

## BATTISTA

### XML

El tipo de archivo XML, creado como un formato de datos de texto, no solo es legible a nivel de usuario, sino que también es comprensible a nivel de máquina y, por tanto, práctico para su uso en servicios web. XML es un lenguaje multiplataforma diseñado para almacenar varios tipos de datos. Destaca por su simplicidad, usabilidad y generalidad, es tan popular como el lenguaje HTML y se distribuye extensamente a través de internet. La popularidad de este formato se debe principalmente a que los archivos XML pueden modificarse fácilmente con un editor de texto.

### Encapsulacion

Java, como un lenguaje orientado a objetos, implementa la encapsulación. Este concepto consiste en la ocultación del estado o de los datos miembro de un objeto, de forma que sólo es posible modificar los mismos mediante los métodos definidos para dicho objeto.

Cada objeto está aislado del exterior, de forma que la aplicación es un conjunto de objetos que colaboran entre sí mediante el paso de mensajes invocando sus operaciones o métodos. De esta forma, los detalles de implementación permanecen "ocultos" a las personas que usan las clases, evitando así modificaciones o accesos indebidos a los datos que almacenan las clases. Además, el usuario de la clase no se tiene que preocupar de cómo están implementados los métodos y propiedades, concentrándose sólo en cómo debe usarlos. La encapsulación es una de las principales ventajas que proporciona la programación orientada a objetos.

### Interfaz (Java)

Ir a la navegaciónIr a la búsquedaUna interfaz en Java es una colección de métodos abstractos y propiedades constantes. En las interfaces se especifica qué se debe hacer pero no su implementación. Serán las clases que implementen estas interfaces las que describen la lógica del comportamiento de los métodos. La principal diferencia entre interface y abstract es que un interface proporciona un mecanismo de encapsulación de los protocolos de los métodos sin forzar al usuario a utilizar la herencia.

#### \*Ventajas

El uso de las interfaces Java proporciona las siguientes ventajas: Organizar la programación.

permiten declarar constantes que van a estar disponibles para todas las clases que queramos (implementando esa interfaz)

Obligar a que ciertas clases utilicen los mismos métodos (nombres y parámetros). Establecer

relaciones entre clases que no estén relacionadas.

// Interface

```
interface Animal {
```

```
    public void animalSound(); // interface method (does not have a body)
```

```
    public void sleep(); // interface method (does not have a body)
```

```
}
```

```
class Pig implements Animal {
```

```
    public void animalSound() {
```

```
        // The body of animalSound() is provided here
```

```
        System.out.println("The pig says: wee wee");
```

```
    }
```

```
    public void sleep() {
```

```
        // The body of sleep() is provided here
```

```
        System.out.println("Zzz");
```

```
    }
```

```
}
```

```
class MyMainClass {
```

```
    public static void main(String[] args) {
```

```
        Pig myPig = new Pig();
```

```
        myPig.animalSound();
```

```
        myPig.sleep();
```

```
    }
```

```
}
```

