

**Universidad Autónoma de Nuevo León.  
Facultad de Ciencias Físico Matemáticas.**

## **Práctica #04 – Herencia.**

Laboratorio de Programación Orientada a Objetos.

**Alumno:** Jeremy Uriel Rossell Segura.

**Matrícula:** 2173396.

**Grupo:** 036.

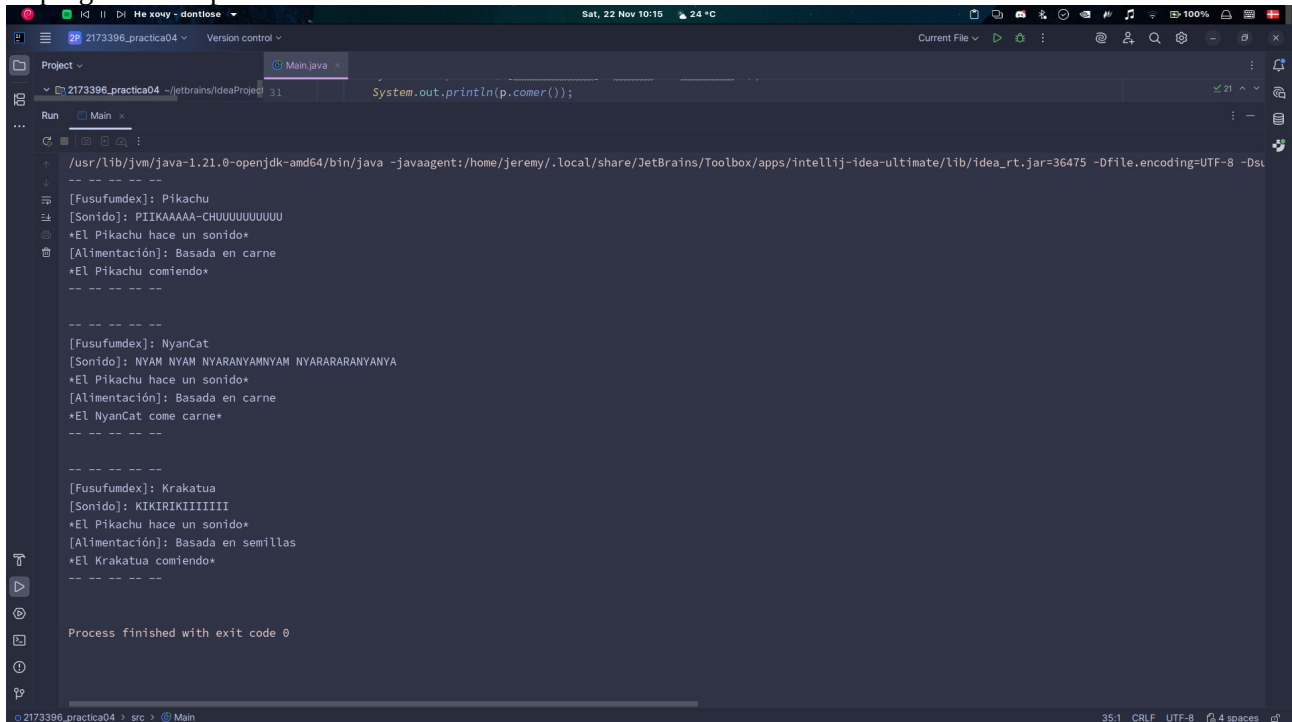
**Horario:** VIERNES 7:00-9:00.

**Materia:** Laboratorio de Programación Orientada a objetos.

**Docente:** Jorge Alberto Islas Pineda.

# Compilación.

El programa compiló adecuadamente.



```
Project 2173396_practica04 ~jetbrains/ideaProject 31
Main.java
System.out.println(p.comer());

Run Main
/usr/lib/jvm/java-1.21.0-openjdk-amd64/bin/java -javaagent:/home/jeremy/.local/share/JetBrains/Toolbox/apps/intellij-idea-ultimate/lib/idea_rt.jar=36475 -Dfile.encoding=UTF-8 -D...

[Fusufumdex]: Pikachu
[Sonido]: PIIKAAAAA-CHUUUUUUUUUU
*El Pikachu hace un sonido*
[Alimentación]: Basada en carne
*El Pikachu comiendo*

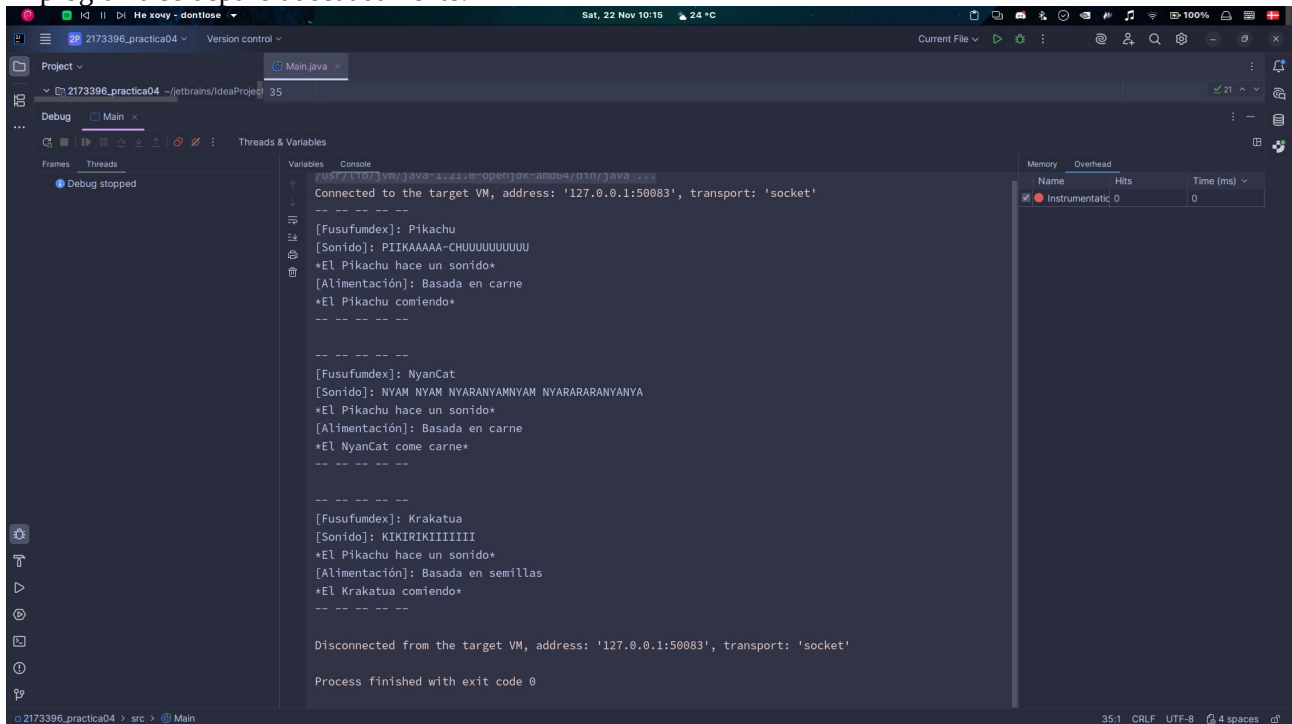
[Fusufumdex]: NyanCat
[Sonido]: NYAM NYAM NYARANYAMNYAM NYARARARANYANYA
*El Pikachu hace un sonido*
[Alimentación]: Basada en carne
*El NyanCat come carne*

[Fusufumdex]: Krakatua
[Sonido]: KIKIRIKIIIIIIII
*El Pikachu hace un sonido*
[Alimentación]: Basada en semillas
*El Krakatua comiendo*

Process finished with exit code 0
```

# Depuración.

El programa se depuró adecuadamente.



```
Project 2173396_practica04 ~jetbrains/ideaProject 35
Main.java
System.out.println(p.comer());

Debug Main
Threads & Variables
Frames Threads
Debug stopped

Variables Console
Connected to the target VM, address: '127.0.0.1:50083', transport: 'socket'

[Fusufumdex]: Pikachu
[Sonido]: PIIKAAAAA-CHUUUUUUUUUU
*El Pikachu hace un sonido*
[Alimentación]: Basada en carne
*El Pikachu comiendo*

[Fusufumdex]: NyanCat
[Sonido]: NYAM NYAM NYARANYAMNYAM NYARARARANYANYA
*El Pikachu hace un sonido*
[Alimentación]: Basada en carne
*El NyanCat come carne*

[Fusufumdex]: Krakatua
[Sonido]: KIKIRIKIIIIIIII
*El Pikachu hace un sonido*
[Alimentación]: Basada en semillas
*El Krakatua comiendo*

Disconnected from the target VM, address: '127.0.0.1:50083', transport: 'socket'

Process finished with exit code 0
```

# Código.

El código del programa.

```
1 // extend es para heredar clase
2 import animales.*;
3
4 public class Main {
5     public static void main(String[] args) {
6         Animal a = new Animal( nombre: "Pikachu");
7         Gato g = new Gato( nombre: "NyanCat");
8         Pajaro p = new Pajaro( nombre: "Karakatua");
9
10        System.out.println("-- -- -- --");
11        System.out.println("[Fusufundex]: " + a.mostrarNombre());
12        System.out.println("[Sonido]: PIKAAAAA-CHUUUUUUUUUU");
13        a.hacerSonido();
14        System.out.println("[Alimentación]: Basada en carne");
15        System.out.println(a.comer());
16        System.out.println("-- -- -- --\n");
17
18        System.out.println("-- -- -- --");
19        System.out.println("[Fusufundex]: " + g.mostrarNombre());
20        System.out.println("[Sonido]: NYAM NYAM NYARANYAMNYAM NYARARARANYANYA");
21        a.hacerSonido();
22        System.out.println("[Alimentación]: Basada en carne");
23        System.out.println(g.comer());
24        System.out.println("-- -- -- --\n");
25
26        System.out.println("-- -- -- --");
27        System.out.println("[Fusufundex]: " + p.mostrarNombre());
28        System.out.println("[Sonido]: KIKIRIKIIIIIIII");
29        a.hacerSonido();
30        System.out.println("[Alimentación]: Basada en semillas ");
31        System.out.println(p.comer());
32        System.out.println("-- -- -- --\n");
33    }
34 }
35
package animales;
1
2
3 public class Animal { 2 inheritors
4     String nombre;
5
6     public Animal(String nombre) { this.nombre = nombre; }
7
8     public String mostrarNombre() { return nombre; }
9
10    public void hacerSonido() { System.out.println("El " + nombre + " hace un sonido"); }
11
12    public String comer() { return "El " + nombre + " comiendo"; }
13
14 }
15
16
package animales;
1
2
3 public interface Descarga {
4     public void descargar();
5 }
6
package animales;
1
2
3 public class Pajaro extends Animal implements Descarga { 2 usages
4     public Pajaro(String nombre) { super(nombre); }
5
6     @Override 3 usages
7     public void hacerSonido() { System.out.println("El " + nombre + " hace un sonido"); }
8
9     @Override no usages
10    public void descargar() { System.out.println("El " + nombre + " descarga"); }
11
12 }
13
14
package animales;
1
2
3 public class Gato extends Animal { 2 usages
4     public Gato(String nombre) { super(nombre); }
5
6     public void hacerSonido() { System.out.println("El " + nombre + " hace un sonido"); }
7
8     public String comer() { return "El " + nombre + " come carne"; }
9
10 }
```

# Archivos.

Los archivos del programa.

