CONSULTA N°2 LENGUAJE DE PROGRAMACION 3

SANTIAGO RESTREPO ZULUAGA

UNIREMINGTON

RIONEGRO-ANTIOQUIA

**TERMINOLOGÍA BASICA DE BASES DE DATOS:**

Base de Datos: Un conjunto estructurado de datos relacionados entre sí, que se almacena en un sistema informático y se utiliza para administrar y acceder a la información.

Tabla: Una colección de datos organizados en filas y columnas. Cada fila representa un registro o entrada en la tabla, mientras que cada columna representa un atributo o campo.

Campo: Una unidad básica de datos en una tabla que almacena información específica, como nombres, direcciones, fechas, números, etc.

Registro: Un conjunto completo de información en una tabla que se refiere a un elemento específico, como un cliente, un producto, una factura, etc.

Clave primaria: Un campo o conjunto de campos que identifica de manera única cada registro en una tabla. Se utiliza para garantizar la integridad de los datos y permitir la identificación y actualización de registros específicos.

Clave foránea: Un campo o conjunto de campos en una tabla que se refiere a la clave primaria de otra tabla y se utiliza para establecer una relación entre ellas. Esto permite la creación de relaciones entre tablas y la recuperación de información de varias tablas.

Índice: Una estructura de datos que mejora la velocidad de búsqueda de registros en una tabla. Un índice se crea en una o varias columnas de una tabla y permite la búsqueda rápida de registros en función del valor de estas columnas.

Consulta: Una solicitud para recuperar datos específicos de una o más tablas. Las consultas se utilizan para seleccionar, filtrar, ordenar y agrupar datos según ciertos criterios.

Informe: Una representación visual de los datos recuperados de una base de datos, que se utiliza para la presentación o análisis de los mismos. Los informes se crean a partir de consultas y permiten la visualización de datos en formatos gráficos y tabulares.

Formulario: Una interfaz de usuario para ingresar, editar y ver datos en una tabla. Los formularios permiten la entrada de datos de manera más fácil y eficiente y se utilizan para garantizar la precisión y consistencia de los datos en una base de datos.

**MULTITAREA EN JAVA:**

La multitarea en Java es la capacidad de un programa para ejecutar varias tareas simultáneamente en un solo proceso. Esto se logra mediante el uso de hilos (threads) que son pequeñas unidades de procesamiento dentro de un programa. Cada hilo representa una tarea o proceso separado que se ejecuta de manera independiente de otros hilos en el mismo programa.

En Java, los hilos se crean utilizando la clase Thread y se pueden ejecutar de manera concurrente. Los hilos pueden ser utilizados para realizar tareas en segundo plano, para mantener una interfaz de usuario sensible y para realizar operaciones en paralelo, lo que puede mejorar el rendimiento y la eficiencia de un programa.

Java también proporciona una API de ejecución de hilos que permite la gestión de hilos en un entorno de aplicación. Esta API permite la creación, inicio y finalización de hilos, así como la sincronización y comunicación entre ellos.

Es importante tener en cuenta que la multitarea en Java debe ser cuidadosamente diseñada y gestionada para evitar problemas de concurrencia, como condiciones de carrera, bloqueos y deadlocks, que pueden afectar negativamente el rendimiento y la estabilidad de un programa.

**HILOS THREADS EN JAVA:**

En Java, los hilos (threads) son unidades de ejecución que permiten que un programa realice varias tareas de forma concurrente. Cada hilo representa una secuencia de instrucciones que se ejecuta de forma independiente del resto del programa, lo que permite que se realicen varias tareas al mismo tiempo.

La clase Thread es la clase principal que se utiliza para crear hilos en Java. Para crear un nuevo hilo, se crea una instancia de la clase Thread y se proporciona como argumento una instancia de una clase que implementa la interfaz Runnable. La interfaz Runnable define un método run() que contiene el código que se ejecutará en el hilo.

Por ejemplo, el siguiente código crea un nuevo hilo que imprime "Hola" en la consola cada segundo:

Thread thread = new Thread(new Runnable() {

public void run() {

while (true) {

System.out.println("Hola");

try {

Thread.sleep(1000);

} catch (InterruptedException e) {

e.printStackTrace();

}

}

}

});

thread.start();

El método start() inicia la ejecución del hilo. Cuando se llama a start(), se crea un nuevo hilo y se ejecuta el método run() definido en la clase Runnable. En el ejemplo anterior, el hilo se ejecuta indefinidamente, imprimiendo "Hola" cada segundo hasta que se detiene manualmente.

En resumen, los hilos permiten que un programa realice varias tareas de forma concurrente en Java, lo que puede mejorar el rendimiento y la eficiencia de la aplicación.

**REFERENCIAS:**

* Oracle. (2021). Oracle Database Concepts. Recuperado el 21 de abril de 2023, de https://docs.oracle.com/en/database/oracle/oracle-database/19/cncpt/index.html
* W3Schools. (s.f.). SQL Tutorial. Recuperado el 21 de abril de 2023, de https://www.w3schools.com/sql/
* Oracle. (2021). Java Tutorials. Recuperado el 21 de abril de 2023, de https://docs.oracle.com/javase/tutorial/
* Baeldung. (2021). Multithreading and Concurrency in Java. Recuperado el 21 de abril de 2023, de https://www.baeldung.com/java-multithreading-concurrency