



UNIVERSIDAD
Popular del Cesar

Ingeniería de Sistemas

Patrones de Diseño de Software

Profesor: Deivis Martínez Acosta

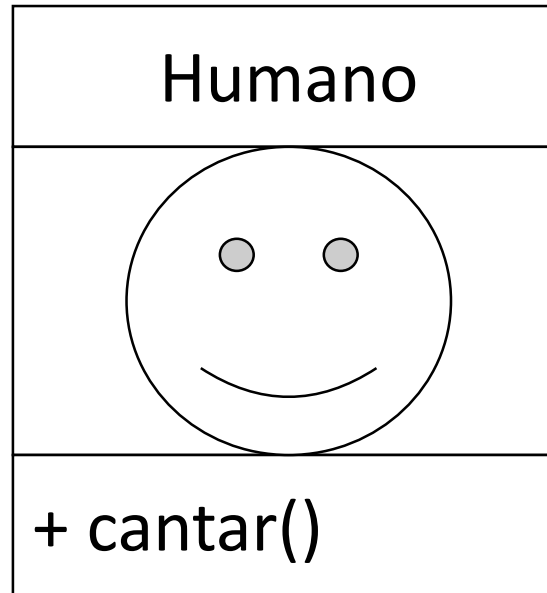


Clases Abstractas e Interfaces

Vicente



```
cantar(){"Rancheras";}
```



Shakira




```
cantar(){"Champeta";}
```



Clases Abstractas


Las clases abstractas son aquellas que definen la estructura de una aplicación con la existencia de métodos pero no describen la implementación que se dará a cada uno de estos.





Clases Abstractas


Una clase abstracta por lo menos posee un método abstracto, un método abstracto es aquel que solo tiene el qué se debe hacer, pero no especifica el cómo se debe hacer.





Clases Abstractas

Su uso es destacado en el polimorfismo, donde un método abstracto dice qué se debe hacer y cada clase que hereda de ella lo implementa de la forma que lo requiera.



Clases Abstractas

Las clases abstractas pueden tener métodos no abstractos. Una clase totalmente abstracta es aquella que no se puede instanciar, y se usa únicamente para definir subclases, para este tipo de clases se recomienda el uso de interfaces.

```
public abstract class Humanos {}.
```

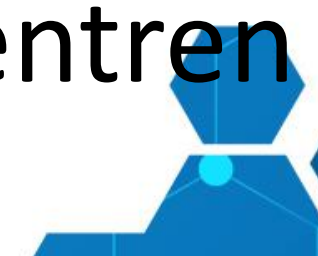
Clases Abstractas

```
public abstract class Humano {  
    public abstract void cantar();  
    public abstract void pintar();  
    public void comer(){  
        System.out.println("Comer Frutas");  
    }  
}
```



Clases Abstractas

Las subclases de una clase abstracta deben sobre-escribir por obligación los métodos abstractos y definir en ellos el cómo deben funcionar estos. Al igual pueden hacer uso o no de los otros métodos heredables que se encuentren en la clase padre.




Clases Abstractas

```
public class Shakira extends Humano{  
    @Override  
    public void cantar() {  
        System.out.println("Champeta");  
    }  
    @Override  
    public void pintar() {  
        System.out.println("Casas");  
    }  
}
```



Interfaces


Las Interfaces son aquellas que permiten al usuario crear una abstracción completa, donde todos sus métodos son abstractos y estos son implementados en las clases que lo desee. Es decir una interface se convierte en una plantilla de clases, o un molde del cual se implementan las clases.





Interfaces

Para usar una interface no tenemos que usar herencia, podemos hacerlo con implement y además del número de interface que necesitemos separándolas por coma, mientras que de las clases abstractas solo podemos heredar de una sola en el caso de “JAVA”.



Interfaces

```
public interface Humano {  
    public void cantar();  
    public void pintar(String pincel);  
    public void hablar(String idioma);  
}
```



UNIVERSIDAD
Popular del Cesar

Vamos a la práctica