PREGUNTAS TEORICAS

**¿Qué es un método en Java?**

Un método en java básicamente es una fracción de código dentro de nuestro sistema que permite realizar procedimientos o funciones específicas en nuestra aplicación.

**¿Cuáles son los cuatro tipos de métodos en Java?**

* Métodos sin retorno y sin parámetros
* Métodos sin retorno y con parámetros
* Métodos con retorno y sin parámetros
* Métodos con retorno y con parámetros

**¿Cuál es la diferencia entre un método con parámetros y sin retorno y un método sin parámetros y con retorno?**

Un método con parámetros y sin retorno realiza operaciones utilizando los valores de los parámetros, pero no devuelve ningún valor al invocador. Un método sin parámetros y con retorno no toma parámetros, pero calcula y retorna un valor que puede ser utilizado por el invocador.

**¿Cuál es la diferencia entre un método con parámetros y con retorno y un método con parámetros y sin retorno?**

Un método con parámetros y con retorno realiza operaciones utilizando los valores de los parámetros y luego devuelve un valor calculado usando return. Un método con parámetros y sin retorno realiza operaciones utilizando los valores de los parámetros, pero no devuelve ningún valor, y sus resultados pueden ser visibles a través de otros efectos secundarios o modificaciones realizadas en el entorno externo al método.

**¿Por qué son importantes los métodos en Java?**

Permiten una mejor utilización del código, haciéndolos más cortos y más eficientes. Ahorra tiempo y hace que el código sea más fácil de entender y mantener.

**¿Qué es una estructura de datos?**

Una estructura de datos es una forma organizada y eficiente de almacenar y gestionar datos, define cómo los datos se relacionan y almacenan para permitir un acceso rápido y operaciones efectivas.

**¿Cuál es la diferencia entre un arreglo, un arrayList y un HashMap?**

* Los arreglos son estáticos y de longitud fija, con acceso rápido pero limitado en la gestión del tamaño.
* ArrayLists son dinámicos, permiten cambios en el tamaño y proporcionan métodos convenientes, pero pueden consumir más memoria.
* HashMaps almacenan pares clave-valor, ofrecen búsquedas eficientes por clave y son útiles para asociar datos.