**1. Clases**

**Definición:** Una clase es una plantilla para crear objetos. Define propiedades (atributos) y métodos (funciones) que los objetos creados a partir de la clase tendrán.

**Características:**

* **Instanciación:** Puedes crear instancias (objetos) de una clase.
* **Implementación:** Proporciona una implementación concreta para sus métodos.
* **Herencia:** Una clase puede heredar de otra clase, obteniendo así sus atributos y métodos.

**Ejemplo en Java:**

public class Animal {

private String nombre;

public Animal(String nombre) {

this.nombre = nombre;

}

public void hacerSonido() {

System.out.println("El animal hace un sonido.");

}

public String getNombre() {

return nombre;

}

}

**2. Clases Abstractas**

**Definición:** Una clase abstracta es una clase que no se puede instanciar directamente. Está diseñada para ser una base para otras clases. Contiene métodos abstractos (sin implementación) que las clases derivadas deben implementar.

**Características:**

* **No Instanciable:** No puedes crear instancias de una clase abstracta directamente.
* **Métodos Abstractos:** Puede contener métodos abstractos que las subclases deben implementar.
* **Implementación Parcial:** Puede contener métodos con implementación concreta.

**Ejemplo en Java:**

abstract class Animal {

private String nombre;

public Animal(String nombre) {

this.nombre = nombre;

}

public abstract void hacerSonido(); // Método abstracto

public String getNombre() {

return nombre;

}

}

class Perro extends Animal {

public Perro(String nombre) {

super(nombre);

}

@Override

public void hacerSonido() {

System.out.println("El perro ladra.");

}

}

**3. Interfaces**

**Definición:** Una interfaz es un contrato que define un conjunto de métodos que una clase debe implementar. Las interfaces no contienen implementación; solo declaran métodos.

**Características:**

* **Implementación Obligatoria:** Las clases que implementan una interfaz deben proporcionar una implementación para todos sus métodos.
* **Multihilo de Implementación:** Una clase puede implementar múltiples interfaces.
* **No contiene Estado:** Las interfaces no pueden tener atributos de instancia.

**Ejemplo en Java:**

interface Animal {

void hacerSonido();

}

class Perro implements Animal {

@Override

public void hacerSonido() {

System.out.println("El perro ladra.");

}

}

**Resumen**

* **Clases:** Pueden ser instanciadas y pueden tener una implementación completa de métodos.
* **Clases Abstractas:** No se pueden instanciar directamente, pueden tener métodos abstractos (sin implementación) y métodos concretos (con implementación).
* **Interfaces:** Solo definen un contrato que las clases deben cumplir, sin implementación propia. En Java, las interfaces se usan para definir qué métodos deben estar presentes, mientras que en Python se puede usar “Protocol” para un propósito similar.