





# ÍNDICE

## CLASES ABSTRACTAS E INTERFACES

1. Clases abstractas .....	3
2. Interfaces .....	3
3. Polimorfismo e interfaces .....	4



## Clases abstractas e interfaces

### 1. Clases abstractas

Una clase abstracta es aquella que puede incluir métodos abstractos, es decir métodos que están declarados en la clase pero no están implementados.

#### Características de las clases abstractas

1. No es posible crear objetos de una clase abstracta
2. Sirven para modelar conceptos.
3. Es una clase diseñada para actuar como base de otras clases derivadas en la cual se pone la funcionalidad común de los objetos derivados.
4. Una clase abstracta puede tener constructores los cuales serán obligatorios para las derivadas.

#### Métodos abstractos

Es un método declarado pero no implementado. Su intención es definir un formato para el método y dejar la implementación para las subclases.

Los métodos estáticos no pueden ser abstractos

Si una clase tiene un método abstracto sus derivadas deben implementar el método abstracto o seguirán siendo clases abstractas

Solo las clases abstractas pueden tener métodos abstractos.

#### Polimorfismo

El polimorfismo consiste en poder aplicar un mismo grupo de instrucciones sobre diferentes tipos de objetos, proporcionando así una gran reutilización de código.

### 2. Interfaces

Es conjunto de métodos abstractos que deberán ser implementados por clases.

#### Características de una interfaz

1. Sus métodos son abstractos no es obligatorio indicar el modificador abstract en su declaración.
2. Genera relaciones entre clases que de otro modo no están relacionadas
3. Distintas clases pueden implementar las mismas interfaces y a la vez derivar de otras clases.

## Clases abstractas e interfaces

4. Una clase puede implementar más de una interface.
5. Una clase que implementa una interface tiene que implementar todos sus métodos o bien declararse como abstracta.

### 3. Polimorfismo e interfaces

El polimorfismo también puede ser aplicado con interfaces, pues en una variable de tipo interfaz, se puede almacenar cualquier objeto de las clases que la implementan.

Ejemplo

```
Television tv = new Sony();
```

Y ejecutar:

```
tv.encender();
```

De la misma forma podríamos asignar a la variable de la interfaz un objeto de otra clase que la implemente y realizar la llamada a sus métodos utilizando la misma expresión:

```
tv = new Panasonic();
```

```
tv.encender();
```