escala. Además tiene más de 20 años de desarrollo activo y en constante mejora. No se han presentado nunca caídas de la base de datos. Ésto es debido a su capacidad de establecer un entorno de alta disponibilidad gracias a *Hot-Standby*¹². [61]

Postgres está diseñado para ambientes de alto volumen, lo que hace que la velocidad de respuesta en inserciones o actualizaciones pueda parecer lenta en comparación con bases de datos de pequeño tamaño.[63]

Otro aspecto muy importante a tener en cuenta a la hora de elegir *SGBD* es el soporte que proporcionan. En el caso de *PostgreSQL* no cuenta con soporte en línea o telefónico. Sin embargo, cuenta con foros oficiales donde

⁷PostgreSQL Global Development Group

⁸https://es.wikipedia.org/wiki/Licencia_BSD

⁹https://es.wikipedia.org/wiki/Licencia_MIT

¹⁰Atomicity, Consistency, Isolation y Durability (Atomicidad, Consistencia, Aislamiento y Durabilidad en castellano)

¹¹Multi Version Concurrency Control

¹²Hot-Standby nos permite que los clientes puedan realizar consultas de solo lectura mientras que los servidores están en modo de recuperación o espera. Así podemos hacer tareas de mantenimiento o recuperación sin bloquear completamente el sistema[62]

usuarios exponen sus dudas y son los propios usuarios de la comunidad los que responden. Cabe resaltar que esta comunidad de usuarios es una de las más activas del mercado. Además, también ofrece soporte empresarial como *EnterpriseDB*¹³ o *TodoPostgreSQL*¹⁴. [63]

Postgres es uno de los gestores con más soporte y recorrido en el mercado, y por tanto, cuenta con una herramienta administración llamada *pgAdmin* (desde 2016 en la versión *pgAdmin4*), que facilita la gestión y administración de bases de datos ya sea mediante instrucciones SQL o con ayuda de un entorno gráfico. Permite acceder a todas las funcionalidades de la base de datos; consulta, manipulación y gestión de datos, incluso opciones avanzadas como manipulación del motor de replicación *Slony-I*¹⁵. A parte de este entorno de escritorio, también existen otras herramientas como *psql* (cliente de consola) o *PhpPgAdmin* (entorno web). [58]

2.4. IBM DB2

IBM DB2 es un gestor de bases de datos relacional orientada a empresas capaz de gestionar datos estructurados y no estructurados en entornos públicos, privados y locales. [12] Actualmente es una marca comercial, propiedad de *IBM*, aunque posee una versión gratuita, *DB2 Express-C* [10], que permite desarrollar aplicaciones con las funcionalidades básicas de *DB2*.

Entre sus principales características destacan la capacidad de manejar objetos de hasta 2 GB, definición de funciones y datos/tipos por parte del usuario, mecanismos para garantizar la integridad referencial, *SQL* recursivo¹⁶, tratamiento de archivos multimedia, procesamiento paralelo, commit en dos fases y *backup* on-line y offline. Además permite la conexión directa de una aplicación escrita en *Java* con la base de datos utilizando *JDBC*, pudiendo ser una base local o remota. [13]

El gestor utiliza *pureXML*¹⁷ para almacenar documentos en formato *XML*, para realizar operaciones y búsquedas dentro del mismo e integrarlo con búsquedas relacionales. [68] De esta manera se puede realizar consultas de tipos de datos no tradicionales. Además utiliza un sistema multiprocesador *SMP* simétrico y un sistema de procesador paralelo masivo, lo que permite su alto rendimiento.

Se trata de un sistema altamente escalable al dar soporte a volúmenes de datos de *petabytes* y acelerar las consultas, pudiendo escalar desde un *laptop* a *clusters* de servidores empresariales paralelos. [9]. Asimismo, *DB2* ofrece múltiples opciones de despliegue (híbrido, *on premise*, cloud), lo que da soporte a un amplio rango de cargas de trabajo.

DB2 utiliza un sistema de *commit-rollback* diferente a otros gestores. Todos cambios en la base se almacenan en el *Log Buffer*, el cual escribe en disco cuando se realiza un *commit*. Un *commit* no es correcto hasta que se ha copiado toda información en disco. [14]

Las plataformas que admite el gestor son *AIX*, *HP-UX*, *UNIX*, *Linux*, *Solaris* y *Windows*, además de poseer una versión para *Z/OS*¹⁸, la cual está optimizada para *SOA*, *CRM* y almacenamiento de datos. [8]

Por último, el gestor ofrece una serie de herramientas de administración, gestión de rendimiento, recuperación, réplicas y aplicaciones. La mayoría de estas herramientas proporcionan una interfaz gráfica de usuario (*GUI*) y una interfaz *ISPF*¹⁹ que permiten al usuario realizar la mayoría de tareas de *DB2* de forma interactiva. [11]

¹³<https://www.enterprisedb.com>

¹⁴<https://todopostgresql.com/>

¹⁵*Slony-I* es un sistema asíncrono de replicación maestro-esclavo para PostgreSQL, que proporciona soporte para conexión en cascada y conmutación por error. Asíncrono significa que cuando una transacción de base de datos se ha comprometido con el servidor maestro, todavía no se garantiza que esté disponible en esclavos. La conexión en cascada significa que las réplicas se pueden crear (y actualizar) a través de otras réplicas, es decir, no necesitan conectarse directamente al maestro. [65]

¹⁶<https://geeks.ms/jirigoyen/2009/05/22/recursividad-con-sql-server/>

¹⁷Función de almacenamiento XML nativo.

¹⁸Sistema actual de las computadoras de *IBM*

¹⁹Interactive System Productivity Facility

2.5. Microsoft SQL Server

Es un gestor de bases relacional desarrollado por *Microsoft*. Este gestor dispone de diferentes ediciones como es el *Enterprise*, que a parte de proporcionar el gestor, añade una serie de servicios y herramientas. La edición *standard* que incluye el gestor y una serie de servicios mas básicos y con limitación en el soporte de nodos del cluster. Otra de las versiones a destacar es *Azure*, que ofrece el servicio de este gestor pero como un *PaaS*. La arquitectura que usan se basa en una capa externa que actúa como interfaz. Todas las operaciones que pueden ser invocadas en *SQL Server* se comunican a través de *Data Stream (TDS)*, este protocolo de nivel aplicación se encargará de transferir los datos entre cliente y servidor. [86]

Microsoft SQL Server almacena tablas con columnas tipadas, al igual que permite el uso y definición de *UDTs*. El lenguaje de consulta que se usa no es SQL sino una variante, *T-SQL*. Este expande SQL incluyendo programación *procedimental*.

Una de las ventajas que tiene este gestor es la cantidad de servicios que ofrece, como por ejemplo, *servicio de Broker*, *servicio de análisis*, *SQLCMD* (permite ejecución de consultas desde terminal), y un *servicio de replicación*. Respecto al servicio de replicación incluye replicación transaccional, replicación por mezcla (cambios tanto en publicador como suscriptor, y luego se sincronizan de manera cruzada), y replicación por instantánea (se realiza una copia de la base y los suscriptores replican estos datos).

En cuanto a soporte de lenguajes, ofrece drivers para muchos, desde *Java* y *C++* hasta *Ruby*, en este enlace podemos encontrar todos. [37]

Como es de esperar es un gestor que garantiza las propiedades *ACID* en las transacciones. Para garantizar una alta disponibilidad nos ofrece replicación, envíos de log, *mirroring*, *clustering* y una funcionalidad llamada *alwaysON Availability Groups*. La replicación usa una serie de *jobs*, donde interactúan un *publicador(publisher)*, un *distribuidor*, y un *suscriptor(subscriber)*. El log se lleva a cabo a través de tareas que realizan transacciones del log con copias de la base desde un *primario* a un *secundario*, al igual que con el servicio de *mirroring*. El clustering consiste en que los diferentes servicios en funcionamiento realizan almacenamiento en diferentes putos compartidos que son altamente fiables. Por último el servicio de *AlwaysON* consiste en realizar copias en diferentes *grupos de disponibilidad* que creamos, estos grupos realizan la conmutación por error conjuntamente.

Respecto a concurrencia ofrece tres tipos de soluciones para mejorarla. La primera es *solo lectura*, permite leer pero no actualizar o borrar, esto brinda una alta concurrencia. Sin embargo, el gestor puede que realice bloqueos a nivel de filas para reforzar ciertas operaciones, también es capaz de realizar bloques de lectura en lugar de escritura. La segunda es a través del *locking*, el cursor usa el menor nivel de bloqueos que garantiza poder garantizar actualizaciones y borrados en el conjunto de datos. Esta opción mencionada ofrece baja concurrencia pero mayor operabilidad. Por último tenemos, la *concurrencia optimista*, consiste en actualizar o borrar solo si tras la última lectura no han sido cambiados. Esta última solución ofrece gran concurrencia pero no asegura las modificaciones. [33]

Respecto al coste de las licencias, la versión *Enterprise* tiene su coste de licencia por núcleos, su precio inicial es de 13.000 euros en adelante. La versión *Standard* tiene un coste de 3.000 euros y al igual que el anterior es en función al número de núcleos. Destacar que ambos comienzan con un total de dos núcleos. Para terminar, destacar que dispone de una licencia *Express* para bdd rápidas con coste gratuito, y de una licencia de desarrollador con limitaciones de uso y almacenamiento también sin coste.

2.6. Microsoft Access

Microsoft Access es un sistema de gestión de bases de datos incluido en el paquete ofimático denominado *Microsoft Office*, sucesor de *Embedded Basic*. *Access* es un gestor de datos que utiliza los conceptos de bases de datos relacionales y pueden manejarse por medio de consultas e informes. Está adaptado para recopilar datos de otras utilidades como *Excel* o *SharePoint*. A partir del 22 de septiembre de 2015 la versión más nueva es *Access 2016*. [64]

Los requisitos para la utilización de este gestor son un sistema operativo *Windows 10*, un procesador de 2 núcleos a 1'6GHz, una memoria RAM de 2GB, un espacio libre en disco duro de 4 GB y una cuenta Microsoft [36]. El precio para adquirir una licencia para 1 PC de *Access* es de 135€. [35]

Las bases de datos *access* son útiles para pequeñas y medianas empresas o departamentos individuales, ya que cuentan con una capacidad limitada. Cualquier sector cuyo uso vaya más allá de las 2 GB descubrirá sus limitaciones [27]. Para evitar esta limitación de tamaño, se puede crear vínculos a tablas de otras bases de datos de *Access*. Puede crear vínculos a tablas de varios archivos de base de datos, cada uno de los cuales puede tener hasta

2 gigabytes de tamaño. [34]

Microsoft Access ofrece consultas parametrizadas²⁰. Estas consultas y tablas *Access* se pueden referenciar desde otros programas como *VB6*²¹ y *.NET* a través de *DAO* o *ADO*. [60] También está el formato de base de datos *Jet*²², que puede contener la aplicación y los datos en un archivo. Esto hace muy conveniente para distribuir la aplicación completa a otro usuario, que puede ejecutarla en entornos desconectados. [60]

El límite técnico de las bases de datos *access* es de 255 usuarios al mismo tiempo, pero el límite real es entre 10 y 80 usuarios dependiendo del tipo aplicación. [27]

2.7. InterSystems Caché

Caché es un SGBD comercial que pertenece a *InterSystems*. Está destinado principalmente a servicios de salud, administración, banca, finanzas, gobierno y otros sectores parecidos. *Caché* permite usar la base de datos SQL y orientado a objetos, al igual que permite a los desarrolladores manipular directamente las estructuras de datos almacenados en arrays multidimensionales. *MUMPS* es el lenguaje que inspiró a *Caché*. [69]

Caché deriva gran parte de su potencial a su arquitectura única. El core de *Caché* es una BBDD que provee los servicios más comunes - incluyendo almacenamiento, concurrencia, transacciones, administración de procesos - necesario para un gestor en condiciones. Sin embargo, donde encontramos lo mejor es en su variedad de modelos de datos, este combina objetos, tablas relacionales y estructuras multidimensionales, todos accediendo a los mismos datos que se describen un única vez. Pero *Caché* no es solo un SGDB, también incluye un servidor de aplicaciones con capacidades de POO. [29]

Caché proporciona diferentes lenguajes de script para creación y acceso de datos (*Caché ObjectScript*, *Caché Basic*...). Soporta lenguajes como Java y C++, conectores como JDBC y ODBC. *Caché* incluye mas funcionalidades a parte de un SGBD, incorpora herramientas para desarrollar aplicaciones Web basadas en navegador a través de *InterSystem Zen* o *Caché Server Pages*. También tiene soporte para aplicaciones que no se basan en web, por ejemplo, aquellas cuya IU está programada en *Java* o *C++* brindando grandes rendimientos si se implementa el resto de la aplicación con *Caché*. También es compatible con otras tecnologías como Angular a través de la *API Rest*.

InterSystem Caché no es muy abierto cuando hablamos de SO. Tiene licencias para unos SOs y hardware específicos que se pueden encontrar en esta web. Además, nos ofrecen dos tipos de licencias, *Independiente de Plataforma* o *Específico para Plataforma*, destacar que si se adquiere la específica y se cambia de SO, se aplican cargos.

En relación a las características, *Caché* incluye un sistema de bloqueos muy potente que hace poder tener acceso concurrente a la BBDD. Ofrece operaciones atómicas sin necesidad de bloqueos a nivel aplicación, esto lo realiza asignando ids únicos a cada objeto o fila. También posee la capacidad de realizar *locks* a nivel lógico, es decir, no bloquea grandes cantidades de datos mientras se realizan actualizaciones. Los bloqueos tiene gran granularidad y es capaz de realizar bloqueos a nivel fila u objeto. Por último, también dispone de bloqueos distribuidos. [30]

Este gestor ofrece una buena escalabilidad debido a que usa un protocolo propio, *ECP* (*Enterprise Cache Protocol*). Este protocolo permite que diferentes máquinas distribuidas usen unos a otros como bases de datos. Lo más destacable de este protocolo es que a nivel aplicación no hay cambios, es transparente por lo que para las aplicaciones es como acceder de forma local. Otro punto a destacar de este proceso, es que si un cliente necesita un dato de otro servidor de datos, el servidor local lo obtiene, se lo proporciona y lo cachea para futuros accesos. Tener datos cuyo acceso es habitual de manera cacheada, hace que el tráfico en red se reduzca y de esta manera se brinde un servicio mucho mejor.

Cuando se habla de disponibilidad, *Caché* ofrece grandes ventajas. La primera de ellas es que hace uso de *Write-Image Journaling* que consiste en una técnica en dos fases. Al hacer una operación primero se escriben de memoria a un "transitional journal" local y luego en la base de datos, de esta manera si la segunda falla, se dispone de una copia en el journal. Es tolerante a fallos de hardware y hasta fallos eléctricos.

Podemos añadir otras funcionalidades que usa para mejorar la disponibilidad como es el *Database Mirroring* (que consiste en replicar en un disco a parte en tiempo real), *ECP distribuido*, y los *Failover Clusters*. [31]

²⁰Se denominan consultas paramétricas a aquellas consultas, normalmente de selección, que bajo un campo por el que se desea establecer un criterio de filtrado, en vez de implicar a valores constantes que hacen que dicha consulta al ser ejecutada siempre realice "lo mismo", se implique a campos (especificados entre corchetes []) a los cuales se les asignan valores diferentes para cada ejecución de la consulta (normalmente serán solicitados al usuario mediante una ventana en pantalla). [1]

²¹Visual Basic 6.0

²²Jet es un motor de bases de datos que permitiría mantener los archivos .mdb a través de ODBC [74]

Por último, del presupuesto no tenemos mucha información. Nos pusimos en contacto con el servicio de ventas de *Caché* pero fue tarea complicada obtener el precio, ya que nos hacían llamar a un servicio de venta y no nos ofrecían documentación.

Sin embargo, nos avisaron de que *Caché* está evolucionando a *Iris*, un servicio que ofrece gestor de bases de datos, análisis y servicios de machine learning.

2.8. VoltDB

VoltDB es una base de datos en memoria diseñada por Michael Stonebraker, Sam Madden y Daniel Abadi. Es un *RDBMS*²³ compatible con ACID que utiliza una arquitectura de nada compartido²⁴. Incluye tanto ediciones empresariales como comunitarias. La edición comunitaria está licenciada bajo la *GNU Affero General Public License*. [77]

VoltDB es una base de datos relacional *NewSQL*²⁵ que admite acceso *SQL* desde procedimientos almacenados Java precompilados. La unidad de transacción es el procedimiento almacenado, escrito en Java intercalado con *SQL*. *VoltDB* se basa en la partición horizontal hasta el subproceso de hardware individual para escalar, k-safety (replicación síncrona) para proporcionar una alta disponibilidad y una combinación de instantáneas continuas y registro de comandos para una mayor durabilidad (recuperación de fallos). [77]

Una gran desventaja de este *SGBD* es que solo se ejecuta en *Linux* y *Mac*, y por tanto para su utilización en *Windows* haría falta un despliegue en *contenedores docker* o la utilización de máquinas virtuales. [3]

El punto clave de *VoltDB* es su velocidad. Puede procesar hasta 45 veces más transacciones por segundo que otros gestores tradicionales como *MySQL*, *Oracle* y *PostgreSQL*. *VoltDB* procesó 53.000 transacciones por segundo (TPS) frente a las 1.155 de otros *DBMS* sobre hardware idéntico. [4] Debido a esto, se debe ejecutar en servidores ricos en memoria multi-core y aunque esté disponible de manera gratuita bajo la licencia *GPL* ofrecen un precio de suscripción anual es de 15 dólares por año para una configuración de cuatro servidores.

VoltDB permite operacionalizar *modelos ML*. *VoltDB* convierte automáticamente el *modelo ML* en un proceso ejecutable como una función definida por el usuario (UDF), que puede implementarse en un entorno de producción. El modelo integrado ingiere / entrena continuamente datos nuevos / históricos para mantenerse actualizado y relevante. El *ML* en la base de datos le permite tomar decisiones inteligentes y procesables en tiempo real con una latencia muy baja en la transmisión de datos. [5]

VoltDB se ejecuta en un entorno distribuido en contenedores. Hasta ahora, orquestar bases de datos *SQL* en *Kubernetes*²⁶ había sido muy desafiante, ya que los sistemas de bases de datos operacionales tienen estado y no pueden ser acelerados o reducidos en cualquier momento. Los desarrollos recientes aseguran la orquestación de contenedores con *Kubernetes*, convirtiendo muchas tareas tediosas y complejas en algo tan simple como un archivo de configuración declarativo, y permitiendo una implementación continua y frecuente. [5]

2.9. Apache Cassandra

Cassandra es un sistema gestor de bases de datos *NoSQL (Not Only SQL)*, con almacenamiento basado en columnas. Fue creado desarrollado por *Avinash Lakshman*, uno de los autores de *Amazon's Dynamo*²⁷ por *Facebook* como un proyecto para mejorar la búsqueda en la bandeja de entrada de la plataforma. En 2008, *Facebook* liberó a *Cassandra* como un proyecto open-source en *Google Code*²⁸. En Marzo de 2009, *Apache* acogió este proyecto, transformándolo en un proyecto de *Apache Incubator* (proyecto en la que los proyectos open-source se pueden convertir en nuevos proyectos de alto nivel de la fundación *Apache* [18]). En Febrero de 2010 se graduó como un

²³Sistema de gestión de bases de datos relacionales

²⁴Una arquitectura de nada compartido es una arquitectura de computación distribuida en la que cada solicitud de actualización es satisfecha por un solo nodo (procesador / memoria / unidad de almacenamiento). La intención es eliminar la contención entre nodos. Los nodos no comparten (acceso independiente) memoria o almacenamiento [67]

²⁵*NewSQL* es una clase de sistemas de gestión de bases de datos relacionales que buscan proporcionar la escalabilidad de los sistemas *NoSQL* para cargas de trabajo de procesamiento de transacciones en línea (OLTP) mientras mantienen las garantías ACID de un sistema de base de datos tradicional. [75]

²⁶*Kubernetes* es un sistema de código libre para la automatización del despliegue, ajuste de escala y manejo de aplicaciones en contenedores que fue originalmente diseñado por Google y donado a la Cloud Native Computing Foundation. Soporta diferentes entornos para la ejecución de contenedores, incluido Docker. [70]

²⁷*Amazon's Dynamo* un conjunto de técnicas que juntas pueden formar un sistema de almacenamiento estructurado de alta disponibilidad o un depósito de datos distribuidos. [84]

²⁸*Google Code* fue una plataforma que proveía control de versiones, seguimiento de incidencias para proyectos open-source. Actualmente se encuentra en modo de solo lectura desde Agosto de 2015. [85]

proyecto de alto nivel. [82]

Cassandra, al estar programado sobre *Java*, es multiplataforma, es decir, se puede instalar sobre cualquier sistema operativo siempre que cuente con *Java* (recomiendan la última versión de *Java* 8). La opción más habitual es utilizar un servidor con SO *Linux*. [24]

Este *SGBD* pertenece a *Apache Software Foundation*, una compañía sin ánimo de lucro que desde 1999 ha desarrollado software open-source. *Cassandra* se graduó como un proyecto de *Apache Incubator* en 2010, y desde entonces ha liberado 11 versiones. Sigue manteniendo de forma activa 3 de ellas. [23]
El dominio de aplicación de *Cassandra* se corresponde con el análisis de datos en entornos distribuidos donde se puede sacrificar la *consistencia* de los datos frente a la disponibilidad y a la tolerancia a particiones de red (Teorema CAP). Se centra en la disponibilidad y la escalabilidad lineal, es decir, se mejora el rendimiento en relación al número de nodos que se encuentran en la red [82]. Esta orientado a entornos donde no existe un nodo maestro, es decir, entornos totalmente distribuidos donde los nodos se comunican por una red *P2P*.

Este software se distribuye bajo una licencia *Apache 2*, es una licencia open-source que implica que se puede utilizar para cualquier propósito de forma gratuita. La licencia *Apache 2* es una licencia permisiva ya que la modificación del software distribuido con la misma no tiene por qué mantener esta licencia, exceptuando las partes que no hayan sido modificadas. [83]

A diferencia de los gestores de bases de datos relacionales, *Cassandra* no sigue las propiedades de transacciones *ACID*, pero en su lugar ofrece transacciones *atómicas*, *aisladas* y transacciones *persistentes* con *consistencia eventual* o modificable, es decir, el usuario puede decidir el nivel de exigencia que quiere fijar sobre la consistencia de una transacción. [7]

La herramienta de realización de consultas es *CQLSH*, una herramienta de línea de comandos para interactuar con *Cassandra* utilizando *CQL* (*Cassandra Query Language*) [21], un lenguaje para la realización de consultas y manipulación de los datos muy similar a *SQL*. [82]

Existen drivers (*ODBC* y *JDBC* (*Java*)) con los cuales podemos realizar el acceso a la base de datos utilizando distintos lenguajes de programación, tales como: *Java*, *Python*, *NodeJS* (*JavaScript*), *Dart*, *C++*... [22]

Para la administración de los nodos de un clúster, *Cassandra* provee la herramienta *Nodetool*. Esta herramienta brinda la capacidad de realizar tareas de administración sobre los nodos (añadir un nodo a un clúster, modificar la configuración de un nodo...) tanto como las herramientas necesarias para realizar una monitorización sobre el mismo. [82]

El soporte técnico de este sistema no lo realiza *Apache*, si no que recae sobre empresas de terceros. Por ejemplo, *DataStax*²⁹ ofrece servicios profesionales, servicio técnico y despliegues de este *SGBD*. [25]

2.10. MongoDB

MongoDB es un sistema de base de datos *NoSQL*, orientado a documentos y de código abierto. Los datos se almacenan en estructuras de tipo *BSON*³⁰ con un esquema dinámico, facilitando y agilizando la integración de los datos en determinadas aplicaciones.

Se puede obtener de forma gratuita bajo la licencia de código abierto *AGPL*³¹. También ofrece una licencia comercial que incluye distintas características como integración con *SASL*, *LADP* o *Kerberos*. [72]

Entre las principales características del gestor [71], se puede destacar la capacidad de consultas *ad hoc*, con búsqueda por campos, consulta de rangos y expresiones regulares; Su capacidad de indexar cualquier campo en un documento³² y crear índices secundarios, aumentando la eficiencia de las consultas [43]; Soporta la replicación *primario-copia*, siendo cada grupo de los mismos un *replica set*³³; La posibilidad de escalar horizontalmente, utilizando *Sharding*³⁴, permitiendo añadir nuevos nodos al sistema mientras está en funcionamiento. [46]; Mediante *GridFS*³⁵, *MongoDB* puede ser utilizado como un sistema de archivos con balanceo de carga y tolerante a fallos, aprovechando las características de *MongoDB*. Por último, permite realizar consultas con *JavaScript* [44], enviándolas a la base de datos para que sean ejecutadas.

Mongo no garantizaba las propiedades *ACID* multidocumento hasta su versión 4.0 (entteriormente se cumplían para un solo documento), aunque posee las siguientes limitaciones [73]:

²⁹<https://www.datastax.com/>

³⁰Especificación similar a *JSON*

³¹Licencia pública general de Affero de *GNU*

³²Los índices son almacenados en una estructura Árbol-B

³³Un *replica set* es un grupo de procesos *mongod* que mantienen en mismo conjunto de datos. [45]

³⁴Método para distribuir datos a través de múltiples máquinas

³⁵Especificación para almacenar y consultar archivos que superan el límite de tamaño de documentos *BSON* de 18 MB. [41]

- Bloquea la base a nivel de documento en las escrituras, impidiendo realizar operaciones de escritura concurrentes en el mismo documento.
- El gestor retorna tras escribir en al menos una réplica, lo que no basta para garantizar la durabilidad ni la verificabilidad.[39]
- Si supera los 100 GB de datos, aparecen problemas de rendimiento.[40]

El gestor posee drivers oficiales para diferentes lenguajes de programación: *C*, *C++*, *C#*, */.NET*, *Java*, *JavaScript*, *Node.js*, *Perl*, *PHP*, *Python*, *Ruby*, *Scala*, *Delphi* y *C++ Builder*.

Un apartado importante de MongoDB es *Mongo Shell*[47], una interfaz interactiva en *JavaScript* mediante la cual se puede consultar y modificar datos e incluso realizar operaciones de administración. Puede ser utilizado en *Windows*, *macOS* o *Linux*, lo que convierte a *MongoDB* en un gestor multiplataforma.

MongoDB proporciona una serie de productos en la nube para poder almacenar y gestionar instancias del mismo, como lo son *MongoDB Stitch*, *MongoDB Atlas Data Lake* o *MongoDB Cloud Manager*, entre otros.[38]

Además, el gestor posee una gran serie guías para cada una de sus funcionalidades en su página web[42], así como un soporte técnico al que se contactar mediante correo o vía telefónica³⁶.

2.11. Apache HBase

HBase es sistema gestor de bases de datos *NoSQL*, con almacenamiento basado en columnas. Fue desarrollado por *Powerset*³⁷ para procesar grandes cantidades de datos para realizar las búsquedas con lenguaje natural. Actualmente es un proyecto de alto nivel de *Apache Software Foundation*. [79]

Apache HBase provee la capa superior del sistema de ficheros de *Apache Hadoop*³⁸, está inspirado en *Google BigTable*³⁹ y el sistema de ficheros de *Google*.

Al estar programado con *Java*, *Apache HBase* es una solución multiplataforma, aunque desde la documentación del *SGBD* no recomiendan utilizar *Windows* en entornos de producción. Uno de los prerequisites para utilizar este sistema es tener *JDK* instalado (recomiendan *JDK8*). [20]

Apache HBase es un *SGBD* orientado al procesamiento de grandes cantidades de datos, es decir, está orientado al análisis de datos (*OLAP*).

Este *SGBD* pertenece a *Apache Software Foundation*, una compañía sin ánimo de lucro que desde 1999 ha desarrollado software open-source. *HBase* es un proyecto de alto nivel de *Apache*. Su última versión estable se liberó en abril de 2006. [79]

HBase se distribuye bajo una licencia *Apache 2*, que, como se ha explicado anteriormente, permite la utilización del software para cualquier propósito de forma gratuita. [83]

Sobre la escalabilidad, este *SGBD* está orientado a un entorno distribuido. Al igual que *Apache Cassandra*, está orientado a la escalabilidad lineal, es decir, el rendimiento del sistema mejora de manera lineal respecto al número de nodos del sistema. [17]

A diferencia de los *SGBD* relacionales, *HBase* provee transacciones *ACID* a nivel de fila. Para ello hace uso de mecanismos de bloqueo por filas (*row level lock*) y un sistema de control de la concurrencia multiversión [6][32]. Es un mecanismo que utilizan los *SGBD* relacionales para permitir la concurrencia, es decir, el sistema gestor almacena distintas versiones del objeto con el que se está trabajando mientras alguien esté trabajando con alguna de ellas. [87]

Para el acceso a este *SGDB*, desde la página web, nos proporcionan una serie de conectores. También, podemos realizar el acceso utilizando *SQL* existe *Apache Phoenix*⁴⁰, que proporciona una capa de acceso a *HBase* así como el acceso con drivers *JDBC* y e integración con herramientas de inteligencia de negocio [79].

Para la realización de tareas como las copias de seguridad, *Apache HBase* provee herramientas integradas para ello. Esta copia de seguridad se guardará automáticamente en la ubicación especificada, pudiendo ser el propio *HDFS* donde está desplegada la máquina, otro *HDFS* de otro centro de datos, o un servicio en cloud compatible con *HDFS*. [19]

³⁶<https://www.mongodb.com/contact>

³⁷*Powerset* es una compañía norteamericana que desarrolló un motor de búsqueda con lenguaje natural.[81]

³⁸*Apache Hadoop* es un framework de software que permite a las aplicaciones trabajar con miles de nodos y petabytes de datos.[78]

³⁹*BigTable* es un sistema gestor de bases de datos creado por *Google* con las características de ser distribuido, de alta eficiencia y con licencia privativa.[80]

⁴⁰*Apache Phoenix* es una base de datos de código abierto, masivamente paralela con soporte *OLTP* para *Hadoop* utilizando *HBase* como servicio de almacenamiento. [66]

3. Conclusiones

Entre los gestores analizados, descartaríamos la utilización de todos aquellos *NoSQL*, ya que una de sus propiedades más atractivas es la escalabilidad horizontal. Dado que no se prevee tener un número de accesos al portal muy elevado, se ha considerado la prioridad de la propiedad de consistencia ante la disponibilidad. Por lo tanto, se estudiarán las opciones de bases de datos centralizadas.

En cuanto a *VoltDB*, su ámbito de actuación son las bases de datos *OLAP*, debido a su velocidad y mecanismos de Inteligencia Artificial (modelos *ML*). Por lo tanto, no se adecua a los requisitos del sistema.

De los gestores restantes, *PostgreSQL* es un gestor *Open Source* diseñado para trabajar con grandes volúmenes de datos, siendo ineficiente para bases de datos pequeñas. A priori no se tiene una previsión acerca del volumen de datos que contendrá nuestra base de datos.

InterSystems Caché es utilizado en un ámbito donde existen diferentes bases de datos que necesitan una interconexión entre ellas, como es el caso de la banca. Por ello se descarta su uso.

En lo referente a *Microsoft Access* al ser un gestor monousuario, que sólo se puede instalar en *Windows* y un tamaño de fichero máximo de 2 *GB*, se descarta ya que desconoce el volumen de datos.

Entre las opciones restantes encontramos a *Microsoft SQL Server*, *Oracle* y *MySQL*. *SQL Server* y *Oracle* se podrían adaptar a los requisitos de nuestro sistema pero su precio excede el presupuesto disponible de la dirección general de turismo.

Tras el análisis realizado anteriormente, se ha llegado a la conclusión de que se seleccionará *MySQL* como sistema gestor debido a las siguientes razones. En primer lugar, se puede resaltar su precio, ya que se dispone de la versión *Open Source* o en su defecto, la versión con soporte de *Oracle* que se ajusta con creces al presupuesto. En segundo lugar se puede destacar su alta compatibilidad con *Oracle*, uno de los gestores utilizados por el Gobierno de Aragón. Otro de los puntos a destacar es que su comunidad es muy amplia, lo que proporciona un buen soporte sin llegar a adquirir servicios adicionales. Respecto a páginas similares que hagan uso de este gestor, se tiene el caso de *Twitter* y *Facebook*, dos empresas de gran envergadura. Cabe destacar el buen soporte de este sistema ante la concurrencia, escalabilidad y el soporte en los sistemas operativos más utilizados en este ámbito.

Referencias

- [1] Consultas parametrizadas. https://www.adrformacion.com/knowledge/ofimatica/consultas_parametricas_en_access.html. Consultado el 29/04/2020.
- [2] Data concurrency and consistency. <https://docs.oracle.com/en/database/oracle/oracle-database/19/cncpt/data-concurrency-and-consistency.html#GUID-1D60EFCC-03F4-4A04-B099-1B4DE5D02C47>. Consultado el 26/04/2020.
- [3] Desventajas y ventajas voltdb. https://nanopdf.com/download/principales-sgbd_pdf. Consultado el 29/04/2020.
- [4] Velocidad de voltdb. <https://itblogpress.wordpress.com/2014/02/25/voltdb-bases-de-datos-45-veces-mas-rapidas-que-los-tradicionales-sistemas-mysql-y-oracle/>. Consultado el 29/04/2020.
- [5] Voltdb. <https://www.voltdb.com/why-voltdb>. Consultado el 29/04/2020.
- [6] G. Chanan. Apache hbase internals: Locking and multiversion concurrency control. https://blogs.apache.org/hbase/entry/apache_hbase_internals_locking_and, Enero 2013. Consultado el 29/04/2020.
- [7] DataStack. About transactions and concurrency control. https://docs.datastax.com/en/cassandra-oss/2.1/cassandra/dml/dml_about_transactions_c.html. Consultado el 27/04/2020.
- [8] DB2. Db2 - compatibilidad. <https://www.ibm.com/support/pages/system-requirements-ibm-db2-linux-unix-and-windows>. Consultado el 29/04/2020.
- [9] DB2. Db2 - escalabilidad. <https://www.eltiempo.com/archivo/documento/MAM-698231>. Consultado el 29/04/2020.
- [10] DB2. Db2 - express-c. https://www.ibm.com/support/knowledgecenter/SSEPGG_9.7.0/com.ibm.db2.luw.qb.server.doc/doc/r0054460.html. Consultado el 29/04/2020.
- [11] DB2. Db2 - herramientas. https://www.ibm.com/support/knowledgecenter/es/SSEPEK_10.0.0/intro/src/tpc/db2z_imtools.html. Consultado el 29/04/2020.
- [12] DB2. Db2 - introduction. <https://www.ibm.com/es-es/products/db2-database>. Consultado el 29/04/2020.
- [13] DB2. Db2 - jdbc. https://publib.boulder.ibm.com/tividd/td/ITIM/SC32-1147-03/es_ES/HTML/im451inst_ux_ws34.htm. Consultado el 29/04/2020.
- [14] DB2. Db2 - propiedades acid. <https://www.ibm.com/support/pages/commit-db2-luw-database>. Consultado el 29/04/2020.
- [15] G. de Aragón. Ingresos y gastos. <https://www.aragon.es/documents/20127/28841092/Ingresos+y+Gastos+2020.pdf/4bd3d1ee-5036-065a-3311-64a7198bed74?t=1578398918892>, 2020. Consultado el 02/05/2020.
- [16] Devcenter. Mvcc. <https://devcenter.heroku.com/articles/postgresql-concurrency>. Consultado el 28/04/2020.
- [17] A. S. Foundation. Apache hbase - features. <https://hbase.apache.org/index.html>. Consultado el 28/04/2020.
- [18] A. S. Foundation. The apache incubator. <https://incubator.apache.org/>. Consultado el 27/04/2020.
- [19] A. S. Foundation. Backup and restore. <https://hbase.apache.org/book.html#backuprestore>. Consultado el 29/04/2020.
- [20] A. S. Foundation. Basic prerequisites. <https://hbase.apache.org/book.html#basic.prerequisites>. Consultado el 27/04/2020.
- [21] A. S. Foundation. The cassandra query language (cql). <https://cassandra.apache.org/doc/latest/cql/index.html#cql>. Consultado el 27/04/2020.
- [22] A. S. Foundation. Client drivers. https://cassandra.apache.org/doc/latest/getting_started/drivers.html#client-drivers. Consultado el 27/04/2020.

- [23] A. S. Foundation. Downloading cassandra. <https://cassandra.apache.org/download/>. Consultado 27/04/2020.
- [24] A. S. Foundation. Installing cassandra. https://cassandra.apache.org/doc/latest/getting_started/installing.html#installing-cassandra. Consultado el 27/04/2020.
- [25] A. S. Foundation. Thirdpartysolution. <https://cwiki.apache.org/confluence/display/CASSANDRA2/ThirdPartySupport>, Julio 2019. Consultado el 27/04/2020.
- [26] Gartner. Magic quadrant for operational database management systems. <https://www.gartner.com/doc/reprints?id=1-1XT9MEFJ&ct=191125&st=sb>, Noviembre 2019. Consultado el 26/04/2020.
- [27] informaticaparatunegocio. Capacidad access. <https://www.informaticaparatunegocio.com/blog/ventajas-desventajas-las-bases-datos-access/>. Consultado el 29/04/2020.
- [28] S. Insights. Developer survey results 2019. <https://insights.stackoverflow.com/survey/2019#technology--databases>, 2019. Consultado del 26/04/2020.
- [29] InterSystems. Cache technology guide. <http://www.intersystems.com/wp-content/uploads/sites/11/CacheTechnologyGuide.pdf>.
- [30] InterSystems. Transactional multidimensional storage. https://cedocs.intersystems.com/latest/csp/docbook/Doc.View.cls?KEY=GIC_dbengine#GIC_dbengine_storage.
- [31] InterSystems. Write image journaling and recovery. https://cedocs.intersystems.com/latest/csp/docbook/DocBook.UI.Page.cls?KEY=GCDI_wij.
- [32] Q. Landing. Multiversion concurrency control in hbase. <https://hellokangning.github.io/en/post/multiversion-concurrency-control-in-hbase/>, Diciembre 2017. Consultado el 29/04/2020.
- [33] Microsoft. Cómo obtener una licencia de sql server. <https://www.microsoft.com/es-es/sql-server/sql-server-2019-pricing#ft3>.
- [34] Microsoft. Especificaciones access. <https://support.office.com/es-es/article/especificaciones-de-access-0cf3c66f-9cf2-4e32-9568-98c1025bb47c>. Consultado el 29/04/2020.
- [35] Microsoft. Microsoft access licencia. https://www.microsoft.com/es-es/p/access/cfq7ttc0k7q8?=&ef_id=CjwKCAjwqJ_1BRBZEiwAv73uwNScKHTjdGey6LznNrbv_YoYDtv_cQXDIQNLLegF7gcGVZtUZCPmVRoCj58QAvD_BwE%3aG%3as&OCID=AID2000750_SEM_xUz8JKCM&MarinID=sxUz8JKCM%7c431595526780%7cms+access%7ce%7cc%7c%7c62233861395%7ckwd-26962656&lnkd=Google_0365SMB_App&gclid=CjwKCAjwqJ_1BRBZEiwAv73uwNScKHTjdGey6LznNrbv_YoYDtv_cQXDIQNLLegF7gcGVZtUZCPmVRoCj58QAvD_BwE&activetab=pivot:overviewtab. Consultado el 28/04/2020.
- [36] Microsoft. Microsoft access requisitos. https://www.microsoft.com/es-es/p/access/cfq7ttc0k7q8?=&ef_id=CjwKCAjwqJ_1BRBZEiwAv73uwNScKHTjdGey6LznNrbv_YoYDtv_cQXDIQNLLegF7gcGVZtUZCPmVRoCj58QAvD_BwE%3aG%3as&OCID=AID2000750_SEM_xUz8JKCM&MarinID=sxUz8JKCM%7c431595526780%7cms+access%7ce%7cc%7c%7c62233861395%7ckwd-26962656&lnkd=Google_0365SMB_App&gclid=CjwKCAjwqJ_1BRBZEiwAv73uwNScKHTjdGey6LznNrbv_YoYDtv_cQXDIQNLLegF7gcGVZtUZCPmVRoCj58QAvD_BwE&activetab=pivot:regionofsystemrequirementstab. Consultado el 28/04/2020.
- [37] Microsoft. Concurrency types. <https://docs.microsoft.com/en-us/sql/odbc/reference/develop-app/concurrency-types?view=sql-server-ver15>, Enero 2019.
- [38] MongoDB. Mongoddb - cloud. <https://docs.mongodb.com/cloud/>. Consultado el 29/04/2020.
- [39] MongoDB. Mongoddb - durability problem. <https://hackingdistributed.com/2013/01/29/mongo-ft/>. Consultado el 29/04/2020.
- [40] MongoDB. Mongoddb - escalability problem. <https://jaxenter.com/mongoddb-mocked-after-posting-100gb-scaling-checklist-106822.html>. Consultado el 29/04/2020.
- [41] MongoDB. Mongoddb - gridfs. <https://docs.mongodb.com/manual/core/gridfs/>. Consultado el 29/04/2020.
- [42] MongoDB. Mongoddb - guides. <https://docs.mongodb.com/guides/>. Consultado el 29/04/2020.
- [43] MongoDB. Mongoddb - indexes. <https://docs.mongodb.com/manual/indexes/>. Consultado el 29/04/2020.

- [44] MongoDB. Mongoddb - javascript. <https://docs.mongodb.com/manual/core/server-side-javascript/index.html>. Consultado el 29/04/2020.
- [45] MongoDB. Mongoddb - mongod processes. <https://docs.mongodb.com/manual/reference/program/mongod/#bin.mongod>. Consultado el 29/04/2020.
- [46] MongoDB. Mongoddb - sharding. <https://docs.mongodb.com/manual/sharding/>. Consultado el 29/04/2020.
- [47] MongoDB. Mongoddb - shell. <https://docs.mongodb.com/manual/mongo/>. Consultado el 29/04/2020.
- [48] MySQL. Ficha técnica mysql enterprise edition. <https://www.mysql.com/why-mysql/white-papers/ficha-tecnica-mysql-enterprise-edition/>.
- [49] MySQL. Mysql enterprise edition product guide. <https://www.oreilly.com/library/view/high-performance-mysql/9780596101718/ch01.html>. concurrencia y lock.
- [50] MySQL. Para obtener precios. <https://shop.oracle.com/apex/product?p1=MySQL>.
- [51] Oracle. Backing up the database. <https://docs.oracle.com/en/database/oracle/oracle-database/19/bradv/backing-up-database.html#GUID-030BA0C5-832E-42A1-A63A-5E2D768BABB9>. Consultado el 26/04/2020.
- [52] Oracle. Data concurrency and consistency. https://docs.oracle.com/cd/B19306_01/server.102/b14220/consist.htm#i5702. Consultado el 26/04/2020.
- [53] Oracle. Oracle 19. <https://www.oracle.com/es/database/technologies/>. Consultado el 26/04/2020.
- [54] Oracle. Oracle database 19c install and upgrade. <https://docs.oracle.com/en/database/oracle/oracle-database/19/install-and-upgrade.html>. Consultado el 26/04/2020.
- [55] Oracle. Oracle real application clusters (rac). <https://www.oracle.com/es/database/technologies/rac.html>. Consultado el 26/04/2020.
- [56] Oracle. Oracle software licensing basics. <https://www.oracle.com/a/ocom/docs/corporate/oracle-software-licensing-basics.pdf>, Enero 2020. Consultado el 26/04/2020.
- [57] Oracle. Oracle technology global price list. <https://www.oracle.com/assets/technology-price-list-070617.pdf>, Marzo 2020. Consultado el 26/04/2020.
- [58] PGADMIN. Pgadmin. <https://www.pgadmin.org/>. Consultado el 28/04/2020.
- [59] PostgreSQL. Postgresql license. <https://www.postgresql.org/about/licence>. Consultado el 28/04/2020.
- [60] tecnologías información. Especificaciones access. <https://www.tecnologias-informacion.com/access.html>. Consultado el 29/04/2020.
- [61] TodoPostgres. Alta disponibilidad. <https://todopostgresql.com/alta-disponibilidad-postgresql/>. Consultado el 28/04/2020.
- [62] TodoPostgres. Hot-standby. <https://todopostgresql.com/tecnica-hot-standby>. Consultado el 28/04/2020.
- [63] TodoPostgres. Ventajas y desventajas. <https://todopostgresql.com/ventajas-y-desventajas-de-postgresql/>. Consultado el 28/04/2020.
- [64] WIKI. Microsoft access. https://es.wikipedia.org/wiki/Microsoft_Access. Consultado el 28/04/2020.
- [65] WIKI. Sloni-i. <https://en.wikipedia.org/wiki/Slony-I>. Consultado el 28/04/2020.
- [66] Wikipedia. Apache phoenix. https://en.wikipedia.org/wiki/Apache_Phoenix. Consultado el 29/04/2020.
- [67] Wikipedia. Arquitectura de nada compartido. https://en.wikipedia.org/wiki/Shared-nothing_architecture. Consultado el 29/04/2020.
- [68] Wikipedia. Db2 - purexml. <https://es.wikipedia.org/wiki/PureXML>. Consultado el 29/04/2020.
- [69] Wikipedia. Intersystems cache. https://en.wikipedia.org/wiki/InterSystems_Cache/.
- [70] Wikipedia. Kubernetes. <https://es.wikipedia.org/wiki/Kubernetes>. Consultado el 29/04/2020.

- [71] Wikipedia. Mongodb - características principales. https://es.wikipedia.org/wiki/MongoDB#Caracter%C3%ADsticas_principales. Consultado el 29/04/2020.
- [72] Wikipedia. Mongodb - licencias y soportes. https://es.wikipedia.org/wiki/MongoDB#Concesi%C3%B3n_de_licencias_y_soporte. Consultado el 29/04/2020.
- [73] Wikipedia. Mongodb - principales limitaciones. https://es.wikipedia.org/wiki/MongoDB#Principales_limitaciones. Consultado el 29/04/2020.
- [74] Wikipedia. Motor jet. https://en.wikipedia.org/wiki/Microsoft_Jet_Database_Engine. Consultado el 29/04/2020.
- [75] Wikipedia. Newsql. <https://en.wikipedia.org/wiki/NewSQL>. Consultado el 29/04/2020.
- [76] Wikipedia. Postgresql. <https://es.wikipedia.org/wiki/PostgreSQL>. Consultado el 28/04/2020.
- [77] Wikipedia. Voltdb. <https://en.wikipedia.org/wiki/VoltDB>. Consultado el 29/04/2020.
- [78] Wikipedia. Apache hadoop. https://es.wikipedia.org/wiki/Apache_Hadoop, Agosto 2019. Consultado el 27/04/2020.
- [79] Wikipedia. Apache hbase. https://es.wikipedia.org/wiki/Apache_HBase, Octubre 2019. Consultado el 27/04/2020.
- [80] Wikipedia. Bigtable. <https://es.wikipedia.org/wiki/BigTable>, Julio 2019. Consultado el 27/04/2020.
- [81] Wikipedia. Powerset (company). [https://en.wikipedia.org/wiki/Powerset_\(company\)](https://en.wikipedia.org/wiki/Powerset_(company)), Diciembre 2019. Consultado el 27/04/2020.
- [82] Wikipedia. Apache cassandra. https://es.wikipedia.org/wiki/Apache_Cassandra, Abril 2020. Consultado el 27/04/2020.
- [83] Wikipedia. Apache license. https://es.wikipedia.org/wiki/Apache_License#Condiciones_de_la_licencia, Enero 2020. Consultado el 27/04/2020.
- [84] Wikipedia. Dynamo (storage system). [https://en.wikipedia.org/wiki/Dynamo_\(storage_system\)](https://en.wikipedia.org/wiki/Dynamo_(storage_system)), Febrero 2020. Consultado el 27/04/2020.
- [85] Wikipedia. Google developers - google code. https://en.wikipedia.org/wiki/Google_Developers#Google_Code, Abril 2020. Consultado el 27/04/2020.
- [86] Wikipedia. Microsoft sql server - architecture. https://en.wikipedia.org/wiki/Microsoft_SQL_Server#Architecture, Abril 2020.
- [87] Wikipedia. Multiversion concurrency control. https://es.wikipedia.org/wiki/Multiversion_concurrency_control#Ejemplo, Febrero 2020. Consultado el 29/04/2020.
- [88] Wikipedia. Oracle corporation. https://es.wikipedia.org/wiki/Oracle_Corporation, Febrero 2020. Consultado el 26/04/2020.
- [89] Wikipedia. Oracle database. https://en.wikipedia.org/wiki/Oracle_Database, Enero 2020. Consultado el 26/04/2020.