



ESCUELA POLITÉCNICA NACIONAL
ESCUELA DE FORMACIÓN DE TECNÓLOGOS



PROGRAMACION ORIENTADA A OBJETOS

ASIGNATURA:

Programación Orientada a Objetos

PROFESOR:

Ing. Yadira Franco

PERÍODO ACADÉMICO:

2023-B

TALLER N°1

TÍTULO:

HERENCIA JAVA

ESTUDIANTE

Yuverly Alexander Verdezoto Lojan
Lady Carolina Marin Flores

FECHA DE REALIZACIÓN: 09/ 12/ 2023

FECHA DE ENTREGA: 11 / 12 / 2023

CALIFICACIÓN OBTENIDA:

FIRMA DEL PROFESOR:

1 PROPÓSITO DE LA PRÁCTICA

Afianzar la comprensión de herencia en Java mediante la creación jerárquica de clases para modelar la diversidad animal.

2 OBJETIVOS ESPECÍFICOS

- Implementar una jerarquía de clases de animales con herencia de múltiples niveles en Java.
- Fomentar la aplicación de conceptos como polimorfismo y encapsulamiento en el diseño de la jerarquía.

3 DESARROLLO Y RESULTADOS DE LA PRÁCTICA

Nivel 1 (Clase Base - Animal):

Se creó una clase base llamada Animal con atributos nombre y edad.

Se implementaron métodos getNombre() y getEdad() para acceder a los atributos

```
1 // Creacion de la clase de Nivel 1
2 class Animal {
3     protected String nombre;
4     protected int edad;
5
6     public Animal(String nombre, int edad) {
7         this.nombre = nombre;
8         this.edad = edad;
9     }
10
11     public String getNombre() {
12         return nombre;
13     }
14
15     public int getEdad() {
16         return edad;
17     }
18 }
```

Nivel 2 (Clases Derivadas de Animal - Mamífero y Ave):

Se crearon las clases Mamifero y Ave que heredan de la clase Animal.

Se añadieron atributos adicionales (numeroPatas, tipoPelaje para Mamifero; tipoPlumaje, habilidadVuelo para Ave).

Métodos adicionales (getNumeroPatas(), getTipoPelaje() para Mamifero; getTipoPlumaje(), puedeVolar() para Ave) fueron implementados.

```
// creacion de la herencia al Nivel 2
4 usages 6 inheritors
class Mamifero extends Animal {
    2 usages
    protected int numeroPatas;
    2 usages
    protected String tipoPelaje;

    3 usages
    public Mamifero(String nombre, int edad, int numeroPatas, String tipoPelaje) {
        super(nombre, edad);
        this.numeroPatas = numeroPatas;
        this.tipoPelaje = tipoPelaje;
    }

    7 usages
    public int getNumeroPatas() {
        return numeroPatas;
    }

    7 usages
    public String getTipoPelaje() {
        return tipoPelaje;
    }
}

class Ave extends Animal {
    2 usages
    protected String tipoPlumaje;
    2 usages
    protected boolean habilidadVuelo;

    1 usage
    public Ave(String nombre, int edad, String tipoPlumaje, boolean habilidadVuelo) {
        super(nombre, edad);
        this.tipoPlumaje = tipoPlumaje;
        this.habilidadVuelo = habilidadVuelo;
    }

    1 usage
    public String getTipoPlumaje() {
        return tipoPlumaje;
    }

    1 usage
    public boolean puedeVolar() {
        return habilidadVuelo;
    }
}
```

Nivel 3 (Clases Derivadas de Mamífero - Felino y Canido):

Las clases Felino y Canido heredaron de la clase Mamifero.

Se agregaron atributos específicos (longitudCola, tipoCaza para Felino; raza, habilidadOlfato para Canido).

Se implementaron métodos adicionales (getLongitudCola(), getTipoCaza() para Felino; getRaza(), tieneBuenOlfato() para Canido).

```
// creacion del Nivel 3
4 usages 2 inheritors
class Felino extends Mamifero {
    2 usages
    protected int longitudCola;
    2 usages
    protected String tipoCaza;

    3 usages
    public Felino(String nombre, int edad, int numeroPatas, String tipoPelaje, int longitudCola, String tipoCaza) {
        super(nombre, edad, numeroPatas, tipoPelaje);
        this.longitudCola = longitudCola;
        this.tipoCaza = tipoCaza;
    }

    3 usages
    public int getLongitudCola() {
        return longitudCola;
    }

    3 usages
    public String getTipoCaza() {
        return tipoCaza;
    }
}

class Canido extends Mamifero {
    2 usages
    protected String raza;
    2 usages
    protected boolean habilidadOlfato;

    2 usages
    public Canido(String nombre, int edad, int numeroPatas, String tipoPelaje, String raza, boolean habilidadOlfato) {
        super(nombre, edad, numeroPatas, tipoPelaje);
        this.raza = raza;
        this.habilidadOlfato = habilidadOlfato;
    }

    3 usages
    public String getRaza() {
        return raza;
    }

    3 usages
    public boolean tieneBuenOlfato() {
        return habilidadOlfato;
    }
}
```

Nivel 4 (Clases Derivadas de Felino y Canido - Leon y Perro):

Se crearon las clases Leon y Perro que heredan de Felino y Canido respectivamente.

Se añadieron atributos específicos (melena, territorioCaza para Leon; colorPelaje, tamano para Perro).

Métodos adicionales (getMelena(), getTerritorioCaza() para Leon; getColorPelaje(), getTamano() para Perro) fueron implementados.

```
// creacion del Nivel 4
2 usages
class Leon extends Felino {
    2 usages
    protected String melena;
    2 usages
    protected String territorioCaza;

    1 usage
    public Leon(String nombre, int edad, int numeroPatas, String tipoPelaje, int longitudCola, String tipoCaza,
        String melena, String territorioCaza) {
        super(nombre, edad, numeroPatas, tipoPelaje, longitudCola, tipoCaza);
        this.melena = melena;
        this.territorioCaza = territorioCaza;
    }

    1 usage
    public String getMelena() {
        return melena;
    }

    1 usage
    public String getTerritorioCaza() {
        return territorioCaza;
    }
}
```

```
}
}

3 usages 1 inheritor
class Perro extends Canido {
    2 usages
    protected String colorPelaje;
    2 usages
    protected String tamano;

    2 usages
    public Perro(String nombre, int edad, int numeroPatas, String tipoPelaje, String raza, boolean habilidadOlfato,
        String colorPelaje, String tamano) {
        super(nombre, edad, numeroPatas, tipoPelaje, raza, habilidadOlfato);
        this.colorPelaje = colorPelaje;
        this.tamano = tamano;
    }

    2 usages
    public String getColorPelaje() {
        return colorPelaje;
    }
}
```

```
2 usages
public String getTamano() {
    return tamano;
}
}
```

Nivel 5 (Clases Específicas - GatoDomestico y LabradorRetriever):

Se desarrollaron las clases finales GatoDomestico y LabradorRetriever que heredan de Felino y Perro respectivamente.

Se agregaron atributos específicos (comportamientoDomestico, raza para GatoDomestico; habilidadNadar, personalidad para LabradorRetriever).

Métodos adicionales (getComportamientoDomestico() para GatoDomestico; puedeNadar(), getPersonalidad() para LabradorRetriever) fueron implementados.

```
// creacion del Nivel 5
2 usages
class GatoDomestico extends Felino {
    2 usages
    protected String comportamientoDomestico;
    2 usages
    protected String raza;

    1 usage
    public GatoDomestico(String nombre, int edad, int numeroPatas, String tipoPelaje, int longitudCola, String tipoCaza,
        String comportamientoDomestico, String raza) {
        super(nombre, edad, numeroPatas, tipoPelaje, longitudCola, tipoCaza);
        this.comportamientoDomestico = comportamientoDomestico;
        this.raza = raza;
    }

    1 usage
    public String getComportamientoDomestico() {
        return comportamientoDomestico;
    }
}

class LabradorRetriever extends Perro {
    2 usages
    protected boolean habilidadNadar;
    2 usages
    protected String personalidad;

    1 usage
    public LabradorRetriever(String nombre, int edad, int numeroPatas, String tipoPelaje, String raza, boolean habilidadOlfato,
        String colorPelaje, String tamano, boolean habilidadNadar, String personalidad) {
        super(nombre, edad, numeroPatas, tipoPelaje, raza, habilidadOlfato, colorPelaje, tamano);
        this.habilidadNadar = habilidadNadar;
        this.personalidad = personalidad;
    }

    1 usage
    public boolean puedeNadar() {
        return habilidadNadar;
    }

    1 usage
    public String getPersonalidad() {
        return personalidad;
    }
}
```

Se demostró la jerarquía de clases a través de instancias, mostrando la capacidad de cada nivel para heredar y extender atributos y métodos.

Se resolvieron errores, como el acceso directo a atributos específicos en lugar de utilizar métodos get en ciertos casos.

El programa principal (Main) instancia objetos de cada nivel y muestra información relevante en la consola para demostrar la correcta implementación de las herencias.

```
public class Main {
    public static void main(String[] args) {
        // Nivel 1
        Animal animal = new Animal( nombre: "AnimalGenerico", edad: 5);
        System.out.println("Animal - Nombre: " + animal.getNombre() + ", Edad: " + animal.getEdad());

        // Nivel 2
        Mamifero mamifero = new Mamifero( nombre: "Tigre", edad: 8, numeroPatas: 4, tipoPelaje: "Rayado");
        System.out.println("Mamifero - Nombre: " + mamifero.getNombre() + ", Edad: " + mamifero.getEdad() +
            ", Patas: " + mamifero.getNumeroPatas() + ", Pelaje: " + mamifero.getTipoPelaje());

        Ave ave = new Ave( nombre: "Águila", edad: 3, tipoPlumaje: "Dorado", habilidadVuelo: true);
        System.out.println("Ave - Nombre: " + ave.getNombre() + ", Edad: " + ave.getEdad() +
            ", Plumaje: " + ave.getTipoPlumaje() + ", Puede Volar: " + ave.puedeVolar());

        // Nivel 3
        Felino felino = new Felino( nombre: "Leopardo", edad: 6, numeroPatas: 4, tipoPelaje: "Manchado", longitudCola: 1, tipoCaza: "Carnivoro");
        System.out.println("Felino - Nombre: " + felino.getNombre() + ", Edad: " + felino.getEdad() +
            ", Patas: " + felino.getNumeroPatas() + ", Pelaje: " + felino.getTipoPelaje() +
            ", Cola: " + felino.getLongitudCola() + ", Tipo de Caza: " + felino.getTipoCaza());

        Canido canido = new Canido( nombre: "Lobo", edad: 7, numeroPatas: 4, tipoPelaje: "Gris", raza: "Gris", habilidadOlfato: true);
        System.out.println("Canido - Nombre: " + canido.getNombre() + ", Edad: " + canido.getEdad() +
            ", Patas: " + canido.getNumeroPatas() + ", Pelaje: " + canido.getTipoPelaje() +
            ", Raza: " + canido.getRaza() + ", Buen Olfato: " + canido.tieneBuenOlfato());

        // Nivel 4
        Leon leon = new Leon( nombre: "Simba", edad: 10, numeroPatas: 4, tipoPelaje: "Dorado", longitudCola: 1, tipoCaza: "Carnivoro", melena: "Melena Exuberante");
        System.out.println("León - Nombre: " + leon.getNombre() + ", Edad: " + leon.getEdad() +
            ", Patas: " + leon.getNumeroPatas() + ", Pelaje: " + leon.getTipoPelaje() +
            ", Cola: " + leon.getLongitudCola() + ", Tipo de Caza: " + leon.getTipoCaza() +
            ", Melena: " + leon.getMelena() + ", Territorio de Caza: " + leon.getTerritorioCaza());

        Perro perro = new Perro( nombre: "Buddy", edad: 4, numeroPatas: 4, tipoPelaje: "Negro", raza: "Labrador", habilidadOlfato: true, colorPelaje: "Café", tamaño:
        System.out.println("Perro - Nombre: " + perro.getNombre() + ", Edad: " + perro.getEdad() +
            ", Patas: " + perro.getNumeroPatas() + ", Pelaje: " + perro.getTipoPelaje() +
            ", Raza: " + perro.getRaza() + ", Buen Olfato: " + perro.tieneBuenOlfato() +
            ", Color de Pelaje: " + perro.getColorPelaje() + ", Tamaño: " + perro.getTamaño());

        // Nivel 5
        GatoDomestico gatoDomestico = new GatoDomestico( nombre: "Whiskers", edad: 2, numeroPatas: 4, tipoPelaje: "Atigrado", longitudCola: 0, tipoCaza: "Caza co
            comportamientoDomestico: "Doméstico", raza: "Siamés");
        System.out.println("Gato Doméstico - Nombre: " + gatoDomestico.getNombre() + ", Edad: " + gatoDomestico.getEdad() +
            ", Patas: " + gatoDomestico.getNumeroPatas() + ", Pelaje: " + gatoDomestico.getTipoPelaje() +
            ", Cola: " + gatoDomestico.getLongitudCola() + ", Tipo de Caza: " + gatoDomestico.getTipoCaza() +
            ", Comportamiento Doméstico: " + gatoDomestico.getComportamientoDomestico() +
            ", Raza: " + gatoDomestico.raza);

        LabradorRetriever labrador = new LabradorRetriever( nombre: "Max", edad: 3, numeroPatas: 4, tipoPelaje: "Dorado", raza: "Labrador", habilidadOlfato: true,
            habilidadNadar: true, personalidad: "Amigable");
        System.out.println("Labrador Retriever - Nombre: " + labrador.getNombre() + ", Edad: " + labrador.getEdad() +
            ", Patas: " + labrador.getNumeroPatas() + ", Pelaje: " + labrador.getTipoPelaje() +
            ", Raza: " + labrador.getRaza() + ", Buen Olfato: " + labrador.tieneBuenOlfato() +
            ", Color de Pelaje: " + labrador.getColorPelaje() + ", Tamaño: " + labrador.getTamaño() +
            ", Puede Nadar: " + labrador.puedeNadar() + ", Personalidad: " + labrador.getPersonalidad());
    }
}
```

Salida en la terminal

```
Animal - Nombre: AnimalGenerico, Edad: 5
Mamifero - Nombre: Tigre, Edad: 8, Patas: 4, Pelaje: Rayado
Ave - Nombre: Águila, Edad: 3, Plumaje: Dorado, Puede Volar: true
Felino - Nombre: Leopardo, Edad: 6, Patas: 4, Pelaje: Manchado, Cola: 1, Tipo de Caza: Carnivoro
Canido - Nombre: Lobo, Edad: 7, Patas: 4, Pelaje: Gris, Raza: Gris, Buen Olfato: true
León - Nombre: Simba, Edad: 10, Patas: 4, Pelaje: Dorado, Cola: 1, Tipo de Caza: Carnivoro, Melena: Melena Exuberante, Territorio de Caza: Sabana
Perro - Nombre: Buddy, Edad: 4, Patas: 4, Pelaje: Negro, Raza: Labrador, Buen Olfato: true, Color de Pelaje: Café, Tamaño: Mediano
Gato Doméstico - Nombre: Whiskers, Edad: 2, Patas: 4, Pelaje: Atigrado, Cola: 0, Tipo de Caza: Caza con Juguetes, Comportamiento Doméstico: Doméstico, Ra
Labrador Retriever - Nombre: Max, Edad: 3, Patas: 4, Pelaje: Dorado, Raza: Labrador, Buen Olfato: true, Color de Pelaje: Amarillo, Tamaño: Grande, Puede
```


4 CONCLUSIONES

La práctica de herencia en Java demostró ser esencial para organizar jerarquías complejas de clases. La estructura de cinco niveles facilita la representación de relaciones específicas entre animales, promoviendo la reutilización de código y la extensibilidad del sistema. Además, la implementación de métodos `get` garantiza el encapsulamiento y acceso controlado a los atributos. Este enfoque jerárquico brinda claridad en el diseño del sistema, permitiendo una fácil comprensión de las relaciones entre las clases. El modularidad resultante simplifica la adaptación y expansión futuras del sistema.

La resolución de errores, como el acceso directo a atributos, destaca la importancia de seguir las prácticas de encapsulamiento. Al utilizar métodos para acceder a los atributos, se promueve una estructura más segura y mantenible. Además, la práctica evidencia la versatilidad de la herencia al representar entidades diversas con características específicas en una jerarquía coherente. Este ejercicio fortalece la comprensión de los principios de programación orientada a objetos y su aplicación en la creación de sistemas extensibles y bien organizados.

Link del REPOSITORIO: <https://github.com/YuverlyHidokun/Deber-Herencias/tree/master>

5 BIBLIOGRAFÍA

Equipo Geek - N. (2019, 5 de junio). *¿Qué es la Herencia en programación orientada a objetos?* IfgeekthenNTTdata. <https://ifgeekthen.nttdata.com/es/herencia-en-programacion-orientada-objetos>