

	<b>Carátula para entrega de prácticas</b>	
Facultad de Ingeniería	Laboratorios de docencia	

# Laboratorio de Computación Salas A y B

*Profesor(a):* JOSE ANTONIO AYALA BARBOSA

*Asignatura:* Programación Orientada a Objetos

*Grupo:* 02

*No de Práctica(s):* 04 Clases y Objetos

*Integrante(s):* Hernández Reyes Rebeca Sarai

*No. de lista o brigada:*

*Semestre:* 2026-1

*Fecha de entrega:* 17 – 09 – 2025

*Observaciones:*

CALIFICACIÓN: \_\_\_\_\_

## PRÁCTICA 4 CLASES Y OBJETOS

### PROGRAMACIÓN ORIENTADA A OBJETOS

Hernández Reyes Rebeca Sarai

#### OBJETIVO

Aplicar los conceptos básicos de la programación orientada a objetos en un lenguaje de programación.

#### PREVIO:

Clase: Perro	Objeto: perro1
numOrejas:int numPatas:int color:String raza:String nombre:String	2 4 "blanco" "maltes" "Ougus"
ladrar():void comer():void correr():void jugar():void	

Clase: Carro	Objeto: carro1
numPuertas:int numLlantas:int color: String marca:String modelo:String	4 4 "blanco" "Toyota" "Corolla"
avanzar():void luces():void reversa():void sonar():void	

Clase: TrianguloRec	Objeto: trianguloRec
base:double altura:double	4 5
area():void	

Clase: Alumno	Objeto: rebeca
numCuenta:double nombre:String edad:int carrera:String	321491864 "Rebeca" 20 "Ingeniería en computación"
estudiar():void comer():void hacerTareas():void dormir():void	

Clase: Profesor	Objeto: carlos
numCuenta:double numMaterias:int nombre:String estudios:String	12345678 3 "Carlos" "Maestría"
calificar():void prepararClase():void comer():void dormir():void	

## INTRODUCCIÓN

En esta práctica se refuerzan los fundamentos de POO trabajando con clases, objetos, constructores y métodos. Se implementan cinco clases sencillas (*Punto*, *Carro*, *Profesor*, *TrianguloRectangulo* y *Perro*) con atributos y comportamientos propios, y se prueba su funcionamiento desde la clase Poop4 (método main). El enfoque está en comprender cómo se modela un objeto con datos y acciones, cómo se construye con diferentes constructores y cómo se interactúa con él desde el programa cliente.

## ACTIVIDADES

- Crear clases.
- Crear objetos o instancias.
- Invocar métodos.
- Utilizar constructores
- Actividades extra

- Creación de clases restantes

## CÓDIGO FUENTE Y DESARROLLO

Se desarrolló un programa en el que se instanciaron dos puntos: uno vacío (que imprime (0,0) antes de asignar valores) y otro con parámetros (10,88).

```

11 public class Punto {
12     int x;
13     int y;
14
15     //Métodos
16     //Metodo constructor
17     public Punto(){
18         //x=0;
19         //y=0;e como inicializarlo, no es necesario ponerlos, el constructor vacio ya lo hace
20
21     }
22     public Punto(int x, int y){
23         this.x= x;
24         this.y= y;
25     }
26
27
28     //Metodo de acción
29     public void imprimirPunto(){
30         System.out.println("(" + x + "," + y + ")");
31     }
32 }

```

[Imagen 1. Punto.java]

```

Punto punto1 = new Punto();
punto1.imprimirPunto();
punto1.x=4;
punto1.y=7;
punto1.imprimirPunto();

Punto punto2 = new Punto(x:10,y:88);
punto2.imprimirPunto();

System.out.println("Punto 2 x = " + punto2.x);

```

[Imagen 2. Punto con valores en el main.]

Punto
x: int y: int
imprimirPunto():void

[Tabla 1. Punto.]

Seguido, se creó una clase Carro con sus métodos y atributos, para luego en el main crear un Carro vacío al que se asignaron valores y otro inicializado directamente con parámetros.

```

11 public class Carro {
12     int anio;
13     int numPuertas;
14     String marca;
15     String color;
16
17     public Carro(){
18
19     }
20
21     public Carro(int anio, int numPuertas, String marca, String color){
22         this.anio=anio;
23         this.numPuertas=numPuertas;
24         this.marca=marca;
25         this.color=color;
26
27     }

```

[Imagen 3. Carro atributos.]

```

29     public void avanzar(){
30         System.out.println(x:"Carro avanza");
31     }
32     public void frenar(){
33         System.out.println(x:"Carro frena");
34     }
35     public void sonar(){
36         System.out.println(x:"Pip Pip");
37     }
38     public void retroceder(){
39         System.out.println(x:"Carro retrocede");
40     }
41
42
43     public void imprimirCarro(){
44         System.out.println("Año: " + anio);
45         System.out.println("Numero de puertas: " + numPuertas);
46         System.out.println("Marca: " + marca);
47         System.out.println("Color: " + color);
48     }
49
50 }
51

```

[Imagen 4. Carro métodos.]

```

Carro carro1 = new Carro();
carro1.imprimirCarro();
carro1.anio= 2020;
carro1.numPuertas = 4;
carro1.marca= "Honda";
carro1.color= "blanco";
carro1.imprimirCarro();

Carro carro2 = new Carro (anio:2022, numPuertas:4, marca:"Chevrolet", color:"Azul");
carro2.imprimirCarro();

```

[Imagen 5. Carro con valores en el main]

Carro
marca: String anio: int color: String numPuertas: String
avanzar():void frenar():void sonar():void retroceder():void

[Tabla 2. Carro.]

Para el siguiente, y último en clase, se realizó la clase Profesor, en el que igual que en el anterior, se crearon dos objetos: uno vacío con valores asignados después, y otro con los parámetros cargados desde el inicio.

```

public class Profesor {
    String materiaQueImparte;
    double horarioDeClase;
    String divisionAlaQuePertenece;
    String universidadEnLaQueTrabaja;

    public Profesor(){
    }

    public Profesor(String materiaQueImparte, double horarioDeClase, String divisionAlaQuePertenece, String universidadEnLaQueTrabaja){
        this.materiaQueImparte = materiaQueImparte;
        this.horarioDeClase = horarioDeClase;
        this.divisionAlaQuePertenece = divisionAlaQuePertenece;
        this.universidadEnLaQueTrabaja = universidadEnLaQueTrabaja;
    }
}

```

[Imagen 6. Profesor atributos.]

```

public void pasarLista(){
    System.out.println("Pasando lista...");
}
public void calificarTareas(){
    System.out.println("Calificando tareas...");
}
public void hacerExámenes(){
    System.out.println("Haciendo exámenes fáciles...");
}
public void imprimirProfesor(){
    System.out.println("Materia que imparte: " + materiaQueImparte);
    System.out.println("Horario de clase: " + horarioDeClase);
    System.out.println("División a la que pertenece: " + divisionALaQuePertenece);
    System.out.println("Universidad en la que trabaja: " + universidadEnLaQueTrabaja);
}

```

[Imagen 7. Profesor métodos]

```

Profesor profesor1 = new Profesor();
profesor1.imprimirProfesor();
profesor1.materiaQueImparte= "Calculo";
profesor1.horarioDeClase= 1300;
profesor1.divisionALaQuePertenece= "Ciencias básicas";
profesor1.universidadEnLaQueTrabaja= "UNAM";
profesor1.imprimirProfesor();

Profesor profesor2 = new Profesor (materiaQueImparte:"Algebra", horarioDeClase:1200, divisionALaQuePertenece:"Ciencias básicas", universidadEnLaQueTrabaja:"UNAM");
profesor2.imprimirProfesor();

```

[Imagen 8. Profesor con valores en el main.]

Profesor
materiaQueImparte: String horarioDeClase: double divisionALaQuePertenece: String UniversidadDondeTrabaja: String
pasarLista():void calificarTareas():void hacerExámenes():void

[Tabla 3. Profesor.]

Después, para trabajo en casa, quedaron pendientes realizar las clases TrianguloRectangulo y Perro, yo inicié con TrianguloRectangulo, siguiendo los mismos pasos que en las anteriores.

```

public class TrianguloRectangulo {
    float tamaniolados;
    float angulos;
    float tamano;
    String color;

    public TrianguloRectangulo(){

    }

    public TrianguloRectangulo(float tamaniolados, float angulos, float tamano, String color){
        this.tamaniolados=tamaniolados;
        this.angulos=angulos;
        this.tamano=tamano;
        this.color=color;
    }
}

```

[Imagen 9. TrianguloRectangulo atributos.]

```

public void teoremaDePitagoras(){
    System.out.println("Calculando...");
}
public void direccionDeFuerzas(){
    System.out.println("Calculando direccion de fuerzas...");
}
public void funcionesTrigonometricas(){
    System.out.println("Funciones Trigonometricas...");
}

public void imprimirTriangulo(){
    System.out.println("Tamaño de lados: " + tamaniolados);
    System.out.println("Angulo: " + angulos);
    System.out.println("Tamaño: " + tamano);
    System.out.println("Color: " + color);
}

```

[Imagen 10. TrianguloRectangulo métodos..]

```

TrianguloRectangulo triangulo1 = new TrianguloRectangulo();
triangulo1.imprimirTriangulo();
triangulo1.tamaniolados= 6;
triangulo1.angulos= 60;
triangulo1.tamano= 36;
triangulo1.color= "azul";
triangulo1.imprimirTriangulo();

TrianguloRectangulo triangulo2 = new TrianguloRectangulo (tamaniolados:9, angulos:45, tamano:36, color:"rojo");
triangulo2.imprimirTriangulo();

```

[Imagen 11. TrianguloRectangulo con valores en el main.]

TrianguloRectangulo
tamaniolados: float angulos: float tamaño: float color: String
teoremaDePitagoras():void direccionDeFuerzas():void funcionesTrigonometricas():void

[Tabla 4. TrianguloRectangulo.]

Por último, trabajé la clase Perro, siguiendo los mismo pasos que en los pasados, teniendo como resultado lo siguiente:

```

public class Perro {
    String nombre;
    String color;
    String raza;
    int numPatas;
    int numOrejas;

    public Perro(){

    }

    public Perro(String nombre, String color, String raza, int numPatas, int numOrejas){
        this.nombre=nombre;
        this.color=color;
        this.raza=raza;
        this.numPatas=numPatas;
        this.numOrejas=numOrejas;
    }
}

```

[Imagen 12. Perro atributos.]

```
public void jugar(){
    System.out.println(x:"Jugando");
}
public void ladrar(){
    System.out.println(x:"Woof Woof");
}
public void comer(){
    System.out.println(x:"Yum Yum");
}
public void dormir(){
    System.out.println(x:"Zzz");
}
public void correr(){
    System.out.println(x:"Corriendo");
}

public void imprimirPerro(){
    System.out.println("Nombre: " + nombre);
    System.out.println("Color: " + color);
    System.out.println("Raza: " + raza);
    System.out.println("Numero de patas: " + numPatas);
    System.out.println("Numero de orejas: " + numOrejas);
}
```

[Imagen 13. Perro métodos.]

```
Perro perro1 = new Perro();
perro1.imprimirPerro();
perro1.nombre= "Odugus";
perro1.color= "Beige";
perro1.raza= "Maltes";
perro1.numPatas= 4;
perro1.numOrejas = 2;
perro1.imprimirPerro();

Perro perro2 = new Perro (nombre:"Doggy", color:"Blanco", raza:"Poodle", numPatas:4, numOrejas:2);
perro2.imprimirPerro();
```

[Imagen 14. Perro con valores en el main.]

Perro
nombre: String color: String raza: String numPatas: int numOrejas: int
jugar():void ladrar():void comer():void dormir():void correr():void

[Tabla 5. Perro.]

- Clases y objetos: representación de entidades con atributos y métodos.
- Constructores: inicialización flexible de los objetos.
- Métodos: impresión de atributos y simulación de comportamientos.
- Instanciación múltiple: creación de varios objetos de la misma clase con características diferentes.

El uso de Apache NetBeans con Ant facilitó la gestión del proyecto, la compilación automática y la ejecución de pruebas. Con ello, se consolidó el entendimiento práctico de cómo diseñar e implementar programas orientados a objetos.

REFERENCIAS

- Thoth, & Thoth. (2023, 4 diciembre). *¿Qué son las APIs de Java?* Formatalent Business School.  
<https://formatalent.com/que-son-las-apis-de-java/>
- Amor, R. V. (2020, 6 marzo). *Introducción a Colecciones en Java*. Adictos Al Trabajo.  
<https://adictosaltrabajo.com/2015/09/25/introduccion-a-colecciones-en-java/>
- Navas, A. (2023, 4 diciembre). *Clases Wrapper (envoltorio) en Java*. Profile Software Services.  
<https://profile.es/blog/clases-wrapper-envoltorio-en-java/>

CONCLUSIONES

Durante esta práctica se afianzaron los principios básicos de POO en Java: