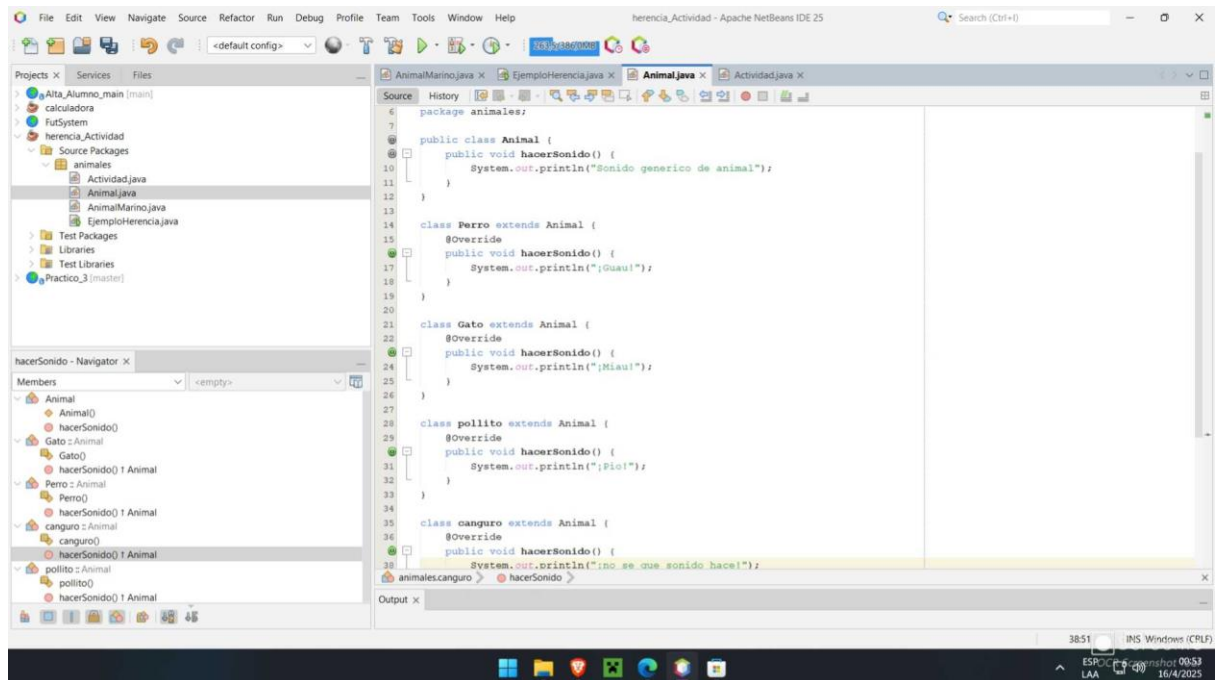


1. Ejecucion y modificacion de su ejemplo Describe la modificacion que realizaste: la modificacion es de agregar 2 animales nuevos dandole sus sonidos



Codigo:

```
package animales;
```

```
public class Animal { public void hacerSonido() { System.out.println("Sonido generico de animal"); }}
```

```
class Perro extends Animal { @Override public void hacerSonido() { System.out.println("¡Guau!"); }}
```

```
class Gato extends Animal { @Override public void hacerSonido() { System.out.println("¡Miau!"); }}
```

```
class pollito extends Animal { @Override public void hacerSonido() { System.out.println("¡Pio!"); }}
```

```
class canguro extends Animal { @Override public void hacerSonido() { System.out.println("¡no se!"); }}
```

```
Animal miAnimal1 = new Perro(); Animal miAnimal2 = new Gato(); Animal miAnimal3 = new pollito(); Animal miAnimal4 = new canguro();
```

```
miAnimal1.hacerSonido();  
miAnimal2.hacerSonido();  
miAnimal3.hacerSonido();
```

```
        miAnimal4.hacerSonido();  
  
    }  
}
```

2.Crea mi propio ejemplo Jerarquía de clases:

AnimalMarino es la superclase o clase base.

Delfin y Tiburon son subclases que heredan de AnimalMarino.

Atributos: nombre (de tipo String): almacena el nombre del animal.

habitat (de tipo String): almacena el lugar donde vive el animal.

Métodos: AnimalMarino define un método: nadar(): imprime un mensaje genérico sobre nadar en el océano.

Delfin: Sobrescribe nadar(): muestra cómo nada y salta fuera del agua.
hacerAcrobacia(): muestra que el delfín está haciendo acrobacias.

vive(): imprime en qué hábitat vive el delfín.

Tiburon: Sobrescribe nadar(): muestra cómo nada en busca de presas.

cazar(): muestra que el tiburón ha cazado su alimento.

vive(): imprime en qué hábitat vive el tiburón.

Codigo:

```
package animales;  
  
public class AnimalMarino { String nombre; String habitat;  
  
    public void nadar() {  
        System.out.println("Estoy nadando en el océano.");  
    }  
  
}
```

```

class Delfin extends AnimalMarino { @Override public void nadar() {
    System.out.println(nombre + " nada rapidamente y salta fuera del agua.");}

    public void vive() {
        System.out.println(nombre + " vive en " + habitat);
    }

    public void hacerAcrobacia() {
        System.out.println(nombre + " esta haciendo acrobacias!");
    }

}

class Tiburon extends AnimalMarino { @Override public void nadar() {
    System.out.println(nombre + " nada sigilosamente en busca de presas.");}

    public void vive() {
        System.out.println(nombre + " vive en " + habitat);
    }

    public void cazar() {
        System.out.println(nombre + " ha cazado su alimento.");
    }

}

package animales;

public class EjemploHerencia {

    public static void main(String[] args) {
        System.out.println("-----Clase AnimalMarino-----");
        Delfin delfin = new Delfin();
        delfin.nombre = "Flipper";
        delfin.habitat = " el acuario ";
        delfin.nadar();
        delfin.hacerAcrobacia();
        delfin.vive();

        System.out.println("");
    }
}

```

```
Tiburon tiburon = new Tiburon();
tiburon.nombre = "Bruce";
tiburon.habitat = " el oceano ";
tiburon.nadar();
tiburon.vive();
tiburon.cazar();

}}
```

1¿Cuál es la principal ventaja de utilizar la herencia en la POO?

La herencia permite reutilizar código, evitando duplicación y facilitando su uso. Además, mejora la organización al estructurar jerarquías claras y lógicas.

2¿En qué situaciones consideras que la herencia múltiple podría ser útil?

La herencia múltiple podría ser útil en situaciones donde una clase necesita comportamientos que provienen de varias fuentes diferentes.

3¿Cómo se relaciona el polimorfismo con la herencia?

El polimorfismo permite que objetos de diferentes subclases sean tratados como objetos de la superclase.

por ejemplo:

tenemos una superclase Animal y subclases como Perro, Gato y pollito, todas sobrescriben el método hacerSonido().

Aunque la variable es de tipo Animal, cada objeto ejecuta su propia versión del método.

4¿Qué dificultades encontraste al aplicar los conceptos de herencia?

De momenton no eh tenido dificultad al tener un ejemplo. solo es guiarme con el y si sale algun error comparo con su ejemplo y trato de ver que puede ser