**Tarea sobre Base de Datos**

**Giam Martínez Ced. 8-779-1599**

**Programación II Profesor: Leonardo Esqueda**

La importancia de conocer sobre base de datos consiste en tener a mano la información en forma de registros para su búsqueda posterior y análisis. Es por eso que hoy en día, su uso es fundamental tanto para las organizaciones como para el uso diario. Cada día se generan millones de gigabytes en datos, por lo que aprender los principios de base de datos es fundamental para manejar los datos, acceder a ellos de manera practica para su uso.

En esta investigación se abarcará los distintos tipos de base de datos y sus ventajas.

Función de una base de datos:

Todas las BD ordenan y clasifican los datos para luego estar dispuesta para su consulta y análisis. Esto es en sí la indexación como la ordenación de una cantidad de datos de acuerdo a un criterio en común para una búsqueda fácil.

En función del tipo de base que se tenga, existen distintos tipos de software especiales para la gestión de base de datos que permiten almacenar y acceder de forma rápida a los datos estructurados alojados en ellos.

Tipos de BBDD

Una base de datos se clasifica según su estructura, contexto y utilidad según su necesidad.

Se tocara en este ensayo el tipo de base de datos clasificados en Base de Datos relacionados y No relacionados.

Relacionales (SQL)

Este modelo se basa fundamentalmente en establecer **relaciones o vínculos** entre los datos, imaginando una tabla aparte por cada relación existente con sus propios registros y atributos.

Los principales Sistemas gestores de bases de datos relacionales (**SGBD SQL**) actualmente son:

**MySQL**

Es el sistema gestor de bases de datos relacional por excelencia.

Es un SGBD **multihilo y multiusuario** utilizado en la gran parte de las páginas web actuales. Además es el más usado en aplicaciones creadas como software libre.

Se ofrece bajo la GNU GPL aunque también es posible adquirir una licencia para empresas que quieran incorporarlo en productos privativos (Desde la compra por parte de Oracle se está orientando a este ámbito empresarial).

Las principales **ventajas** de este Sistema Gestor de Bases de datos son:

* Facilidad de uso y gran rendimiento
* Facilidad para instalar y configurar
* Soporte multiplataforma
* Soporte SSL

La principal **desventaja** es la escalabilidad, es decir, no trabaja de manera eficiente con bases de datos muy grandes que superan un determinado tamaño.

**MariaDB**

**Este SGBD es una derivación de MySQL que cuenta con la mayoría de características de este e incluye varias extensiones.**

**Nace a partir de la adquisición de MySQL por parte de Oracle para seguir la filosofía Open Source y tiene la ventaja de que es totalmente compatible con MySQL.**

Entre las principales **características** de este Sistema Gestor de Bases de datos se encuentran:

* Aumento de motores de almacenamiento
* Gran escalabilidad
* Seguridad y rapidez en transacciones
* Extensiones y nuevas características relacionadas con su aplicación para Bases de datos NoSQL.

No tiene desventajas muy aparentes salvo algunas pequeñas incompatibilidades en la migración de MariaDB y MySQL o pequeños atrasos en la liberación de versiones estables.

SQLite

Más que un Sistema Gestor de bases de datos como tal, SQLite es una **biblioteca** escrita en C **que implementa un SGBD** y que permite transacciones sin necesidad de un servidor ni configuraciones.

Es una biblioteca utilizada en multitud de aplicaciones actuales ya que es **open source** y las consultas son muy eficientes.

Las principales **características** de SQLite son:

* El tamaño, al tratarse de una biblioteca, es mucho menor que cualquier SGBD
* Reúne los cuatro criterios ACID (Atomicidad, Consistencia, Aislamiento y Durabilidad) logrando gran estabilidad
* Gran portabilidad y rendimiento

La gran **desventaja** de SQLite es la escalabilidad ya que no soporta bases de datos que sean muy grandes.

**PostgreSQL**

Este sistema gestor de base de datos relacional está **orientado a objetos** y es libre, publicado bajo la licencia BSD.

Sus principales **características** son:

* Control de Concurrencias multiversión (MVCC)
* Flexibilidad en cuanto a lenguajes de programación
* Multiplataforma
* Dispone de una herramienta (pgAdmin, https://www.pgadmin.org/) muy fácil e intuitiva para la administración de las bases de datos.
* Robustez, Eficiencia y Estabilidad.

La principal **desventaja** es la lentitud para la administración de bases de datos pequeñas ya que está optimizado para gestionar grandes volúmenes de datos.

Microsoft SQL Server

Es un sistema gestor de bases de datos relacionales basado en el lenguaje **Transact-SQL**, capaz de poner a disposición de muchos usuarios grandes cantidades de datos de manera simultánea.

Es un sistema propietario de **Microsoft**. Sus principales **características** son:

* Soporte exclusivo por parte de Microsoft.
* Escalabilidad, estabilidad y seguridad.
* Posibilidad de cancelar consultas.
* Potente entorno gráfico de administración que permite utilizar comandos DDL y DML.
* Aunque es nativo para Windows puede utilizarse desde hace ya un tiempo en otras plataformas como Linux o Docker.

Su principal **desventaja** es el precio. Cuenta con un plan gratuito (Express) pero lo normal es la elección de alguno de los **planes de pago** disponibles (Standard, Developer, Enterprise o SQL Azure, la versión de SQL Server en la nube).

**Oracle**

Tradicionamente, Oracle ha sido el **SGBD por excelencia para el mundo empresarial**, considerado siempre como el más **completo y robusto**, destacando por:

* Soporte de transacciones.
* Estabilidad.
* Escalabilidad.
* Multiplataforma.

La principal **desventaja**, al igual que SQL Server, es el coste del software ya que, aunque cuenta con una versión gratuita (Express Edition o XE), sus principales opciones son de pago.

Las **opciones de pago** disponibles son:

* + 1. Standard Edition (SE)
  + 2. Standard Edition One (SE1)
  + 3. Standard Edition 2 (SE2)
  + 4. Personal Edition (PE)
* 5. Lite Edition (LE)
* 6. Entreprise Edition (EE)

## Sistemas Gestores de bases de datos No Relacionales (NoSQL)

Una base de datos no relacional (**NoSQL**) es aquella base de datos que:

* No requiere de estructuras de datos fijas como tablas
* No garantiza completamente las características ACID
* Escala muy bien horizontalmente.

Se utilizan en **entornos distribuidos** que han de estar siempre disponibles y operativos y que gestionan un **importante volumen de datos**.

Para la administración de este tipo de bases de datos, actualmente los principales sistemas gestores de bases de datos (**SGBD NoSQL**) son:

### MongoDB

Estamos ante el Sistema Gestor de Bases de Datos no relacionales (SGBD NoSQL) más **popular y utilizado** actualmente.

MongoDB es un SBGD NoSQL **orientado a ficheros** que almacena la información en **estructuras BSON** con un esquema dinámico que permite su facilidad de integración.

Empresas como **Google, Facebook, eBay, Cisco o Adobe** utilizan MongoDB como Sistema Gestor de Bases de datos.

Las principales **características** de MongoDB son:

* Indexación y replicación
* Balanceo de carga
* Almacenamiento en ficheros
* Consultas ad hoc
* Escalabilidad horizontal
* Open Source

Como **desventaja** principal, MongoDB no es un SGBD adecuado para realizar transacciones complejas.

### Redis

Redis está basado en el **almacenamiento clave-valor**. Podríamos verlo como un vector enorme que almacena todo tipo de datos, desde cadenas, hashses, listas, etc.

El principal uso de este SGBD es para el **almacenamiento en memoria caché y la administración de sesiones**.

Las **características** principales son:

* Atomicidad y persistencia
* Gran velocidad
* Simplicidad
* Multiplataforma

### Cassandra

Al igual que Redis, Cassandra también utiliza **almacenamiento clave-valor**. Es un SGBD NoSQL **distribuido y masivamente escalable**.

**Facebook, Twitter, Instagram, Spotify o Netflix** utilizan Cassandra.

Dispone de un lenguaje propio para las consultas denominado **CQL** (Cassandra Query Languaje).

Las principales **características** de este SGBD NoSQL son:

* Multiplataforma
* Propio lenguaje de consultas (CQL)
* Escalado lineal y horizontal
* Es un SGBD distribuido
* Utiliza una arquitectura peer-to-peer

**Firebase**

Es un servicio web que nos proporciona un **backend** en la nube con una fuente de datos **NoSQL** en **tiempo real** y **librerías** para poder acceder a la base de datos desde aplicaciones **Web**, **IOS** o **Android**.

Los datos almacenados en **Firebase** se **sincronizan** con los clientes en **tiempo real**. Es básicamente con una base de datos remota que responde en tiempo real a los cambios realizados en los datos. De esta forma podemos escribir aplicaciones que almacenen o actualicen datos en Firebase y todos los clientes que utilicen dicha base de datos serán notificados en tiempo real de los cambios realizados.

**Firebase** también facilita la autenticación con el propio sistema de Firebase o mediante conectores como pueden ser Facebook, Twitter, etc.

**Firebase** nos permitirá programar aplicaciones centrándonos en la parte del frontend, dejando que todo el backend lo gestione Firebase.

## Principales características de firebase

Esta plataforma móvil tiene distintas características, repasemos alguna de ellas:

### 1. Es una herramienta soportada en múltiples plataformas

Firebase, se encuentra disponible para las principales plataformas móviles, ya sea Android, IOS o la Web.

### 2. Permite ganar dinero

Mediante el uso de Admob, firebase habilita la posibilidad de generar dinero mediante anuncios y publicidades.

### 3. Permite un desarrollo gratuito de tus apps

Esta plataforma digital, no pide pagar para la iniciación y uso de la misma.

Empieza a cobrar a la hora de manejarte en un nivel más avanzado y experimentado, puedes solicitar planes de mayor jerarquía según el uso que vayas a darle.

### 4. Genera crecimiento y desarrollo en tus aplicaciones

Esta es una de las características más importantes, puesto que se centra en el objetivo primordial de todo lo que es Firebase.

Mediante los servicios ofrecidos, permite que puedas desarrollar de una forma sencilla, segura y rápida tus aplicaciones en el dispositivo que te encuentres.

## ¿Qué servicios ofrece?

Los servicios ofrecidos se pueden dividir en 3 categorías:

### En el desarrollo

Son diferentes y con distintas utilidades:

* **Real time data base**: nos ayudan, dándonos una base de datos en tiempo real.
* **Autenticación**: se ejecutan para identificar a los usuarios mediante el email o las redes sociales.
* **Nube de almacenamiento**: almacenamiento y envío de archivos a la escala de Google.
* **Hosting**: se usa para publicar nuestra página web.
* **Remote config**: se emplea para modificar ciertos aspectos de nuestra app sin la necesidad de actualizar la misma.
* **Test lab**: es para probar la aplicación antes de publicarla.
* **Crash reporting**: se utiliza para reportar errores de la aplicación.

### Para hacer crecer nuestras aplicaciones y/o negocio

Para hacer crecer nuestras aplicaciones y con esto, nuestro negocio, nos da la posibilidad de:

* enviar notificaciones a los usuarios, con **Notifications;**
* permite mostrar las aplicaciones de una forma bastante adecuada en los resultados de los motores de búsqueda, con el uso de **App indexing;**
* proporciona la forma para acceder a la aplicación desde otros links desde otras aplicaciones, mediante **Dynamic Links;**
* permite hacer publicidad de nuestra aplicación usando **AdWords;**
* ayuda a monetizar nuestra aplicación mediante la publicidad, utilizando **AdMob.**

Como podemos observar, existen múltiples funcionalidades para lograr hacer crecer nuestro negocio.

Hay que tener en cuenta que las aplicaciones hoy en día representan un mercado que aún se está descubriendo y se mantiene en constante evolución, por ende, adherirse a este se debe hacer utilizando las mejores plataformas.

### Las Analíticas o Analytics

Analytics muestra los resultados sobre el comportamiento del usuario en su IOS o Android apps. Permitiendo que puedas tomar las mejores decisiones sobre tu producto y una optimización de tu estrategia de marketing.

**Firebase Analytics se puede utilizar conjunto a Google Adwords** para lograr obtener estadísticas sobre tu aplicación y cómo esta ha llegado a los usuarios y sus reacciones a la misma.

Son totalmente gratuitas e ilimitadas para todas nuestras aplicaciones.

## Valores (en dólar) de Firebase

Firebase, en principio, te da la opción de comenzar por un plan gratuito y luego, acceder a uno pago.

El plan sin cargos se llama **Spark**.

Los planes con costo son:

* Plan **Flame**, el cual, tiene precio fijo para las aplicaciones en expansión por USD $25 por mes.
* Plan **Blaze**, tiene una calculadora de precios que estima los gastos mensuales del plan.

Es así, como Firebase se adapta al uso que le vayas a dar al mismo.

**ANEXO**

\*Sobre ACID

ACID, letra por letra

**Atomicidad**

La Atomicidad requiere que cada transacción sea "todo o nada": si una parte de la transacción falla, todas las operaciones de la transacción fallan, y por lo tanto la base de datos no sufre cambios. Un sistema atómico tiene que garantizar la atomicidad en cualquier operación y situación, incluyendo fallas de alimentación eléctrica, errores y caidas del sistema.

**Consistencia**

La propiedad de Consistencia se asegura que cualquier transacción llevará a la base de datos de un estado válido a otro estado válido. Cualquier dato que se escriba en la base de datos tiene que ser válido de acuerdo a todas las reglas definidas, incluyendo (pero no limitado a) los constraints, los cascades, los triggers, y cualquier combinación de estos.

**aIslamiento**

El aislamiento ("Isolation" en inglés) se asegura que la ejecución concurrente de las transacciones resulte en un estado del sistema que se obtendría si estas transacciones fueran ejecutadas una atrás de otra. Cada transacción debe ejecutarse en aislamiento total; por ejemplo, si T1 y T2 se ejecutan concurrentemente, luego cada una debe mantenerse independiente de la otra.

**Durabilidad**

La durabilidad significa que una vez que se confirmó una transacción (commit), quedará persistida, incluso ante eventos como pérdida de alimentación eléctrica, errores y caidas del sistema. Por ejemplo, en las bases de datos relacionales, una vez que se ejecuta un grupo de sentencias SQL, los resultados tienen que almacenarse inmediatamente (incluso si la base de datos se cae inmediatamente luego).

### Otros SGBD NoSQL

Otros Sistemas Gestores de bases de datos no relacionales muy utilizados son:

* **[Azure Cosmos DB](https://azure.microsoft.com/es-es/services/cosmos-db/" \t "_blank)**
* **[RavenDB](https://ravendb.net/" \t "_blank)**
* **[ObjectDB](https://www.objectdb.com/" \t "_blank)**
* **[Apache CouchDB](http://couchdb.apache.org/" \t "_blank)**
* **[Neo4j](https://neo4j.com/" \t "_blank)**
* **[Google BigTable](https://cloud.google.com/bigtable/?hl=es" \t "_blank)**
* **[Apache Hbase](https://hbase.apache.org/" \t "_blank)**
* **[Amazon DynamoDB](https://aws.amazon.com/es/dynamodb/" \t "_blank)**