Nicolás Olima

nicolima200@gmail.com

* [REPO GITHUB](https://github.com/nicolima200/TPs_Progra2doCuatrimestre/tree/0d351dfbed2e61e2294b55da6e0c5b1f288f09ef/TP%207)

RESUMEN

Aplicación de los conceptos de herencia y polimorfismo en la Programación Orientada a Objetos: reutilización de código, creación de jerarquías de clases y diseño flexible de soluciones en Java.

PROGRAMACIÓN II

TRABAJO PRÁCTICO 7: HERENCIA Y POLIMORFISMO EN JAVA

**PROGRAMACIÓN II**

**Trabajo Práctico 7: Herencia y**  **Polimorfismo en Java**



**TECNICATURA**



**UNIVERSITARIA**



**EN**



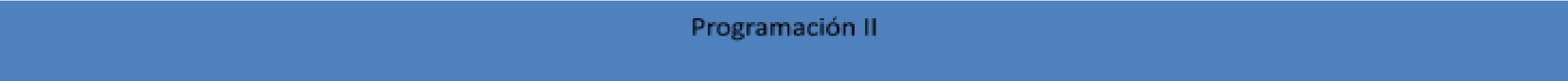
**PROGRAMACIÓN**



**A**



**DISTANCIA**



# 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
   * Clase base: Vehículo con atributos marca, modelo y método **mostrarInfo()**
   * Subclase: Auto con atributo adicional **cantidadPuertas**, sobrescribe **mostrarInfo()**
   * Tarea: Instanciar un auto y mostrar su información completa.

A computer code with text and numbers

AI-generated content may be incorrect.

A screen shot of a computer code

AI-generated content may be incorrect.

A blue screen with white text

AI-generated content may be incorrect.

SALIDA POR CONSOLA:



1. Figuras geométricas y métodos abstractos
   * Clase abstracta: Figura con método **calcularArea()** y atributo nombre
   * Subclases: **Círculo y Rectángulo** implementan el cálculo del área
   * Tarea: Crear un array de figuras y mostrar el área de cada una usando polimorfismo.

A screen shot of a computer code

AI-generated content may be incorrect.

A screen shot of a computer code

AI-generated content may be incorrect.

A computer screen shot of code

AI-generated content may be incorrect.

A screenshot of a computer code

AI-generated content may be incorrect.

SALIDA POR CONSOLA:

A screen shot of numbers

AI-generated content may be incorrect.

1. Empleados y polimorfismo
   * Clase abstracta: Empleado con método **calcularSueldo()**
   * Subclases: **EmpleadoPlanta, EmpleadoTemporal**
   * Tarea: Crear lista de empleados, invocar **calcularSueldo()** polimórficamente, usar instanceof para clasificar



**TECNICATURA**



**UNIVERSITARIA**



**EN**



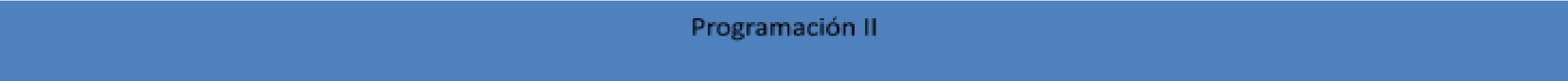
**PROGRAMACIÓN**



**A**



**DISTANCIA**



A screen shot of a computer code

AI-generated content may be incorrect.





A screen shot of a computer program

AI-generated content may be incorrect.

SALIDA POR CONSOLA:

A screenshot of a computer code

AI-generated content may be incorrect.

1. Animales y comportamiento sobrescrito
   * Clase: Animal con método **hacerSonido() y describirAnimal()**
   * Subclases: Perro, Gato, Vaca sobrescriben **hacerSonido()** con **@Override**
   * Tarea: Crear lista de animales y mostrar sus sonidos con polimorfismo

A screen shot of a computer code

AI-generated content may be incorrect.

A screen shot of a computer

AI-generated content may be incorrect.

A screen shot of a computer

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.

A screen shot of a computer code

AI-generated content may be incorrect.

SALIDA POR CONSOLA:

A black background with white text

AI-generated content may be incorrect.

# CONCLUSIONES ESPERADAS

* Comprender el mecanismo de herencia y sus beneficios para la reutilización de código.
* Aplicar polimorfismo para lograr flexibilidad en el diseño de programas.
* Inicializar objetos correctamente usando **super** en constructores.
* Controlar el acceso a atributos y métodos con modificadores adecuados.
* Identificar y aplicar **upcasting, downcasting** y **instanceof** correctamente.
* Utilizar clases y métodos abstractos como base de jerarquías lógicas.
* Aplicar principios de diseño orientado a objetos en la implementación en Java