

Pruebas de rendimiento bases de datos

Breve descripción:

Este componente explicará cómo se afina un motor de base de datos, cómo se analizan y se optimizan las consultas mejorando sus tiempos de respuesta, y explica cómo funcionan los algoritmos de búsqueda.

Abril 2024

Tabla de contenido

Introducción	3
1. Tiempos computacionales.....	4
1.1. Estructura de datos	4
1.2. Complejidad algorítmica.....	5
1.3. Algoritmos de búsqueda	6
1.4. Índices y rendimiento en bases de datos	8
2. Análisis de rendimiento	9
2.1. Análisis de plan de ejecución de consultas	9
2.2. Optimización de consultas.....	10
2.3. Recomendaciones	11
3. Afinamiento de bases de datos	12
3.1. Motor de almacenamiento	12
3.2. Parámetros de configuración y herramientas de monitoreo.....	13
Síntesis	15
Material complementario	16
Glosario	18
Referencias bibliográficas	19
Créditos	20

Introducción

Bienvenido al presente componente formativo. Desarrollar “software” de calidad, implica también conocer cómo funcionan y cómo se comportan las bases de datos, aprovechando al máximo sus funcionalidades, por lo tanto, entender las estructuras de datos, el funcionamiento de los algoritmos de búsqueda, y en qué caso utilizarlos, así como analizar una sentencia de búsqueda y mejorarlas, junto con la afinación adecuada del motor de la base de datos según la arquitectura y las necesidades del cliente, logra mejorar el desempeño de las bases de datos, ofreciendo una buena experiencia al usuario con respecto a las solicitudes que él mismo realice.

Cabe resaltar, que para este proceso, existen técnicas como la medición a través de indicadores de gestión, observación, encuestas, entre otras herramientas, que permiten el mejoramiento continuo de los procedimientos relacionados con los clientes internos y externos.

Y todo esto, al final, se convierte en calidad del “software”.

1. Tiempos computacionales

La complejidad informática estudia la clasificación de los problemas computacionales, teniendo en cuenta su dificultad, además mide el tiempo en que un algoritmo le da respuesta a una necesidad o a un problema, utilizando ciertas fórmulas matemáticas en las que no se profundizará, pero se explicarán algunos conceptos que se deben conocer para entender el ámbito de este tema, las ecuaciones matemáticas y todo su desarrollo se pueden investigar dirigiéndose a las referencias más adelante.

1.1. Estructura de datos

Es la forma de organizar los datos en una computadora para que después sean utilizados de manera eficiente. Existen diferentes estructuras de datos y estas estructuras son utilizadas dependiendo de la necesidad de la aplicación. Para el almacenamiento y recuperación de los datos, se manejan dos tipos de estructura de datos, dinámica y estática, y los recorridos de búsqueda de estas estructuras pueden ser lineales y no lineales.

Se crean algoritmos para poder recorrer y hacer búsquedas de información en grandes cantidades de datos, y se explica el funcionamiento de cada uno de ellos, para mejorar su comprensión:

- **Estructura de datos lineal:** las estructuras lineales son estructuras que se definen inicialmente como una secuencia de elementos en donde se crea una relación de predecesor y sucesor. Estas estructuras utilizan cuatro operaciones básicas que son: crear, añadir, borrar y consultar, y existen tres estructuras lineales que son: las pilas ejecuta las tres operaciones al final de la secuencia, listas las tres operaciones, se ejecutan sobre una posición

en específico y puede ser desplazada, y colas se añade al final y se consulta y borra al inicio.

- **Árbol binario:** es una estructura de datos en donde se representa por nodos, y cada nodo puede tener dos hijos, uno a la izquierda y otro a la derecha. Si uno de los nodos tiene un valor de “null”, se le dice que es un nodo hoja.

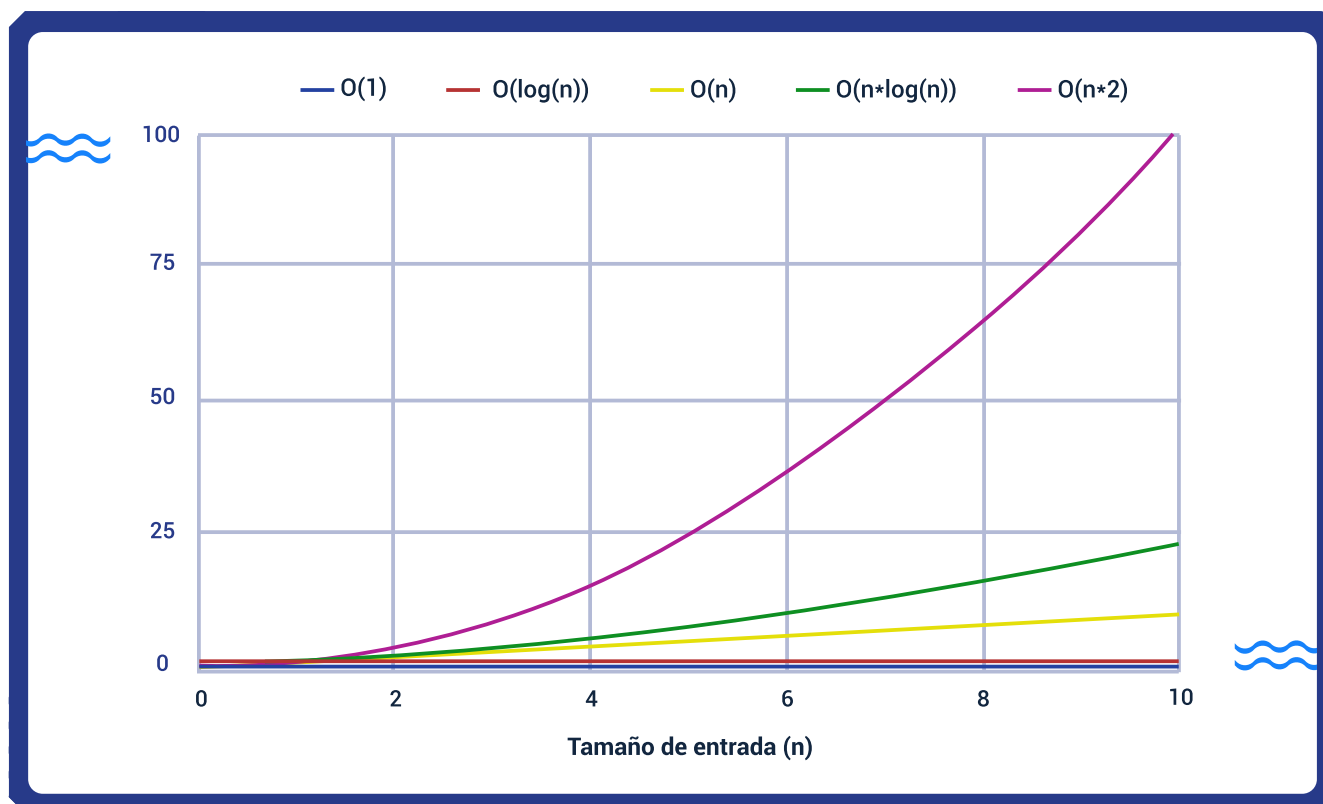
Estas estructuras se pueden recorrer de tres formas:

- **Preorden:** se sigue el orden primero con el nodo raíz, después el nodo de la izquierda y por último, el de la derecha.
- **Postorden:** se recorre primero el nodo de la izquierda, luego el de la derecha y por último, el nodo raíz.
- **Inorden:** primero se recorre el nodo izquierdo, después la raíz y por último, el derecho.

1.2. Complejidad algorítmica

Es una métrica que ayuda a describir el comportamiento de un algoritmo en dos características, una es el tiempo que se demora para solucionar un problema y la otra es la memoria que utiliza para hacerlo; esto ayuda a seleccionar qué algoritmo es más eficiente que otro para solucionar un problema, y una forma de medir esto es utilizando la Notación Asintótica, la cual mide la eficiencia y complejidad de un algoritmo. Según esta notación, se puede interpretar que el algoritmo más eficiente es el que no varía o varía lo menos posible, sin importar las entradas. Su comportamiento se representa de la siguiente forma:

Figura 1. Gráfica de complejidad Big-O



1.3. Algoritmos de búsqueda

Es simplemente buscar un dato en un conjunto de datos; existen algoritmos diferentes para realizar este tipo de búsquedas y depende de cómo estén ordenados los datos. Aquí se debe observar qué tipo de algoritmo utilizar: uno que busque secuencialmente o uno que busque aleatoriamente, teniendo en cuenta que la búsqueda secuencial, es el algoritmo menos eficiente, porque solo toma el número que se desea buscar y lo compara uno a uno con el conjunto de datos hasta encontrarlo.

Hablemos ahora sobre la ordenación. Este es un proceso para organizar los elementos o un conjunto de datos de forma ascendente o descendente, si es el caso que los elementos sean números y si son alfabéticos, en orden alfabético; esta

ordenación se repite hasta que los elementos o datos estén ordenados correctamente.

Para lograr este proceso, se utilizan dos simples operaciones, comparación e intercambio.

- **Método de burbuja**

Recorre el “array”, comparando valores que se encuentran en posiciones seguidas. Si los datos de las dos posiciones no se encuentran ordenadas, se intercambian, y se sigue iterando hasta que el “array” quede en el orden correcto.

- **Método de selección**

En este método lo primero se debe hacer, es seleccionar el elemento de la primera posición y compararlo como el más pequeño o el más grande de todo el conjunto de elementos. En este caso, seleccionaremos el más pequeño, recorreremos posición por posición, comparando que el elemento que seleccionamos es el más pequeño que el de la posición actual; si después de recorrer todo el “array” encontramos el más pequeño, lo intercambiamos con el elemento que está en la primera posición y nos desplazamos a la segunda posición, volvemos a comprar hasta el final del “array” y si encontramos el más pequeño, lo intercambiamos con el elemento de la segunda posición y así sucesivamente hasta terminar de desplazarnos por todo el “array”.

- **Método de inserción**

Este método lo que hace es comprar el elemento con el elemento que se encuentra a la izquierda. Como el primer elemento está en la posición 0 y no hay ningún elemento al lado izquierdo, pasamos al elemento 1 y lo

comparamos con el elemento de la posición 0, si este elemento es menor que el elemento de la posición anterior, entonces lo intercambiamos, de lo contrario pasamos a la siguiente posición, el elemento debe desplazarse hacia la izquierda tantas veces hasta que encuentre uno menor.

- **Quicksort**

Este método lo que hace es tomar inicialmente un pivót. Normalmente se toma el primer elemento, después se empieza a comparar cada uno de los elementos, y en el “array” de la izquierda se van agregando uno a uno los menores o iguales al pivót y a la derecha se van agregando los mayores; después se concatenan teniendo en cuenta que el pivót queda en el medio, y se toman los “array” a la izquierda y derecha y vuelve y se toma un pivót para volver a hacer el proceso, al final devuelve el “array” ordenado.

1.4. Índices y rendimiento en bases de datos

Para mejorar el rendimiento de la base de datos y optimizar las consultas, es importante utilizar índices que permitan especificar las búsquedas de la información.

Índice es un identificador que permite que la consulta se haga teniendo en cuenta este parámetro y pueda ejecutarse de forma más eficiente y rápida, existen tipos de índices que se ajustan a la implementación de las búsquedas, cuando se crean las tablas de las bases de datos, normalmente se crea una llave primaria que indica que la búsqueda debe ser relacionada con esa llave en especial, esta llave primaria es a su vez un índice.

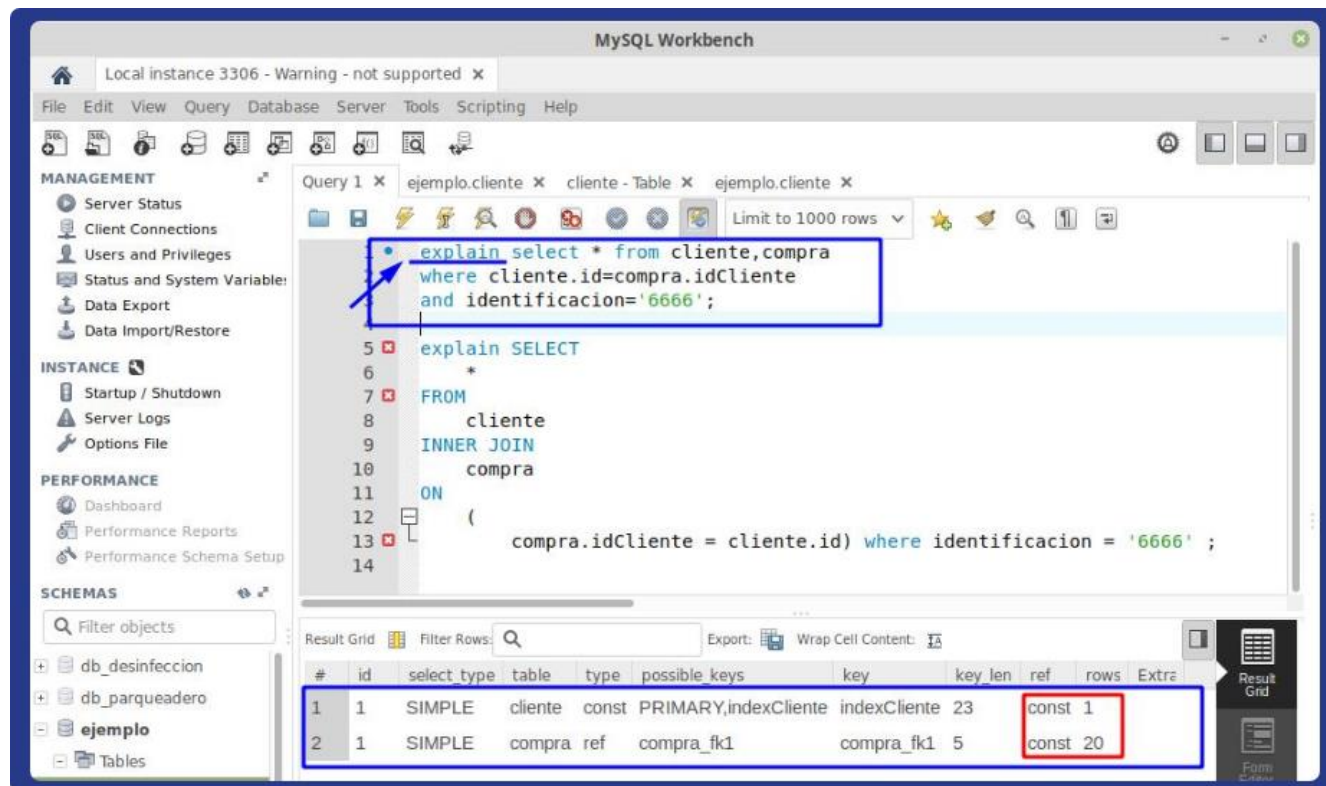
2. Análisis de rendimiento

La base de datos es el centro de toda aplicación, si la aplicación no responde de la mejor manera después de un tiempo, se debe a la escalabilidad de la base de datos y del rendimiento de la misma; esto bajará la calidad de la aplicación por los tiempos de respuesta de la información. Además, existen muchos factores que si no se tienen en cuenta, pueden generar cuellos de botella y muchos inconvenientes más.

2.1. Análisis de plan de ejecución de consultas

Lo primero es identificar las consultas que van a traer más información y que pueden tornarse lentas al momento en que los usuarios comiencen a realizar solicitudes.

Figura 2. Monitoreo de la consulta



Luego se inicia el monitoreo de esas consultas identificando los tiempos de respuesta, utilizando el comando “explain” en MySQL. Hay que tener en cuenta que este comando puede variar, dependiendo del motor de base de datos que se utilice.

Este comando lo que hace es desplegar el plan de ejecución de la consulta. En este caso se va a observar las dos últimas columnas que muestran que se está utilizando solo un índice y que arroja 20 resultados, de esta forma se puede verificar y monitorear qué hace la consulta.

2.2. Optimización de consultas

Ahora se puede monitorear el tiempo de respuesta de las consultas, en este caso se implementaron dos consultas que arrojan el mismo resultado, pero los tiempos de respuesta entre una y otra cambian.

- **Consulta 1:** si se observa la sentencia marcada con azul, no utiliza “Join” y solo utiliza la llave primaria, y la segunda sentencia utiliza el “Join”. Si se detalla el tiempo de la sentencia roja, es mucho mejor el tiempo de respuesta, por lo tanto, una forma de optimizar las sentencias de búsqueda con respecto al rendimiento es utilizando los “Join”.
- **Consulta 2:** ahora se agrega además un “index” a la consulta con el campo identificación.
- **Consulta 3:** el tiempo entre las dos consultas, agregándole el índice mejora aún más, de esta forma se puede monitorear y mejorar el rendimiento de nuestras consultas y de la base de datos.

2.3. Recomendaciones

Al momento de mejorar un sistema, se debe tener muy en cuenta cómo se arman las consultas en la base de datos y como tenemos claro cómo se arman estas sentencias y se mejoran dichas consultas, se darán unas recomendaciones para mejorar el rendimiento. Estas son:

- a) No utilizar consultas anidadas.
- b) Tratar de utilizar los “Joins” si es posible.
- c) Si se debe usar por fuerza mayor una consulta anidada usarla solo en el FROM.
- d) Normalizar la base de datos.
- e) Tratar de guardar cálculos en un campo siempre y cuando se necesite en determinado tiempo.
- f) Utilizar los tipos y las longitudes de los campos de forma adecuada.

3. Afinamiento de bases de datos

El afinamiento de base de datos es el proceso que se hace de acuerdo al sistema operativo, a la arquitectura del servidor, como el disco duro, memoria, etc., y a las transacciones y usuarios que se puedan conectar y hacer solicitudes.

Los informes basados en esquemas de rendimiento, proporcionan información sobre las operaciones del servidor MySQL a través de útiles informes de alto nivel. MySQL “Workbench” utiliza las vistas “SYS” en el esquema de rendimiento para generar más de 20 informes para ayudar a analizar el rendimiento de sus bases de datos MySQL. Los informes ayudan a analizar los puntos de acceso de E / S, descubrir declaraciones SQL de alto costo y revisar las estadísticas de espera y las métricas del motor InnoDB.

3.1. Motor de almacenamiento

Un motor de base de datos es el que ofrece un sistema de administración para utilizar operaciones para editar, leer y actualizar datos. En este caso se hablará de MySQL y MariaDB, y existen dos tipos de motores: transaccionales y no transaccionales:

- a) InnoDB.
- b) MyISAM.
- c) CSV.
- d) Merge.
- e) Archive.
- f) Blackhole.
- g) Federated.

3.2. Parámetros de configuración y herramientas de monitoreo

Las configuraciones para un óptimo funcionamiento de una base de datos, dependen del motor que se use, el tipo de base de datos, las necesidades del sistema, entre otras características. No es lo mismo configurar una base de datos en un servidor básico que en un servidor con discos en espejo con espacio sólido y 32 Gb de memoria; tampoco es lo mismo configurar un servidor para que almacene una base de datos de forma local en un servidor propio, que configurarlo en un contenedor ofrecido por Amazon.

A continuación, se presentan algunos conceptos para tener en cuenta, según las necesidades y arquitecturas:

Aplicación web

- Normalmente vinculado a la CPU.
- DB mucho más pequeño que RAM.
- 90 % o más consultas simples.

Procesamiento de transacciones en línea (OLTP)

- Normalmente vinculado a CPU o E / S.
- DB un poco más grande que la RAM a 1TB.
- 20-40 % de consultas de escritura de datos pequeños.
- Algunas transacciones largas y consultas de lectura complejas.

Almacén de datos (DW)

- Normalmente vinculado a E / S o RAM.
- Grandes cargas masivas de datos.

- Grandes consultas de informes complejos.
- También llamado "Apoyo a la toma de decisiones" o "Inteligencia empresarial".

Aplicación de escritorio

- No es una base de datos dedicada.
- Una estación de trabajo general, quizás para un desarrollador.

Tipo de aplicación mixta

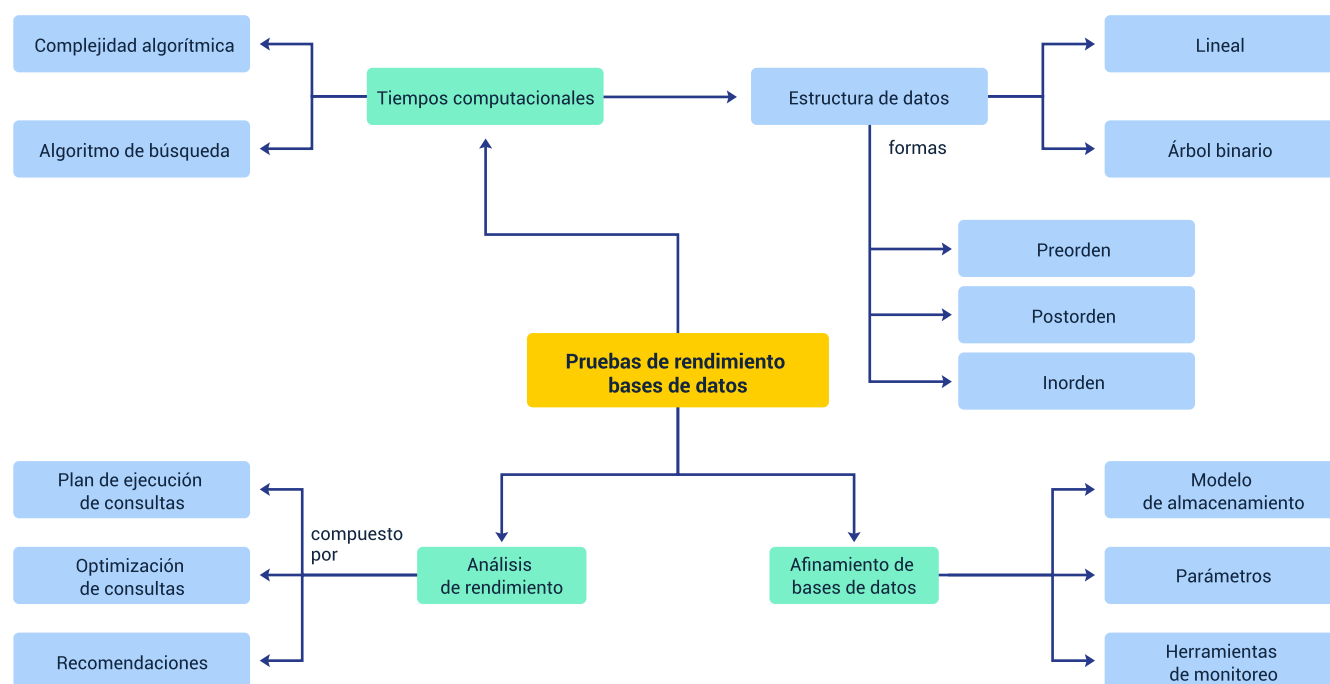
- Características mixtas de DW y OLTP.
- Una amplia mezcla de consultas.

A continuación, se mencionan algunas herramientas que podrían ayudar con el monitoreo del rendimiento de las bases de datos.

- Germain APM.
- Datadog.
- Percona.
- MariaDB.
- Sowflake.

Síntesis

Lo invitamos a ver el siguiente mapa, el cual presenta una síntesis del componente formativo.



Material complementario

Tema	Referencia	Tipo de material	Enlace del recurso
Estructura de datos	Universitat Politècnica de València. (2011). Estructura de datos Lineales: Pila, Cola y Lista con Punto de Interés [video]. YouTube.	Video	https://www.youtube.com/watch?v=-Shr2s0gYao
Estructura de datos	Francisco, T. (2015). Explicación del método de la burbuja (BubbleSort) [video]. YouTube.	Video	https://www.youtube.com/embed/zVjZdrh3tSA
Estructura de datos	Chio Code. (2021). Ordenamiento por selección Selection Sort [video]. YouTube.	Video	https://www.youtube.com/embed/Myy-eU-SWbE
Estructura de datos	KhanAcademyEspañol. (2014). Algoritmo de ordenamiento por inserción [video]. YouTube.	Video	https://www.youtube.com/embed/bB8Px8D9QdQ
Estructura de datos	Montero, L. (2017). Algoritmos - Ordenamiento rapido - Quicksort [video]. YouTube.	Video	https://www.youtube.com/embed/DYmTpUfcyT8
Índices y rendimiento en bases de datos	Domínguez Gutú, J. (2020). Índices en bases de datos - conceptos [video]. YouTube.	Video	https://www.youtube.com/embed/C4_kHhpY5jE
Análisis de rendimiento	Cerebrum VE. (2013). BD Avanzadas – Cómo optimizar una consulta [video]. YouTube.	Video	https://www.youtube.com/embed/1HH4ZYXhJYE

Tema	Referencia	Tipo de material	Enlace del recurso
Análisis de rendimiento	Decharlas Castellón. (2014). Optimización del rendimiento con MySQL [video]. YouTube.	Video	https://www.youtube.com/embed/rsHXazV_Ec
Afinamiento de bases de datos	Mora, J. (2015). Herramientas para optimizar mysql mysltuner [video]. YouTube.	Video	https://www.youtube.com/watch?v=WUKZ2SCuMUQ

Glosario

Aplicación: es un programa informático diseñado como una herramienta para realizar operaciones o funciones específicas. Generalmente, son diseñadas para facilitar ciertas tareas complejas y hacer más sencilla la experiencia informática de las personas.

“Array”: colección de elementos y sus posiciones.

Base de datos: es una recopilación organizada de información o datos estructurados, que normalmente se almacena de forma electrónica en un sistema informático.

Implementación: poner en funcionamiento o aplicar métodos, medidas, etc., para llevar algo a cabo.

Iterar: realizar cierta acción o acciones varias veces.

Servidor: es un conjunto de computadoras capaz de atender las peticiones de un cliente y devolverle una respuesta en concordancia.

Referencias bibliográficas

Davidochobits. (2020). Motores de almacenamiento en MySQL y MariaDB.

<https://www.ochobitshacenunbyte.com/2020/08/11/motores-de-almacenamiento-en-mysql-y-mariadb/>

Guillermo, J. (2018). ¿Qué es la complejidad algorítmica y con qué se come?

<https://medium.com/@joseguillermo /qu%C3%A9-es-la-complejidad-algor%C3%ADmica-y-con-qu%C3%A9-se-come-2638e7fd9e8c>

Montero, L. (2017). Algoritmos y notación asintótica.

<https://medium.com/laboratoria-developers/algoritmos-y-notaci%C3%B3n-asint%C3%B3tica-817a666ca444>

Créditos

Nombre	Cargo	Centro de Formación y Regional
Milady Tatiana Villamil Castellanos	Responsable del Ecosistema	Dirección General
Olga Constanza Bermúdez Jaimes	Responsable de Línea de Producción	Centro de Servicios de Salud - Regional Antioquia
David Eduardo Lozada Cerón	Experto Temático	Centro de Teleinformática y Producción Industrial - Regional Cauca
Ana Catalina Córdoba Sus	Evaluable Instruccional	Centro de Servicios de Salud - Regional Antioquia
Yerson Fabián Zárate Saavedra	Diseñador de Contenidos Digitales	Centro de Servicios de Salud - Regional Antioquia
Edward Leonardo Pico Cabra	Desarrollador Fullstack	Centro de Servicios de Salud - Regional Antioquia
Edgar Mauricio Cortés García	Actividad Didáctica	Centro de Servicios de Salud - Regional Antioquia
Luis Gabriel Urueta Álvarez	Validador de Recursos Educativos Digitales	Centro de Servicios de Salud - Regional Antioquia
Jaime Hernán Tejada Llano	Validador de Recursos Educativos Digitales	Centro de Servicios de Salud - Regional Antioquia
Margarita Marcela Medrano Gómez	Evaluable para Contenidos Inclusivos y Accesibles	Centro de Servicios de Salud - Regional Antioquia
Daniel Ricardo Mutis Gómez	Evaluable para contenidos inclusivos y accesibles	Centro de Servicios de Salud - Regional Antioquia