

PROGRAMACIÓN II

Trabajo Práctico 7: Herencia & Polimorfismo

25/10/2025

Alumno:

- Federico Garcia Bengolea - feddericogarciaa@gmail.com
- Comisión: M2025-14
- Link GitHub: <https://github.com/FeddericoGarcia/utn-tupad-p2>

Profesores:

- **Profesor:** Alberto Cortez
- **Tutor:** Ramiro Hualpa

Caso Práctico:

Desarrollar las siguientes Katas en Java aplicando herencia y polimorfismo. Se recomienda repetir cada kata para afianzar el concepto.

1. Vehículos y herencia básica

```
package javaapp.TP7.Vehiculo;

public class Main {

    public static void main(String[] args) {
        Auto a = new Auto("Ferrari", "2021", 2);

        a.mostrarInfo();
    }
}

package javaapp.TP7.Vehiculo;

public class Vehiculo {

    protected String marca, modelo;

    public Vehiculo(String marca, String modelo) {
        this.marca = marca;
        this.modelo = modelo;
    }

    public void mostrarInfo(){
        System.out.println("Marca: " + marca + "\nModelo: " + modelo);
    }
}
```

```
package javaapp.TP7.Vehiculo;

public class Auto extends Vehiculo{

    private int cantidadPuertas;

    public Auto(String marca, String modelo, int cantidadPuertas) {
        super(marca, modelo);
        this.cantidadPuertas = cantidadPuertas;
    }

    @Override
    public void mostrarInfo(){
        System.out.println("Marca: "+ this.marca + "\nModelo: "+ this.modelo +
            "\nCantidad de puertas: "+ cantidadPuertas);
    }
}
```

```
Marca: Ferrari
Modelo: 2021
Cantidad de puertas: 2
BUILD SUCCESSFUL (total time: 0 seconds)
```

2. Figuras geométricas y métodos abstractos

```
package javaapp.TP7.FigurasGeometricas;

import java.util.ArrayList;

public class Main {

    public static void main(String[] args) {
        ArrayList<Figura> listaFiguras = new ArrayList<>();

        Circulo c1 = new Circulo("Circulo #1",4);
        Circulo c2 = new Circulo("Circulo #2",8);
        Rectangulo r1 = new Rectangulo("Rectangulo #1",3,5);
        Rectangulo r2 = new Rectangulo("Rectangulo #1",2,8);

        listaFiguras.add(c1);
        listaFiguras.add(c2);
        listaFiguras.add(r1);
        listaFiguras.add(r2);

        for (Figura f : listaFiguras){
            f.calcularArea();
        }
    }
}
```

```
package javaapp.TP7.FigurasGeometricas;

public abstract class Figura {
    protected String nombre;

    public Figura(String nombre) {
        this.nombre = nombre;
    }

    public void calcularArea(){};
}
```

```
package javaapp.TP7.FigurasGeometricas;

import static java.lang.Math.PI;

public class Circulo extends Figura {
    private double radio;
    public Circulo(String nombre, double radio) {
        super(nombre);
        this.radio = radio;
    }

    @Override
    public void calcularArea() {
        System.out.println(nombre.toUpperCase() +
            "\n Area: " + (radio * PI));
    }
}
```

```
package javaapp.TP7.FigurasGeometricas;

public class Rectangulo extends Figura{
    private double base, altura;

    public Rectangulo(String nombre, double base, double altura) {
        super(nombre);
        this.base = base;
        this.altura = altura;
    }

    @Override
    public void calcularArea() {
        System.out.println(nombre.toUpperCase() +
            "\n Area: " + (base * altura));
    }
}
```

```
CIRCULO #1
Area: 12.566370614359172
CIRCULO #2
Area: 25.132741228718345
RECTANGULO #1
Area: 15.0
RECTANGULO #1
Area: 16.0
BUILD SUCCESSFUL (total time: 0 seconds)
```

3. Empleados y polimorfismo

```
package javaapp.TP7.Empleados;
import java.util.ArrayList;

public class Main {

    public static void main(String[] args) {
        ArrayList<Empleado> listaEmpleados = new ArrayList<>();

        Empleado e1 = new EmpleadoTemporal();
        Empleado e2 = new EmpleadoFijo();
        Empleado e3 = new EmpleadoTemporal();
        Empleado e4 = new EmpleadoFijo();

        listaEmpleados.add(e1);
        listaEmpleados.add(e2);
        listaEmpleados.add(e3);
        listaEmpleados.add(e4);

        for (int i = 0; i < listaEmpleados.size(); i++) {
            Empleado e = listaEmpleados.get(i);
            System.out.println("Empleado: " + (i+1) +
                               " - Sueldo: " + e.calcularSueldo());
        }
    }
}
```

```
package javaapp.TP7.Empleados;

public abstract class Empleado {
    public double calcularSueldo(Empleado e){
        if (e instanceof EmpleadoFijo){
            return 1500000.0;
        } else if (e instanceof EmpleadoTemporal){
            return 1000000.0;
        } else {
            return 0;
        }
    }
}
```

```
package javaapp.TP7.Empleados;

public class EmpleadoTemporal extends Empleado {
}
```

```
package javaapp.TP7.Empleados;

public class EmpleadoFijo extends Empleado{
}
```

```
Empleado: 1 - Sueldo: 1000000.0
Empleado: 2 - Sueldo: 1500000.0
Empleado: 3 - Sueldo: 1000000.0
Empleado: 4 - Sueldo: 1500000.0
BUILD SUCCESSFUL (total time: 0 seconds)
```

4. Animales y comportamiento sobrescrito

```
package javaapp.TP7.Animales;
import java.util.ArrayList;

public class Main {

    public static void main(String[] args) {
        ArrayList<Animal> listAnimales = new ArrayList<>();

        Animal a1 = new Gato();
        Animal a2 = new Perro();
        Animal a3 = new Vaca();

        listAnimales.add(a1);
        listAnimales.add(a2);
        listAnimales.add(a3);

        for (Animal a : listAnimales){
            a.describirAnimal();
            a.hacerSonido();
            System.out.println("----");
        }

    }
}
```

```
package javaapp.TP7.Animales;

public class Animal {
    public void hacerSonido(){};
    public void describirAnimal(){};
}
```

```
package javaapp.TP7.Animales;

public class Gato extends Animal{

    @Override
    public void describirAnimal() {
        System.out.println("Soy un Gato");
    }

    @Override
    public void hacerSonido() {
        System.out.println("MIAUUU");
    }

}
```

```
package javaapp.TP7.Animales;

public class Perro extends Animal {

    @Override
    public void describirAnimal() {
        System.out.println("Soy un Perro");
    }

    @Override
    public void hacerSonido() {
        System.out.println("GUAUUU");
    }
}
```

```
package javaapp.TP7.Animales;

public class Vaca extends Animal{

    @Override
    public void describirAnimal() {
        System.out.println("Soy una Vaca");
    }

    @Override
    public void hacerSonido() {
        System.out.println("MUUUUUUUUU");
    }
}
```

```
Soy un Gato
MIAUUU
----
Soy un Perro
GUAUUU
----
Soy una Vaca
MUUUUUUUUU
----
BUILD SUCCESSFUL (total time: 0 seconds)
```