**FORMATO PARA EL DESARROLLO DE COMPONENTE FORMATIVO**

|  |  |
| --- | --- |
| PROGRAMA DE FORMACIÓN | Tecnología en Gestión Eficiente de la Energía |

|  |  |  |  |
| --- | --- | --- | --- |
| COMPETENCIA | 210601026-Procesar datos de acuerdo con procedimiento técnico y metodología estadística | RESULTADOS DE APRENDIZAJE | 210601026-3. Elaborar base de datos de acuerdo con las necesidades del sistema de gestión de la energía. |

|  |  |
| --- | --- |
| NÚMERO DEL COMPONENTE FORMATIVO | 019 |
| NOMBRE DEL COMPONENTE FORMATIVO | Base de datos |
| BREVE DESCRIPCIÓN | Los datos están en constante evolución y transformación en todo el mundo. Ha introducido nuevos mecanismos de mejora para todos los sistemas, también ha hecho que su operación sea más desafiante. El registro, la supervisión y el almacenamiento de datos precisos ayuda a abordar de una manera más eficaz estos desafíos y necesidades que contemplan los sistemas. |
| PALABRAS CLAVE | Atributos, cliente, datos, MySQL, NoSQL. |

|  |  |
| --- | --- |
| ÁREA OCUPACIONAL | 2 - Ciencias Naturales, aplicadas y relacionadas |
| IDIOMA | Español |

1. **TABLA DE CONTENIDOS**

**1. Introducción a las bases de datos**

1.1. Evolución de las bases de datos

1.2. Tipos de bases de datos

1.3. Herramientas de “*software”* más conocidas para bases de datos

**2. Fases del proceso de desarrollo de bases de datos**

**3. Entidades, relaciones y atributos**

**4. Modelo ER en base de datos**

4.1. Entidad

4.2. Tipo de relación y conjunto de relaciones

**5. Procedimientos más importantes en bases de datos**

5.1. Índice de base de datos

5.2. Vistas en base de datos

5.3. Reportes o informes en bases de datos

5.4. Permisos, roles y control de acceso de usuarios

5.5. Optimización de bases de datos

5.6. Limpieza de la base de datos

5.7. Pruebas en la base de datos

**6. MySQL**

**B DESARROLLO DE CONTENIDOS**

**Introducción**

1. **Introducción a las bases de datos**

¡Bienvenido al fascinante mundo de las bases de datos! Este componente formativo lo sumergirá en todo lo relacionado con este tema tan importante.

Los sistemas de administración de bases de datos son fundamentales para organizar la información relevante en cualquier organización. Simplifican la conexión y el acceso a los datos, lo que resulta vital para los objetivos y estrategias comerciales.

Imagine la importancia de los datos organizacionales para el funcionamiento efectivo de una empresa. Un sistema de gestión de bases de datos almacenados organiza y monitorea la información de manera eficiente, lo que mejora el desempeño y la eficiencia empresarial, ahorrando costos y permitiendo una gestión rentable.

Un sistema de bases de datos eficientes no solo almacena y organiza datos vitales para las metas y estrategias empresariales, sino que también utiliza aplicaciones de software avanzadas. Además, su capacidad para almacenar y compartir datos en toda la organización permite un acceso y una partición ágiles entre los usuarios, lo que facilita la búsqueda instantánea en grandes bases de datos y mejora la eficiencia en la gestión de múltiples tipos de información.

Este componente formativo le brindará una comprensión profunda de los componentes clave de una base de datos, que incluyen datos, hardware, software, personas y procedimientos. ¡Prepárate para explorar y adentrarte en el emocionante mundo de las bases de datos!

* 1. **Evolución de las bases de datos**

La base de datos ha completado más de 50 años de viaje desde su evolución del sistema de archivos planos a los sistemas relacionales y relacionales de objetos. Ha pasado por varias generaciones y esta evolución ha sido así:

Cartas de diálogo

DI\_CF019\_1.1 \_Evolución de las bases de datos.

**DBMS (Sistema de gestión de bases de datos):** el sistema de gestión de bases de datos es un *software* que se utiliza para almacenar y recuperar la base de datos, por ejemplo, Oracle, MySQL, etc.; estas son algunas herramientas populares de DBMS.

* DBMS proporciona la interfaz para realizar las diversas operaciones como creación, eliminación, modificación, etc.
* DBMS permite al usuario crear sus bases de datos según sus necesidades.
* DBMS acepta la solicitud de la aplicación y proporciona datos específicos a través del sistema operativo.
* DBMS contiene el grupo de programas que actúa de acuerdo con las instrucciones del usuario.
* Proporciona seguridad a la base de datos.

**Ventaja de DBMS:**

* **Controla la redundancia:** almacena todos los datos en un solo archivo de base de datos, por lo que puede controlar la redundancia de datos.
* **Compartir datos:** un usuario autorizado puede compartir los datos entre varios usuarios.
* **Respaldo:** proporciona un subsistema de copia de seguridad y recuperación. Este sistema de recuperación crea datos automáticos a partir de fallas del sistema y restaura los datos si es necesario.
* **Varias interfaces de usuario:** proporciona un tipo diferente de interfaces de usuario como GUI, interfaces de aplicaciones.

**Desventaja de DBMS:**

* **Tamaño:** ocupa un gran espacio en disco y una gran memoria para funcionar de manera eficiente.
* **Costo:** DBMS requiere un procesador de datos de alta velocidad y una memoria más grande para ejecutar el *software* DBMS, por lo que es costoso.
* **Complejidad:** DBMS crea requisitos y complejidad adicionales.

**RDBMS (Sistema de gestión de bases de datos relacionales):** se representa como una tabla que contiene filas y columnas. RDBMS se basa en el modelo relacional; fue presentado por EF Codd. Una base de datos relacional contiene los siguientes componentes:

* Mesa
* Grabar / tupla
* Campo / nombre de columna / atributo
* Ejemplo
* Esquema
* Teclas

Un RDBMS es un DBMS tabular que mantiene la seguridad, integridad, precisión y consistencia de los datos.

**1.2. Tipos de bases de datos**

Hay varios tipos de bases de datos que se utilizan para almacenar diferentes variedades de datos. A través del siguiente recurso se invita a explorar los tipos de bases de datos.

Slyders

DI\_CF019\_1.2 \_Tipos de bases de datos.

|  |
| --- |
| Base de datos relacional  Esta base de datos se centra en el modelo de datos relacionales, que almacena datos en forma de filas (tupla) y columnas (atributos), y juntos forman una tabla (relación). Una base de datos relacional utiliza SQL para almacenar, manipular y mantener los datos. EF Codd inventó la base de datos en 1970. Cada tabla de la base de datos lleva una clave que hace que los datos sean únicos, los demás ejemplos de bases de datos relacionales son MySQL, Microsoft SQL Server, Oracle, etc.  base de datos relacional.jpg  <https://blog.powerdata.es/el-valor-de-la-gestion-de-datos/la-base-de-datos-relacional-y-los-grandes-volumenes-de-datos>  Propiedades de la base de datos relacional: existen las siguientes cuatro propiedades comúnmente conocidas de un modelo relacional denominadas propiedades ACID, donde:   * A de *atomicity* que significa atomicidad: asegura que la operación de datos se completará con éxito o con fracaso. Sigue la estrategia de "todo o nada", por ejemplo, una transacción se confirmará o se abortará. * C de *consistency* que significa consistencia: si se realiza alguna operación sobre los datos se debe conservar su valor antes y después de la operación, por ejemplo, el saldo de la cuenta antes y después de la transacción debe ser correcto, es decir, debe permanecer conservado. * I de *isolation* que significa aislamiento: puede haber usuarios concurrentes para acceder a los datos al mismo tiempo desde la base de datos. Por lo tanto, el aislamiento entre los datos debe permanecer aislado, por ejemplo, cuando ocurren múltiples transacciones al mismo tiempo los efectos de una transacción no deben ser visibles para las otras transacciones en la base de datos. * D de *durability* que significa durabilidad: asegura que una vez que completa la operación y confirma los datos, los cambios de datos deben quedar permanentes. |

**Base de datos NoSQL**

NonSQL / Not Only SQL es un tipo de base de datos que se utiliza para almacenar una amplia gama de conjuntos de datos. No es una base de datos relacional, ya que almacena datos no solo en forma tabular, sino de varias formas diferentes. Surgió cuando aumentó la demanda de crear aplicaciones modernas, por lo tanto, NoSQL presentó una amplia variedad de tecnologías de bases de datos en respuesta a las demandas; se puede dividir aún más una base de datos NoSQL en los siguientes cuatro tipos:

|  |
| --- |
| * Almacenamiento de clave-valor: es el tipo más simple de almacenamiento de base de datos donde almacena cada elemento individual como una clave (o nombre de atributo) que mantiene su valor juntos. * Base de datos orientada a documentos: un tipo de base de datos que se utiliza para almacenar datos como documentos similares a JSON. Ayuda a los desarrolladores a almacenar datos utilizando el mismo formato de modelo de documento que se utiliza en el código de la aplicación. * Bases de datos de gráficos: se utiliza para almacenar grandes cantidades de datos en una estructura similar a un gráfico. Por lo general, los sitios web de redes sociales utilizan la base de datos de gráficos. * Almacenes de columna ancha: es similar a los datos representados en las bases de datos relacionales. Aquí, los datos se almacenan juntos en columnas grandes, en lugar de almacenarse en filas. Ver figura.   Figura 1  *Bases de datos SQL y NoSQL*  Bases de datos Relacionales y NoSQL  Nota. Tomado de <https://www.diegocalvo.es/bases-de-datos/> |

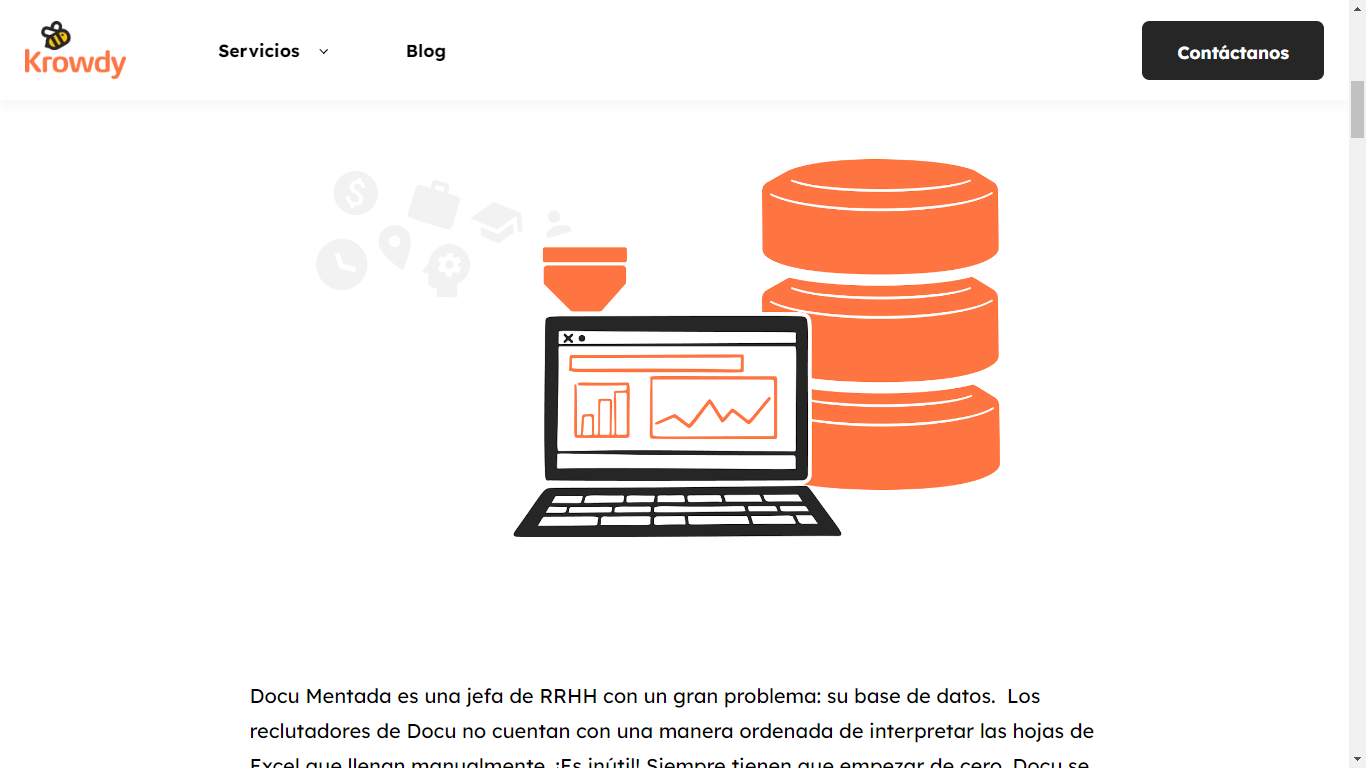
Siguiendo con la temática de tipos de bases de datos se invita a explorar el siguiente recurso.

Slyders

DI\_CF019\_1.2 \_Tipos de bases de datos.

**Base de datos personal**

La recopilación y el almacenamiento de datos en el sistema del usuario define una base de datos personal. Dicha base de datos está diseñada básicamente para un solo usuario.



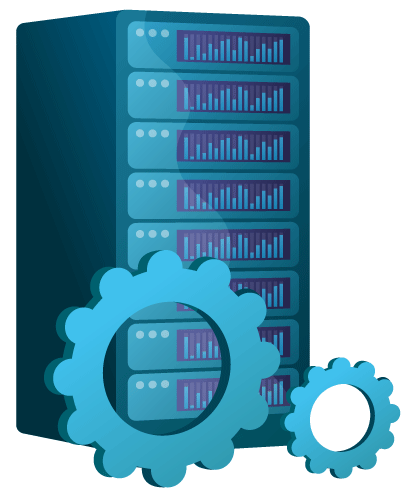
<https://blog.krowdy.com/todo-para-tener-base-de-candidatos-perfecta>

**Ventaja de la base de datos personal**

* Es simple y fácil de manejar.
* Ocupa menos espacio de almacenamiento, ya que es de tamaño pequeño.

**Base de datos operativa**

El tipo de base de datos que crea y actualiza la base de datos en tiempo real. Básicamente está diseñada para ejecutar y manejar las operaciones diarias de datos en varios negocios, por ejemplo, una organización utiliza bases de datos operativas para administrar transacciones diarias.



<https://idital.com/diccionario-seo/base-de-datos/>

**Base de datos empresarial**

Las grandes organizaciones o empresas utilizan esta base de datos para gestionar una gran cantidad de datos. Ayuda a las organizaciones a aumentar y mejorar su eficiencia. Una base de datos de este tipo permite el acceso simultáneo a los usuarios.



<https://revistabyte.es/comparativa/comparativa-bases-de-datos-2020/>

**Ventajas de la base de datos empresarial**

* Los procesos múltiples son compatibles con la base de datos empresarial.
* Permite ejecutar consultas paralelas en el sistema.

|  |
| --- |
| 1.3. Herramientas de *software* más conocidas para bases de datos  A continuación puede ver las herramientas más comunes.  Base de Datos Relacionales  <https://www.diegocalvo.es/bases-de-datos/>  Bases de Datos NoSQL  <https://www.diegocalvo.es/bases-de-datos/> |

|  |
| --- |
| 2. Fases del proceso de desarrollo de bases de datos  Serie de fases en las que cada una de ellas tiene su propia experiencia. Las fases principales son la planificación, el análisis de requisitos y la implementación; cada proceso se divide en pasos. El diagrama de estas fases se muestra como:    **PLANIFICACIÓN**  Fase 1  **ANÁLISIS DE REQUISITOS**  Fase 2  **DISEÑO**  Fase 3  **SELECCIÓN DBMS**  Fase 4  **IMPLEMENTACIÓN**  Fase 5  **MANTENIMIENTO**  Fase 6 |

Se invita a explorar el siguiente componente donde hallará las fases del proceso de desarrollo de las bases de datos.

Acordeón

DI\_CF019\_2 \_Fases del proceso de desarrollo de bases de datos.

**3. Entidades, relaciones y atributos**

Por medio del siguiente recurso explore todo sobre entidades, relaciones y atributos.

Slider

DI\_CF019\_3 \_Entidades, relaciones y atributos.

|  |
| --- |
| Para mayor comprensión, diríjase al Anexo 1 - Relaciones en una base de datos. <https://docs.google.com/document/d/1ANIIiI0KBSos-w0JTjed5nBRzR-NuEjP/edit?usp=sharing&ouid=103879860323491984918&rtpof=true&sd=true> |

**4. Modelo ER en base de datos**

Se utiliza para modelar la vista lógica del sistema desde la perspectiva de los datos, que consta de estos componentes:

* Entidad.
* Tipo de entidad.
* Conjunto de entidades.

|  |
| --- |
| 4.1. Entidad  Puede ser un objeto con existencia física - una persona, automóvil, casa o empleado en particular - o puede ser un objeto con existencia conceptual - una empresa, un trabajo o un curso universitario.  Una entidad es un objeto de tipo de entidad y el conjunto de todas las entidades se denomina conjunto de entidades, por ejemplo, E1 es una entidad que tiene el tipo de entidad estudiante y el conjunto de todos los estudiantes se denomina conjunto de entidades. En el diagrama ER, el tipo de entidad se representa como:  **ESTUDIANTE**  **Tipo de entidad**  **Conjunto de entidades**  Atributos: son las propiedades que definen el tipo de entidad, por ejemplo, No. ID, nombre, DOB, edad, dirección, No. de teléfono son los atributos que definen el tipo de entidad estudiante. En el diagrama ER el atributo está representado por un óvalo.   * Atributo clave: identifica de manera única a cada entidad en el conjunto de entidades, por ejemplo, No. ID será único para cada estudiante. En el diagrama ER el atributo clave está representado por un óvalo con líneas subyacentes. * Atributo compuesto: está compuesto por muchos otros atributos, por ejemplo, el atributo dirección del tipo de entidad del estudiante consta de calle, ciudad, departamento y país. En el diagrama ER, el atributo compuesto está representado por un óvalo que consta de óvalos. * Atributo multivalor: consta de más de un valor para una entidad determinada, por ejemplo, el número de teléfono (puede ser más de uno para un estudiante determinado). En el diagrama ER, el atributo multivalor está representado por un óvalo doble.   **No. Teléfono**   * Atributo derivado: un atributo que puede derivarse de otros atributos del tipo de entidad, por ejemplo, edad (se puede derivar de DOB). En el diagrama ER, el atributo derivado está representado por un óvalo punteado. |

**4.2. Tipo de relación y conjunto de relaciones**

A partir de este recurso podrá descubrir los tipos de relación y el conjunto de relaciones.

Slyders

DI\_CF019\_4.2 \_Tipo de relación y conjunto de relaciones.

**Restricción de participación:** se aplica a la entidad que participa en el conjunto de relaciones.

* **Participación total:** cada entidad del conjunto de entidades debe participar en la relación. Si cada alumno debe matricularse en un curso, la participación del alumno será total. La participación total se muestra con una línea doble en el diagrama ER.
* **Participación parcial:** la entidad en el conjunto de entidades puede o **no** participar en la relación. Si algunos cursos no están matriculados por ninguno de los alumnos, la participación del curso será parcial.

|  |
| --- |
| El diagrama muestra el conjunto de relaciones “inscrito en” con el conjunto de entidad de estudiante con participación total y el conjunto de entidad del curso con participación parcial.  **Estudiante**  **Inscrito en:**  **Curso**  **m**  **n**  Usando conjuntos se puede representar como:  Todos los estudiantes del conjunto de entidades de estudiantes participan en una relación; pero existe un curso el C4 que no participa en la relación. |

**Tipo de entidad débil y relación de identificación**

Como se discutió anteriormente, un tipo de entidad tiene un atributo clave que identifica de manera única a cada entidad en el conjunto de entidades; pero existe algún tipo de entidad para la cual no se puede definir un atributo clave, este se denomina tipo de entidad débil, por ejemplo, una empresa puede almacenar la información de los dependientes (padres, hijos, cónyuge) de un empleado; pero los dependientes no tienen existencia sin el empleado. Por lo tanto, el dependiente será un tipo de entidad débil y el empleado será el tipo de entidad de identificación para el dependiente.

Un tipo de entidad débil está representado por un rectángulo doble, la participación del tipo de entidad débil es siempre total.

|  |
| --- |
| La relación entre el tipo de entidad débil y su tipo de entidad fuerte de identificación se denomina relación de identificación y está representada por un doble diamante.  **Empleado**  **Tiene**  **Dependientes** |

**5. Procedimientos más importantes en bases de datos**

Los procedimientos de la base de datos son compilados, almacenados y administrados por el servidor DBMS. Los procedimientos de la base de datos se pueden utilizar junto con las reglas para hacer cumplir la integridad de la base de datos o para realizar operaciones que se repiten con frecuencia. Cuando se crea el procedimiento se guarda su plan de ejecución, reduciendo la sobrecarga de ejecución.



<https://disenowebakus.net/bases-de-datos.php>

Los procedimientos de la base de datos se pueden crear de forma interactiva o en un programa integrado. Un procedimiento de base de datos se puede ejecutar en un programa de lenguaje *host,* en el monitor de terminal, en otro procedimiento de base de datos o en un programa 4GL. Los procedimientos de la base de datos también pueden invocarse mediante reglas.

**5.1. Índice de base de datos**

En este espacio conocerá todo sobre el índice de las bases de datos.

Slyders

DI\_CF019\_5.1 \_Índice de base de datos.

* **Índice agrupado ordenado según el nombre (tecla de búsqueda)**

**Indexación principal:** este es un tipo de indexación agrupada en el que los datos se ordenan de acuerdo con la clave de búsqueda y la clave principal de la tabla de la base de datos y se utiliza para crear el índice. Es un formato predeterminado de indexación donde induce la organización secuencial de archivos; como las claves primarias son únicas y se almacenan ordenadas, el rendimiento de la operación de búsqueda es bastante eficiente.

**Indexación secundaria o no agrupada:** un índice no agrupado simplemente dice dónde se encuentran los datos, es decir, da una lista de punteros virtuales o referencias a la ubicación donde se almacenan realmente los datos. Los datos no se almacenan físicamente en el orden del índice. En cambio, los datos están presentes en los nodos hoja, por ejemplo, la página de contenido de un libro; cada entrada da el número de página o la ubicación de la información almacenada. Los datos reales aquí (información en cada página del libro) no están organizados; pero se tiene una referencia ordenada (página de contenido) de dónde se encuentran realmente los puntos de datos. Solo se puede tener un orden denso en el índice no agrupado, ya que el orden disperso no es posible porque los datos no están organizados físicamente en consecuencia.

|  |
| --- |
| Requiere más tiempo en comparación con el índice agrupado porque se realiza una cierta cantidad de trabajo adicional para extraer los datos siguiendo más el puntero. En el caso de un índice agrupado los datos están directamente delante del índice.  https://ichi.pro/assets/images/max/724/1*XP1DJYiGYGQYS2MpGpESBA.png  <https://ichi.pro/es/el-diseno-de-indices-secundarios-globales-fuertemente-consistentes-en-apache-phoenix-parte-1-159124442202640> |

|  |
| --- |
| Indexación multinivel: con el crecimiento del tamaño de la base de datos los índices también crecen. Como el índice se almacena en la memoria principal, un índice de un solo nivel puede llegar a ser de un tamaño demasiado grande para almacenarlo con múltiples accesos al disco. La indexación multinivel segrega el bloque principal en varios bloques más pequeños para que el mismo pueda almacenarse en un solo bloque. Los bloques externos se dividen en bloques internos que, a su vez, apuntan a los bloques de datos. Esto se puede almacenar fácilmente en la memoria principal con menos gastos generales. Ver figura.  Figura 2  *Índice multinivel*    Nota. Tomado de <https://rua.ua.es/dspace/bitstream/10045/23898/2/T1B%20ficheros.pdf> |

|  |
| --- |
| Para mayor comprensión diríjase al Anexo 2 - Índices basados en varios campos utilizando Microsoft Access <https://docs.google.com/document/d/10awlM03bJA91vV4uUmInNTFgCqx4_RqY/edit?usp=sharing&ouid=103879860323491984918&rtpof=true&sd=true> y al Anexo 3 - Crear un índice y utilizarlo para mejorar el rendimiento en Microsoft Access <https://docs.google.com/document/d/1zEMgXdZwoDtGa1Ix7Kd8_rXsVQ-POIcD/edit?usp=sharing&ouid=103879860323491984918&rtpof=true&sd=true> |

**5.2. Vistas en base de datos**

Se invita a conocer un poco más sobre las bases de datos, en esta ocasión descubra todo sobre las vistas en las bases de datos.

Slyders

DI\_CF019\_5.2 \_Vistas en base de datos.

**5.3. Reportes o informes en bases de datos**

En el siguiente recurso podrá revisar los reportes o informes en las bases de datos, cómo realizarlos y aplicarlos.

Slyders

DI\_CF019\_5.3 \_Reportes o informes en base de datos.

**5.4. Permisos, roles y control de acceso de usuarios**

|  |
| --- |
| Se refiere a las diferentes autorizaciones que se deben establecer en una base de datos, en otras palabras, es la función de especificar derechos / privilegios de acceso a los recursos relacionados con la seguridad de la información. Más formalmente, "autorizar" es definir una política de acceso; con la gestión de identidad controlada, el control de acceso de usuario se concede en función de los permisos asignados a un rol.  Principio de mínimo privilegio  <https://www.cloudcenterandalucia.es/blog/el-principio-de-minimo-privilegio-como-base-de-una-estrategia-de-seguridad/> |

|  |
| --- |
| Cada aplicación protegida por el habilitador de la base de datos puede definir un conjunto de permisos, es decir, un conjunto de cosas que se pueden hacer dentro de la aplicación que gestiona la base de datos, por ejemplo, dentro de la aplicación, la capacidad de enviar un comando para desbloquear una puerta inteligente. De manera similar, la capacidad de enviar un comando para hacer sonar la alarma y la capacidad de alterar los precios.  https://www.cloudcenterandalucia.es/wp-content/uploads/2019/10/seguridad-1.jpg  <https://www.cloudcenterandalucia.es/blog/el-principio-de-minimo-privilegio-como-base-de-una-estrategia-de-seguridad/>  Estos permisos se agrupan en una serie de roles, por ejemplo, el desbloqueo de la puerta y la de activar una alarma podrían asignarse al rol de seguridad, lo que significa que los usuarios a los que posteriormente se les otorgue ese rol obtendrían ambos permisos. Los permisos pueden superponerse y asignarse a múltiples roles, también se asignan al rol de administración. |

A su vez, los usuarios u organizaciones serán asignados a uno o más roles: cada usuario obtendrá la suma de todos los permisos para cada rol que tenga. El concepto de un rol es desconocido para un usuario: solo conocen la lista de permisos que se les ha otorgado, no cómo se dividen los permisos dentro de una determinada aplicación. En resumen, los permisos son todas las acciones posibles que se pueden realizar a los recursos dentro de una aplicación, mientras que los roles son grupos de acciones que puede realizar un tipo de usuario de esa aplicación.

**Conceptos estándar de gestión de identidades**

Los siguientes objetos comunes se encuentran con la base de datos:

Tarjetas

DI\_CF019\_5.4 \_Permisos, roles y control de acceso de usuarios.

**Protección de los datos dentro de la base de datos**

La autorización es un permiso otorgado al usuario, programa o proceso para acceder a un objeto o conjunto de objetos. El tipo de acceso a datos otorgado a un usuario puede ser de solo lectura o lectura y escritura. Los privilegios especifican el tipo de operaciones del lenguaje de manipulación de datos (DML) como seleccionar, insertar, actualizar, eliminar, etc., que el usuario puede realizar con los datos. Los dos métodos mediante los cuales se realiza el control de acceso son mediante el uso de privilegios y roles.

|  |
| --- |
| Privilegios:  Son permisos para acceder a un objeto con nombre de una manera prescrita, por ejemplo, permiso para consultar una tabla. Se pueden otorgar privilegios para permitir que un usuario en particular se conecte a la base de datos (crear una sesión); seleccionar filas de la tabla de otra persona; o ejecutar el procedimiento almacenado de otra persona.  Privilegios de la base de datos:  Un privilegio es un derecho a ejecutar un tipo particular de declaración SQL o acceder al objeto de otro usuario. Algunos ejemplos de privilegios incluyen:   * El derecho a conectarse a la base de datos (crear una sesión). * El derecho para crear una mesa. * El derecho a seleccionar filas de la tabla de otro usuario. * El derecho a ejecutar el procedimiento almacenado de otro usuario.   Se otorgan privilegios a los usuarios para que puedan realizar las tareas necesarias para su trabajo. Debe otorgar un privilegio solo al usuario que lo requiera, absolutamente para realizar el trabajo necesario. La concesión excesiva de privilegios innecesarios puede comprometer la seguridad.  Privilegios del sistema:  Un privilegio del sistema es el derecho a realizar una acción en particular, en un tipo particular de objeto, por ejemplo, los privilegios para crear tablas y eliminar el OWS de cualquier tabla en una base de datos son privilegios del sistema. En muchos sistemas comerciales de administración de bases de datos hay cientos de privilegios de sistema distintos.  Privilegios de objeto:  Un privilegio de objeto es un privilegio o derecho para realizar una acción particular en una tabla, vista, secuencia, procedimiento, función o paquete específico, por ejemplo, el privilegio de eliminar filas de la tabla DEPT es un privilegio de objeto. Los privilegios de objeto otorgados para una tabla, vista, secuencia, procedimiento, función o paquete se aplican tanto si se hace referencia al objeto base por su nombre como si se utiliza un sinónimo. |

**Roles:**

Son mecanismos que se pueden utilizar para proporcionar autorización. Se puede otorgar un rol o un grupo de roles a una sola persona o un grupo de personas. Se puede otorgar un rol a su vez a otros roles. Al definir diferentes tipos de roles, los administradores pueden administrar los privilegios de acceso con mucha más facilidad.

Los sistemas de administración de bases de datos brindan una administración de privilegios fácil y controlada a través de roles. Los roles son grupos con nombre de privilegios relacionados que otorga a los usuarios u otros roles. Los roles están diseñados para facilitar la administración del sistema del usuario final y los privilegios de objetos. Las siguientes propiedades de los roles permiten una administración de privilegios más sencilla dentro de una base de datos:

* **Administración de privilegios reducidos:** en lugar de otorgar explícitamente el mismo conjunto de privilegios para varios usos, puede otorgar los privilegios a un grupo de usuarios relacionados con un rol. Entonces, solo se debe otorgar el rol a cada miembro del grupo.
* **Administración dinámica de privilegios:** si los privilegios de un grupo deben cambiar, solo se deben modificar los privilegios del rol.
* **Disponibilidad selectiva de privilegios:** puede habilitar o deshabilitar selectivamente los roles otorgados a un usuario. Esto permite un control específico de los privilegios de un usuario en cualquier situación dada.
* **Seguridad específica de la aplicación:** puede proteger el uso de roles con una contraseña. Las aplicaciones se pueden crear específicamente para habilitar un rol cuando se proporciona la contraseña correcta. Los usuarios no pueden habilitar el rol si no conocen la contraseña.

|  |
| --- |
| Para mayor comprensión diríjase al Anexo 4 - Concesión y revocación de privilegios y funciones usando comandos SQL <https://docs.google.com/document/d/1_iwUhC5lbBQsVjcg48wsv5rf2KNpxeVH/edit?usp=sharing&ouid=103879860323491984918&rtpof=true&sd=true>  y al Anexo 5 - Gestionar permisos de usuario en Microsoft Access <https://docs.google.com/document/d/1rwYvNb754xgZf-Xnc22ioWS4vC2NuMzh/edit?usp=sharing&ouid=103879860323491984918&rtpof=true&sd=true> |

**5.5. Optimización de bases de datos**

Por medio de la siguiente infografía sabrá cómo optimizar las bases de datos.

Infografía dinámica

DI\_CF019\_5.5 \_Optimización de bases de datos.

|  |
| --- |
| Para mayor comprensión diríjase al Anexo 6 - Usar el analizador de rendimiento para optimizar una base de datos de Microsoft Access <https://docs.google.com/document/d/1ZXzWmm861-TWsRFt0Le6mSIANq1Wt8MN/edit?usp=sharing&ouid=103879860323491984918&rtpof=true&sd=true>  y al Anexo 7 - Aumentar la velocidad de ejecución de Microsoft Access <https://docs.google.com/document/d/1shmE4h_uPAtblt48RaOo2c3-nkXPca-e/edit?usp=sharing&ouid=103879860323491984918&rtpof=true&sd=true> |

**5.6. Limpieza de la base de datos**

En este apartado descubrirá todo sobre la limpieza de una base de datos.

Cartas de diálogo

DI\_CF019\_5.5 \_Limpieza de la base de datos.

A continuación, se ofrece una descripción general de los pasos que deberá seguir para asegurarse de que sus datos estén limpios y sean utilizables:

Pasos

DI\_CF019\_5.6 \_Limpieza de la base de datos. (Pasos)

**5.7. Pruebas en la base de datos**

A continuación puede consultar las pruebas correspondientes.

Slyders

DI\_CF019\_5.7 \_Pruebas en la base de datos.

|  |
| --- |
| Actas: las transacciones son operaciones sensibles de la base de datos porque requieren que se realicen diferentes consultas con una recuperación adecuada de las fallas. Una transacción de base de datos estándar es seguir los principios de ACID.   * A de *atomicity* - atomicidad: debe completar todas las operaciones o no realizar ninguna operación en absoluto. * C de *consistency* - consistente: el estado de la base de datos siempre debe ser válido y se deben cumplir todas las restricciones. * I de *isolation* - aislamiento: cada transacción debe realizarse de forma aislada de otras transacciones que se ejecutan al mismo tiempo. El estado final de la base de datos debe ser como si las transacciones se ejecutaran secuencialmente. * D de *durability* - durabilidad: sin pérdida de datos por ningún motivo, una vez que se confirma la transacción.   El probador de la base de datos elabora consultas SQL para verificar y validar estas propiedades en las operaciones de transacción en la aplicación. Este tipo de prueba es especialmente importante para probar aplicaciones financieras. |

**Esquema:** es el modelo de la estructura de la base de datos y los evaluadores deben estar muy familiarizados con él. Los evaluadores pueden usar el comando DESC de SQL para revelar el esquema de la base de datos y asegurarse de que se ajuste a lo que espera la aplicación. Los evaluadores también usan expresiones regulares para validar los nombres de los campos de la tabla y verificar que el valor se ajuste al tipo de datos esperado.

**Disparadores:** los desencadenadores son como controladores de eventos en el código. Cuando ocurre una operación en una tabla (por ejemplo, cuando se agrega una nueva fila) se puede configurar un disparador para ejecutar un fragmento de código o consulta en respuesta a la operación que acaba de ocurrir. Los disparadores crean un efecto de cascada en algunas operaciones, por ejemplo, eliminar a un usuario de una base de datos puede provocar que las publicaciones del usuario también se eliminen. Como evaluador desea asegurarse de que esta operación se realice con precisión y no elimine las publicaciones de otro usuario.

Un evaluador puede ejecutar una consulta SQL para iniciar la operación de origen que desencadena la cascada. El probador también podría realizar la operación desde la interfaz de la aplicación para examinar su efecto en los registros de la base de datos.

**Otros:** las operaciones que describí anteriormente no son el único conjunto de operaciones que se pueden probar, son solo algunas de las más importantes para la mayoría de las aplicaciones sensibles a los datos. Otras operaciones y atributos que se pueden probar son:

* Restricciones de la base de datos.
* Procedimientos almacenados.
* Vista.
* Índices.

|  |
| --- |
| ¿Cómo probar las bases de datos?  Ahora que se sabe qué probar, ¿cómo se deben realizar las pruebas? Sorprendentemente, ejecutar las pruebas de las bases de datos no es tan diferente a ejecutar pruebas en aplicaciones. Como se mencionó anteriormente, estas pruebas se pueden realizar manualmente realizando operaciones desde la interfaz de usuario de la aplicación para asegurarse de que se devuelvan los datos correctos después de cada operación o verificando los registros de la base de datos para ver los resultados de la operación. También puede utilizar un DBMS para ejecutar sus consultas de prueba y validar los resultados. Si se prefiere no realizar pruebas manualmente, puede sentirse aliviado al saber que las pruebas automatizadas de bases de datos son muy similares a las pruebas automatizadas en el código de la aplicación. La principal diferencia es que esta vez sus pruebas ejecutan consultas en lugar de código de aplicación, también se aplican los siguientes pasos que se encuentran en el código de la aplicación de prueba:   * Prepare su entorno de prueba. * Ejecute sus pruebas con un corredor de prueba (por ejemplo, Selenium). * Verifique y valide sus resultados. * Informe sus afirmaciones.   Esta es la razón por la que el mejor atributo de un probador de bases de datos es poder crear consultas SQL para cada caso de prueba requerido. |

**Herramientas de prueba de las bases de datos**

Existen muchas herramientas de prueba de bases de datos. La herramienta que elija dependerá de uno o más de los siguientes:

* Estrategia de prueba (manual, automatizada o híbrida).
* Tipo de base de datos (SQL o No-SQL).
* Proveedor de base de datos (MySQL, MSSQL, Oracle u otro).

|  |
| --- |
| La mayoría de los sistemas de prueba de bases de datos se componen de más de una herramienta, por ejemplo, Selenium se puede utilizar con TestNG para pruebas de bases de datos en aplicaciones Java. De manera similar, SeLite usa Selenium para probar bases de datos SQLite, mientras que SQL Server viene con herramientas para bases de datos de pruebas unitarias.  Para mayor comprensión de este tema diríjase al Anexo 8 - Lista de herramientas de prueba de bases de datos  <https://docs.google.com/document/d/11QAJFwL-xSQJ2ja4fvmUQRpQMEYLDcRT/edit?usp=sharing&ouid=103879860323491984918&rtpof=true&sd=true> |

**6. MySQL**

|  |
| --- |
| MySQL es un verdadero servidor de base de datos SQL multiproceso y multiusuario. SQL es el lenguaje de bases de datos más popular del mundo. MySQL es una implementación cliente / servidor que consta de un dominio de servidor MySQLD y muchos programas y bibliotecas cliente diferentes.  comandos útiles de MySQL Server  <https://espaciotecnologico.co/comandos-utiles-de-mysql-server/>  SQL es un lenguaje estandarizado que facilita el almacenamiento, la actualización y el acceso a la información, por ejemplo, puede utilizar SQL para recuperar información de productos y almacenar información de clientes para un sitio web. MySQL también es lo suficientemente rápido y flexible como para permitirle almacenar registros e imágenes. |

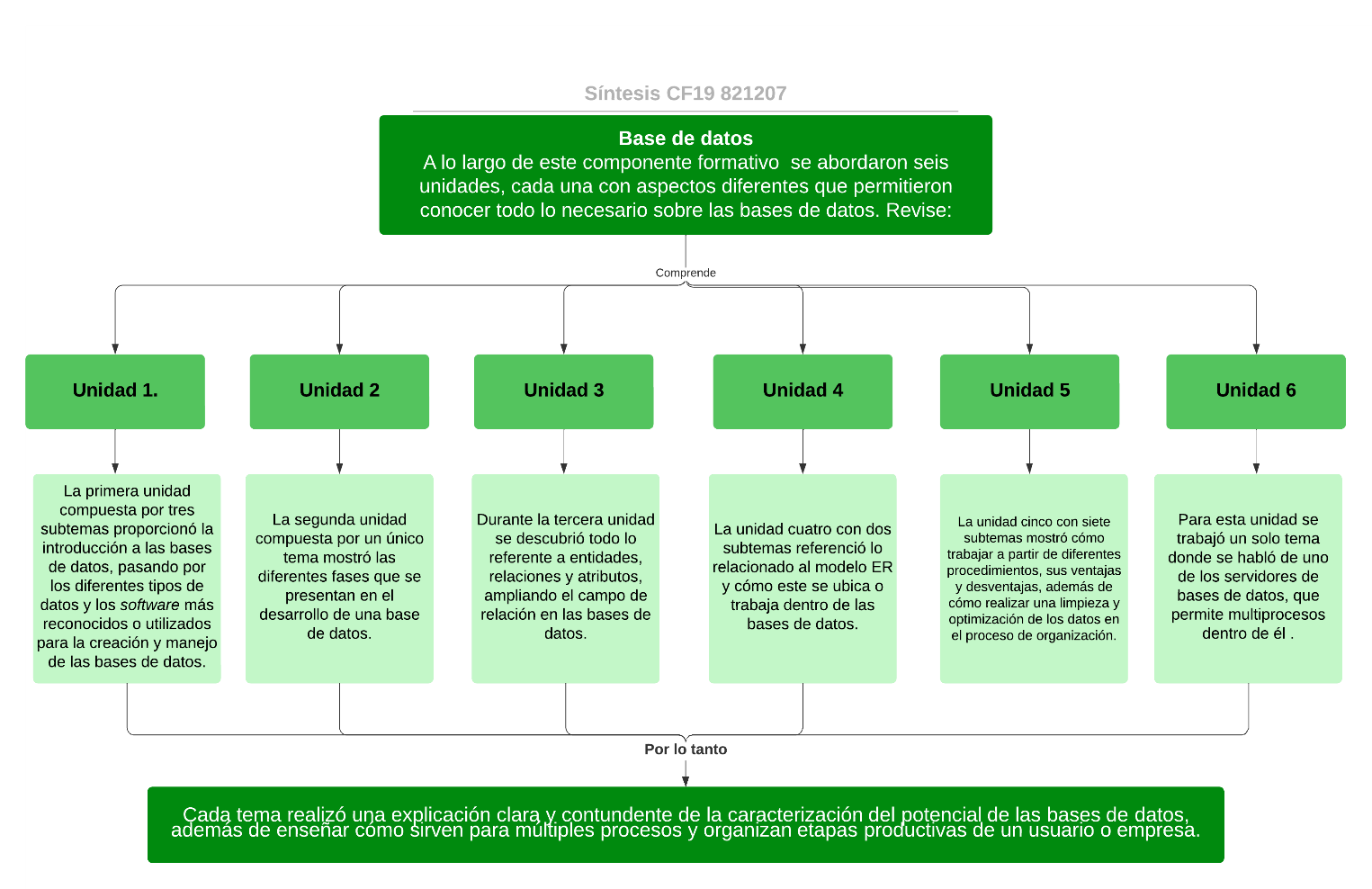
Los principales objetivos de MySQL son la velocidad, la solidez y la facilidad de uso. MySQL se desarrolló originalmente porque en TcX se necesitaba un servidor SQL que pudiera manejar bases de datos muy grandes en un orden de magnitud más rápido de lo que cualquier proveedor de bases de datos podría ofrecer. Ahora se ha estado usando MySQL desde 1996 en un entorno con más de 40 bases de datos que contienen 10.000 tablas, de las cuales más de 500 tienen más de 7 millones de filas. Se trata de unos 100 *gigabytes* de datos de misión crítica. La base sobre la que se construye MySQL es un conjunto de rutinas que se han utilizado en un entorno de producción muy exigente durante muchos años. Aunque MySQL todavía está en desarrollo, ya ofrece un conjunto de funciones poderosas y muy útil.

|  |
| --- |
| Para mayor comprensión de este tema diríjase al Anexo 9 - Más sobre MySQL <https://docs.google.com/document/d/1RWZLt1yUqcpsGXJztD9T4dPCpu3RX8_o/edit?usp=sharing&ouid=103879860323491984918&rtpof=true&sd=true> |

En el material complementario de este componente se recomiendan una serie de videos donde se explica paso a paso el diseño de una base de datos completa desde cero utilizando la aplicación Microsoft Access. Este material de apoyo amplía mucho más el conocimiento sobre base de datos y permite generar una gran experiencia de diseño.

**Síntesis**

El componente formativo bases de datos abarca diversos aspectos clave para su comprensión y aplicación. En primer lugar, se explora la introducción a las bases de datos, incluyendo la evolución y los diferentes tipos existentes, así como las herramientas de software más conocidas utilizadas en este ámbito. Luego, se analizan las fases del proceso de desarrollo de bases de datos, centrándose en la conceptualización de entidades, relaciones y atributos, y el modelo ER. Se examinan también los procedimientos fundamentales en bases de datos, como los índices, las vistas, los reportes o informes, los permisos y roles de usuario, la optimización y la limpieza de la base de datos, así como las pruebas necesarias. Finalmente, se destaca la importancia de MySQL, una popular herramienta de gestión de bases de datos ampliamente utilizada en la industria.

****

1. **ACTIVIDADES DIDÁCTICAS (OPCIONALES SI SON SUGERIDAS)**

|  |  |
| --- | --- |
| DESCRIPCIÓN DE LA ACTIVIDAD DIDÁCTICA | |
| Nombre de la actividad | Desafío de conocimiento “Base de datos” |
| Objetivo de la actividad | El objetivo de esta actividad didáctica evaluar el conocimiento y comprensión de los conceptos fundamentales en bases de datos de los aprendices a través de preguntas sobre los temas vistos en el componente. |
| Tipo de actividad sugerida | Cuestionario |
| Archivo de la actividad  (Anexo donde se describe la actividad propuesta) | <https://sena4.sharepoint.com/:f:/r/sites/EcosistemaRED/Dctos_Proceso/LP_Tolima/_Tecnologos/821207%20(Gesti%C3%B3n%20Energ%C3%ADa)/3.%20FinalxVirtualizar/Contenidos/CF019?csf=1&web=1&e=FrPpfu> |

1. **MATERIAL COMPLEMENTARIO**

|  |  |  |
| --- | --- | --- |
| Referencia APA del material | Tipo de material  (Video, capítulo de libro, artículo, otro) | Enlace del recurso o  archivo del documento o material |
| Edteam. (2020, 13, febrero). ¿Que son las bases de datos? [Archivo de video]. Recuperado de https://www.youtube.com/watch?v=knVwokXITGI | Video | https://www.youtube.com/watch?v=knVwokXITGI |
| Edteam. (2022, 9, septiembre). ¿SQL o NoSQL? ¿Cuál base de datos es mejor? [Archivo de video]. Recuperado de https://www.youtube.com/watch?v=8K1PKSqHFRQ | Video | https://www.youtube.com/watch?v=8K1PKSqHFRQ |

1. **GLOSARIO**

|  |  |
| --- | --- |
| TÉRMINO | SIGNIFICADO |
| ACIDO | El acrónimo de las propiedades mantenidas por sistemas de gestión de bases de datos estándar, que significa atomicidad, consistencia, aislamiento y durabilidad. |
| Atomicidad | La propiedad de una transacción que garantiza que todos o ninguno de los cambios realizados por la transacción se escriben en la base de datos. |
| AVL-Árbol | Es un árbol de búsqueda binaria autoequilibrado. |
| Caché | La memoria de la computadora que se reserva para contener una parte de los datos de la base de datos a los que el programa de aplicación de la base de datos ha accedido más recientemente. |
| Cascada | Un atributo de clave externa que migra automáticamente los cambios realizados a una referencia (es decir, clave primaria), tabla a todas las referencias (clave externa), filas de la tabla. |
| Columna | Una sola unidad de datos con nombre que tiene un tipo de datos en particular (por ejemplo, número, texto o fecha). Las columnas solo existen en tablas. |
| Compilador | Una herramienta de desarrollo de *software* que traduce programas de lenguaje de alto nivel a instrucciones en lenguaje de máquina, que un procesador en particular puede comprender y ejecutar. |
| DBMS | Un acrónimo de sistema de administración de base de datos. |
| Depurador | Una herramienta utilizada para probar y depurar *software.* Un depurador remoto típico se ejecuta en una computadora *host* y se conecta al destino a través de un puerto serie o en una red. |
| DML | Lenguaje de manipulación de bases de datos. En SQL, declaraciones como *update, insert y delete* se consideran DML. |
| Durabilidad | La propiedad de una transacción en la que el DBMS garantiza que todas las transacciones comprometidas sobrevivirán a cualquier tipo de falla del sistema. |
| E / S | De entrada y salida. Para un DBMS, normalmente se trata de una unidad de disco que se utiliza para crear una base de datos, durabilidad. |
| Gota | Abreviatura de Binary Large Object. En SQL, BLOB puede ser un término general para cualquier dato de tipo *varbinary* largo, *varchar* largo o largo *varchar.* |
| IEC | Comisión Electrotécnica Internacional. Junto con ISO, IEC controla el estándar SQL (ISO / IEC 9075) y muchos otros también. |
| IIOT | Abreviación de Internet industrial de las cosas. |
| Índice | Una estructura separada que permite un acceso rápido a las filas de una tabla en función de los valores de datos de las columnas utilizadas en el índice. |
| Java | Un lenguaje de programación multiplataforma, orientado a objetos, similar a C ++, que está disponible gratuitamente para todos y cada uno de los desarrolladores de *software*. |
| JDBC | API de conectividad de base de datos de Java. JDBC proporciona una API estándar de manipulación y acceso a la base de datos para programas Java. |
| JSON | Una representación de datos ofrecida como una alternativa más compacta; pero aún legible por humanos para XML. |
| Little-Endian | Es un tipo de direccionamiento que se refiere al orden de los datos almacenados en la memoria. |
| Marcas | Marcas comerciales, nombres comerciales, marcas de servicio o logotipos identificados en el sitio web y/o material impreso de una empresa. |
| Metadatos | "Datos sobre datos". En un contexto DBMS, los datos almacenados en columnas de una tabla tienen ciertos atributos, como el tipo, largo, descripción u otras características que permitan al DBMS procesar los datos de manera significativa o permitir que los usuarios los comprendan mejor. |
| Mesa | Una colección de columnas estrechamente relacionadas. Una tabla consta de filas, cada una de las cuales comparte las mismas columnas; pero varían en los valores de las columnas. |
| Pila | Es una estructura conceptual que consta de un conjunto de elementos homogéneos y se basa en el principio último en entrar, primero en salir (LIFO). |
| Replicación | Un proceso en el que las modificaciones seleccionadas en una base de datos maestra se replican (reproducen) en otra base de datos. |
| RTOS | Abreviatura común de sistema operativo en tiempo real. Raima Database Manager se ejecuta en la mayoría de RTOS, como VxWorks, Integrity, Linux integrado y QNX. |
| Servidor (*software*) | Un asiento que reside en una sola máquina servidor y es capaz de aceptar conexiones desde uno o más asientos que residen en las máquinas cliente. |
| Sincronización | El método de implementación (con frecuencia semáforos) mediante el cual la ejecución simultánea de múltiples subprocesos o procesos informáticos pueden acceder de forma segura y actualizar los datos compartidos. |
| SQL | Es el lenguaje comúnmente aceptado que se utiliza para definir, consultar y manipular una base de datos relacional. La etimología de "SQL" no está clara, posiblemente una progresión de "QueL" (Query Language) a "SeQueL" y a "SQL". |
| XML | Lenguaje de marcado extensible. Los documentos XML se utilizan mucho en la World Wide Web de Internet; pero también se utilizan en muchos contextos informáticos en los que es necesario compartir datos. |

1. **REFERENCIAS BIBLIOGRÁFICAS**

Aguilar, L. (s.f.). *Big data Análisis de grandes volúmenes de datos en organizaciones*. Alfaomega Grupo Editor.

Cowart, R. (1993). *El ABC de base.* Ventura Ediciones.

Cuadra, D. (s.f.). *Desarrollo de bases de datos. Casos prácticos desde el análisis a la implementación.* RA-MA.

Godoc, E. (s.f.). *Los fundamentos del lenguaje.* Eni.

Orbegozo, A. (2016). *Access 2016: Manual práctico paso a paso*. Alfaomega Grupo Editor.

Peña, R. (2013). *Access 2013: manual práctico para todos*. Alfaomega Grupo Editor.

1. **CONTROL DEL DOCUMENTO**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Nombre | Cargo | Dependencia  *(Para el SENA indicar Regional y Centro de Formación)* | Fecha |
| Autor | Jaime Mauricio Peñaloza Trespalacios | Experto técnico | Regional Distrito Capital - Centro Electricidad, Electrónica y Telecomunicaciones | Octubre de 2021 |
| [Leidy Carolina Arias Aguirre](mailto:leidyc.arias@misena.edu.co) | Diseñadora instruccional | Regional Distrito Capital - Centro de Diseño y Metrología | Noviembre de 2021 |
| Carolina Coca Salazar | Revisora metodológica y pedagógica | Regional Distrito Capital - Centro de Diseño y Metrología | Diciembre de 2021 |
| Rafael Neftalí Lizcano Reyes | Responsable Equipo desarrollo curricular | Regional Santander – Centro Industrial del Diseño y la Manufactura | Diciembre de 2021 |
| Julia Isabel Roberto | Correctora de estilo | Regional Distrito Capital – Centro de Diseño y Metrología | Junio de 2022 |
| Juan Gilberto Giraldo Cortés | Diseñador instruccional | Regional Tolima – Centro de Comercio y Servicios | Junio 2023 |
| María Inés Machado López | Metodólogo | Regional Tolima – Centro de Comercio y Servicios | Junio 2023 |

1. **CONTROL DE CAMBIOS**

**(Diligenciar únicamente si realiza ajustes a la Unidad Temática)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Nombre | Cargo | Dependencia | Fecha | Razón del Cambio |
| Autor (es) |  |  |  |  |  |