



## Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

# Laboratorios de computación salas A y B

<i>Profesor:</i>	Edgar Tista Garcia
<i>Asignatura:</i>	Programación Orientada a Objetos.
<i>Grupo:</i>	Grupo 3
<i>No de Práctica(s):</i>	Práctica 7 – 8. Herencia y Polimorfismo.
<i>Integrante(s):</i>	Edwin Jaret Santiago Díaz
<i>No. de Equipo de cómputo empleado:</i>	Trabajo en casa.
<i>No. de Lista o Brigada:</i>	38
<i>Semestre:</i>	2022 - 2
<i>Fecha de entrega:</i>	3 abril 2022
<i>Observaciones:</i>	

CALIFICACIÓN: \_\_\_\_\_

# Herencia y Polimorfismo.

## Objetivos

### Objetivo General:

(7) Implementar los conceptos de herencia en un lenguaje de programación orientado a Objetos.

(8) Implementar el concepto de polimorfismo en un lenguaje de programación orientado a objetos.

**Objetivos de clase:** Realizar aplicaciones sencillas donde el estudiante pueda visualizar de manera clara los conceptos relacionados con herencia como clases abstractas e interfaces, así como la importancia del polimorfismo y la sobrescritura de métodos.

## Desarrollo

### Ejemplos de la guía

### Ejercicios de la clase

#### Ejercicio 1. Jerarquía de Herencia.

Se desarrolló un programa en donde se puede implementar en una tienda de tecnología. El programa contiene listas de productos disponibles para los diferentes tipos de usuarios en los cuales se muestra el tipo de dispositivo, la marca y el precio del producto.

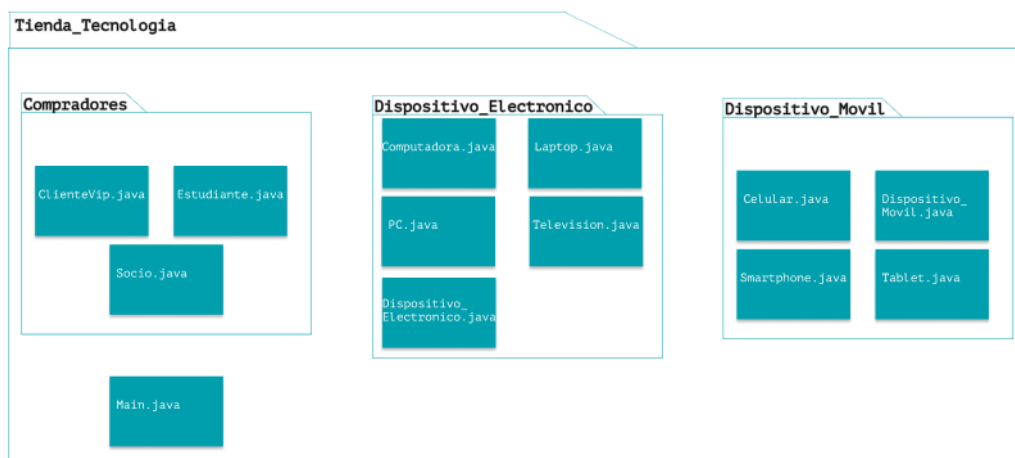
En el programa se implementó el uso de paquetes y para compilar el programa se utiliza el siguiente comando:

```
javac Tienda_Tecnologia/main.java
```

Una vez compilado el programa se ejecuta utilizando el siguiente comando.

```
java Tienda_Tecnologia.main
```

La administración de paquetes se muestra en la siguiente imagen.



En la imagen se muestra que el paquete **Tienda\_Tecnologia** contiene la clase main (programa principal en donde se ejecutará todo), los paquetes **Compradores** (contiene los tipos de compradores), **Dispositivo\_Electronico** (contiene los tipos de dispositivos electrónicos) y **Dispositivo\_Movil** (contiene los tipos de dispositivos móviles).

Al ejecutar la clase main se mostrará un menú al usuario en donde se le pregunta qué tipo de comprador va a realizar una compra (cada comprador tiene disponible distintas listas de productos tecnológicos), hay 3 opciones y estas son **Socio**, **Estudiante** y **Cliente VIP** como se muestra en la siguiente imagen.

Si se ingresa que el comprador es un socio, el programa imprime la lista de productos que tiene disponible el socio.

```
Practica 7 - 8

Bienvenido a la tienda de tecnologia!

Compradores:
1. Socio
2. Estudiante
3. ClienteVip
Que tipo de comprador eres?
█
```

```
Dispositivos moviles para el Socio:

1 Smartphone: Samsung S20 1000.0
2 Tablet: iPad Pro 3000.0
3 Celular: Huawei 1000.0
4 Celular: Apple 1200.0
5 Tablet: iPad 8 2000.0
6 Smartphone: iPhone 11 1200.0
7 Smartphone: Huawei P20 1000.0
8 Celular: Samsung 1000.0

Ingresa 77 para terminar de comprar

Que desea comprar?
█
```

Una vez impreso la lista de productos, se solicita el número del producto que se desea comprar y luego solicitará la cantidad de este.

Se realizará un ejemplo del funcionamiento del programa al ingresar que el socio quiere comprar 2 smartphones Samsung S20 (opción 1), 1 celular Samsung (opción 8) y 1 iPad 8 (opción 5), el resultado sería el siguiente.

<pre>Que desea comprar? 1 Cuanto desea comprar? 2  Dispositivo agregado al carrito Que desea comprar? 8 Cuanto desea comprar? 1  Dispositivo agregado al carrito Que desea comprar? 5 Cuanto desea comprar? 1  Dispositivo agregado al carrito</pre>	<div>2 smartphones Samsung S20</div> <div>1 celular Samsung</div> <div>1 iPad 8</div>
--	---

Una vez agregado los productos y el comprador no desea agregar más productos, se ingresa la opción 77 para que se muestre la cantidad de productos que va a comprar el Socio y el total a pagar, como se muestra en la siguiente imagen.

Si se ingresa que el comprador es un estudiante, el programa imprime la lista de productos que tiene disponible el estudiante.

```
Que desea comprar?  
77  
  
No existe ese producto, intente de nuevo  
  
Total de productos: 4  
Total a pagar: 5000
```

```
Dispositivos moviles para el Estudiante:  
  
1 Smartphone: Apple 1200.0  
2 Smartphone: Xiaomi 600.0  
  
Dispositivos electronicos para el Estudiante:  
  
3 Laptop: Acer 1500.0  
4 Computadora: Asus 1900.0  
5 Computadora: MacBook 1600.0  
6 Laptop: Asus 2000.0  
7 Computadora: Hp 1200.0  
8 Laptop: Lenovo 1800.0  
  
Ingrese 77 para terminar de comprar  
  
Que desea comprar?  
□
```

Una vez impreso la lista de productos, se solicita el número del producto que se desea comprar y luego solicitará la cantidad de este.

Se realizará un ejemplo del funcionamiento del programa al ingresar que el estudiante quiere comprar 4 smartphones Xiaomi (opción 2) y 1 Laptop Asus (opción 6).

```
Que desea comprar?  
2  
Cuanto desea comprar?  
4  
  
Dispositivo agregado al carrito  
Que desea comprar?  
6  
Cuanto desea comprar?  
1  
  
Dispositivo agregado al carrito
```

4 smartphones Xiaomi

1 Laptop Asus

Una vez agregado los productos y el comprador no desea agregar más productos, se ingresa la opción 77 para que se muestre la cantidad de productos que va a comprar el Estudiante y el total a pagar, como se muestra en la siguiente imagen.

```
Que desea comprar?  
77  
  
No existe ese producto, intente de nuevo  
  
Total de productos: 5  
Total a pagar: 4400
```

Si se ingresa que el comprador es un Cliente VIP, el programa imprime la lista de productos que tiene disponible el cliente vip.

```
Dispositivos electronicos disponibles para Cliente VIP:

1 PC: Hp 1200.0
2 Television: Sony 1000.0
3 Laptop: Asus 2000.0
4 Laptop: Lenovo 1800.0
5 Computadora: Asus 1900.0
6 PC: MacBook 1600.0
7 Computadora: MacBook 1600.0
8 Television: Samsung 1200.0
9 Computadora: Hp 1200.0
10 Laptop: Acer 1500.0
11 PC: Asus 1900.0
12 Television: LG 1000.0

Elija 77 si desea salir del programa

Que desea comprar?
█
```

Una vez impreso la lista de productos, se solicita el número del producto que se desea comprar y luego solicitará la cantidad de este.

Se realizará un ejemplo del funcionamiento del programa al ingresar que el Cliente VIP quiere comprar 1 computadora MacBook (opción 7), 3 PC MacBook (opción 6), 2 Televisiones LG (opción 12) y 4 Laptops Acer (opción 10).

```
Que desea comprar?
7
Cuanto desea comprar?
1
Dispositivo agregado al carrito
```

1 computadora MacBook

```
Que desea comprar?
6
Cuanto desea comprar?
3
Dispositivo agregado al carrito
```

3 PC MacBook

```
Que desea comprar?
12
Cuanto desea comprar?
2
Dispositivo agregado al carrito
```

2 Televisiones LG

```
Que desea comprar?
10
Cuanto desea comprar?
4
Dispositivo agregado al carrito
```

4 Laptops Acer

Una vez agregado los productos y el comprador no desea agregar más productos, se ingresa la opción 77 para que se muestre la cantidad de productos que va a comprar el Cliente VIP y el total a pagar, como se muestra en la siguiente imagen.

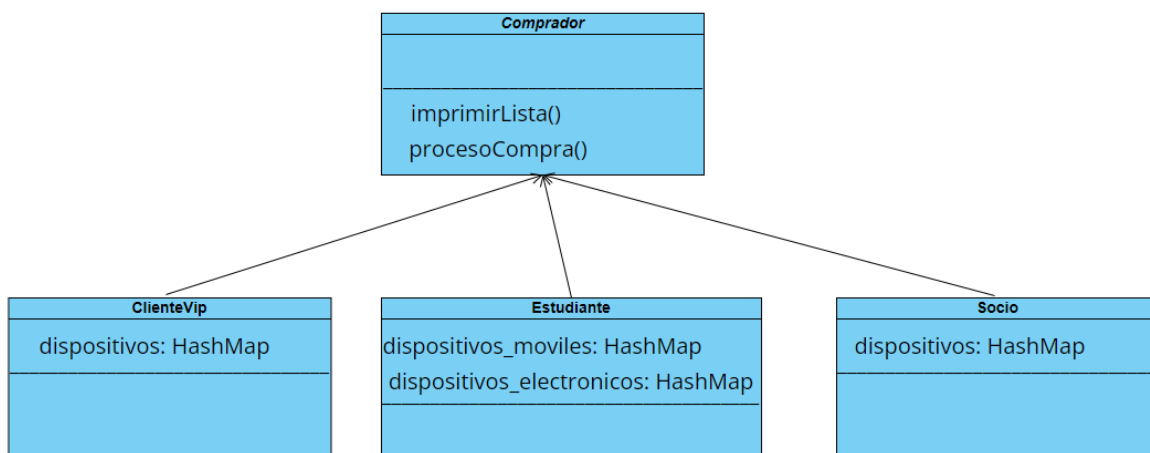
```
Que desea comprar?
77

No existe ese producto, intente de nuevo

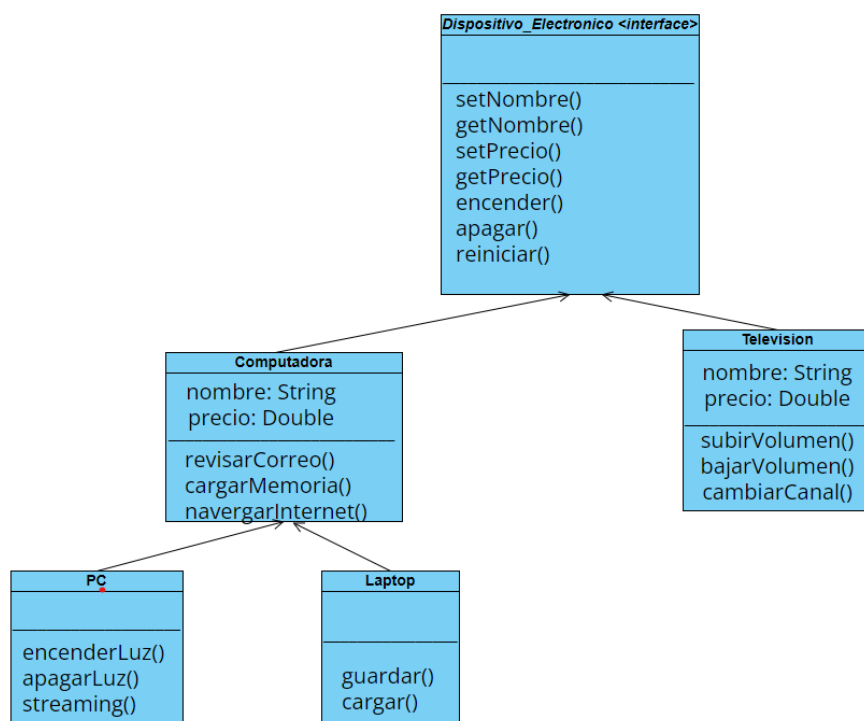
Total de productos: 10
Total a pagar: 14400
```

Después de que se imprime el total a pagar, el programa termina de ejecutarse, por lo tanto, cada vez que un comprador desea comprar se tiene que ejecutar el programa.

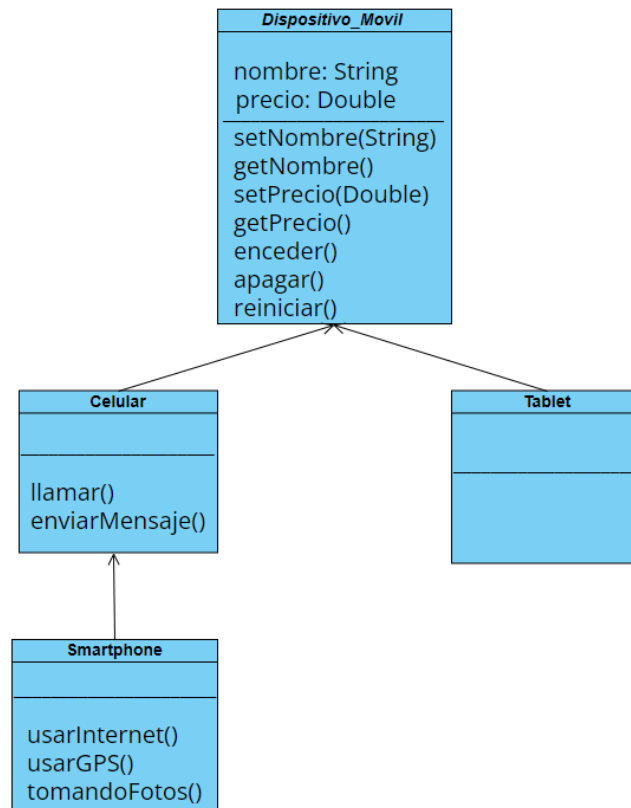
Diagrama UML de los programas es el siguiente.



La clase **Comprador** es una clase abstracta en donde se declaran los métodos *imprimirLista()* y *procesoComprar()*. De la clase abstracta se hereda a las clases **ClienteVIP**, **Estudiante** y **Socio**. En estas clases se utilizan los métodos heredados y cada clase contiene una colección de tipo HashMap en donde se guardan los dispositivos disponibles para cada comprador.



Se utiliza una interface con el nombre **Dispositivo\_Electronico** en donde contiene 7 métodos. La interface se utiliza para la clase **Televisión** y para la clase abstracta **Computadora**. De esta ultima clase se va a heredar a las clases **PC** y **Laptop** las cuales , aparte de los métodos heredados contienen sus propios métodos.



La clase **Dispositivo\_Movil** es una clase abstracta en donde se declaran los métodos 7 métodos. De la clase abstracta se hereda a las clases **Celular** y **Tablet**. En estas clases se utilizan los métodos heredados, para la clase **Celular** se utilizan 2 métodos propios y para la clase **Smartphone** se utilizan, además de los métodos heredados, 3 métodos propios.

## Ejercicio 2. Colección de objetos.

Se desarrolló un programa en donde se establece una jerarquía de herencia. El programa contiene 9 distintos animales en donde cada animal actúa de diferente manera al utilizar el mismo método que tienen estos 9 animales.

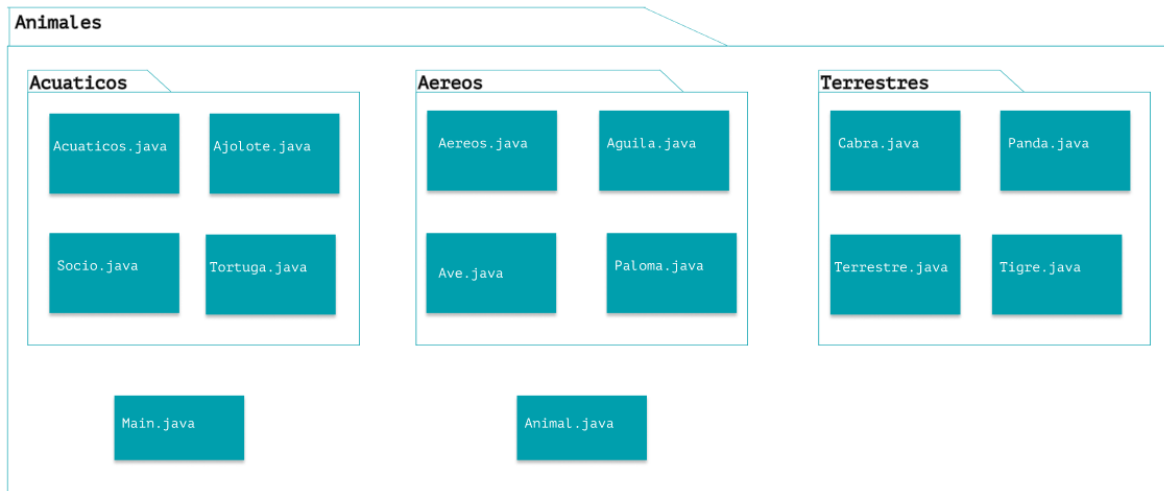
En el programa se implementó el uso de paquetes y para compilar el programa se utiliza el siguiente comando:

```
javac Animales/main.java
```

Una vez compilado el programa se ejecuta utilizando el siguiente comando.

```
java Animales.main
```

La administración de paquetes se muestra en la siguiente imagen.



En la imagen se muestra que el paquete **Animales** contiene la clase main (programa principal en donde se ejecutará todo), los paquetes **Acuaticos** (contiene 3 clases de animales acuaticos), **Aereos** (contiene 3 clases de animales aéreos) y **Terrestres** (contiene 3 clases de animales terrestres.).

Al ejecutar la clase main se mostrará un menú al usuario con 3 opciones en donde se le pregunta cual opción desea realizar, estas opciones son para agregar un animal, ver un animal y ver el catálogo de animales.

Al elegir la opción 1 (agregar un animal) se le solicita al usuario cuál de los 9 animales disponibles desea agregar (guardar en una lista)

```

=====
Catalogo de animales
=====

Que desea hacer?
1. Agregar animal
2. Ver un animal
3. Ver catalogo
4. Salir


```

```

Que animal desea agregar?

1. Cabra
2. Panda
3. Tigre

4. Paloma
5. Aguila
6. Ave

7. Tiburon
8. Ajolote
9. Tortuga
Solo se puede escoger el animal 1 vez

```

Para este ejemplo se va a agregar 1 panda, 1 ajolote, 1 águila y 1 tortuga en este orden.

```

2
Panda agregada

```

Al agregar un animal, el programa vuelve a solicitarle al usuario que se desea hacer (agregar un animal, ver un animal o ver el catálogo) y por lo tanto se debe de reingresar la opción 1 hasta que los animales (que se desea agregar) hayan sido agregados.

```

8
Ajolote agregado

```

```

5
Aguila agregada

```

```

9
Tortuga agregada

```



Una vez agregado los animales, se puede escoger la opción 2 en donde se le solicitará al usuario qué animal desea ver. Al escoger un animal, el programa ejecutará los métodos de este animal (los cuales son sus comportamientos) y estos resultados serán distintos para cada animal.

```
2
Que animal desea ver?

1. Cabra
2. Panda
3. Tigre

4. Paloma
5. Aguila
6. Ave

7. Tiburon
8. Ajolote
9. Tortuga
```

Antes de elegir qué animal desea ver, se debe considerar que solo se puede escoger los animales registrados previamente (en la opción 1), estos son el panda, ajolote, águila y la tortuga. Para este ejemplo, los animales que se desea ver son los mismos que fueron registrados previamente.

```
Panda
Comer carne, frutas, verduras, hojas
Dormir en el bosque unas 8 horas
Morir por la edad
Soy viviparo y tardo 3 a 5 meses
Comunicarse por el sonido
Desplazarse por el terreno moviendo sus pies
Respira aire
Habitat: Tierra
```

```
Aguila
Comer insectos, roedores y aves
Dormir 12 horas mientras estoy en tierra
Morir por la edad
Soy oviparo y eclosiono en 35 dias
Comunicarse por el sonido
Desplazarse por el aire
Respirar aire
Habitat: Arboles
```

```
Tortuga
Comer medusas y peces
Dormir estatico por 5 horas
Morir por un popote en la nariz
Soy oviparo y nazco en la arena, al eclosionar me tengo que mover rapido al oceano
Comunicarse por el sonido
Desplazarse por el agua
Respirar por el agua
Habitat: Agua
```

```
Ajolote
Comer larvas de insecto, caracoles y lombrices
Dormir estatico
Morir por el cambio climatico
Nacer en el agua después de 15 dias en un huevo
Comunicarse por el sonido
Desplazarse por el agua
Respirar por el agua
Habitat: Agua
```

Se comprobará los animales registrados por el usuario.

Si fuera el caso de que se quiere ver un animal que no fue registrado anteriormente, se le notifica al usuario, por ejemplo, ver al animal **Paloma** (opción 4).

```
4
Este animal no se ha agregado, intente agregarlo!
```

Si se desea visualizar el catálogo de animales (opción 3) el programa imprimirá el nombre de los animales que han sido registrados por el usuario previamente (en la opción 1)

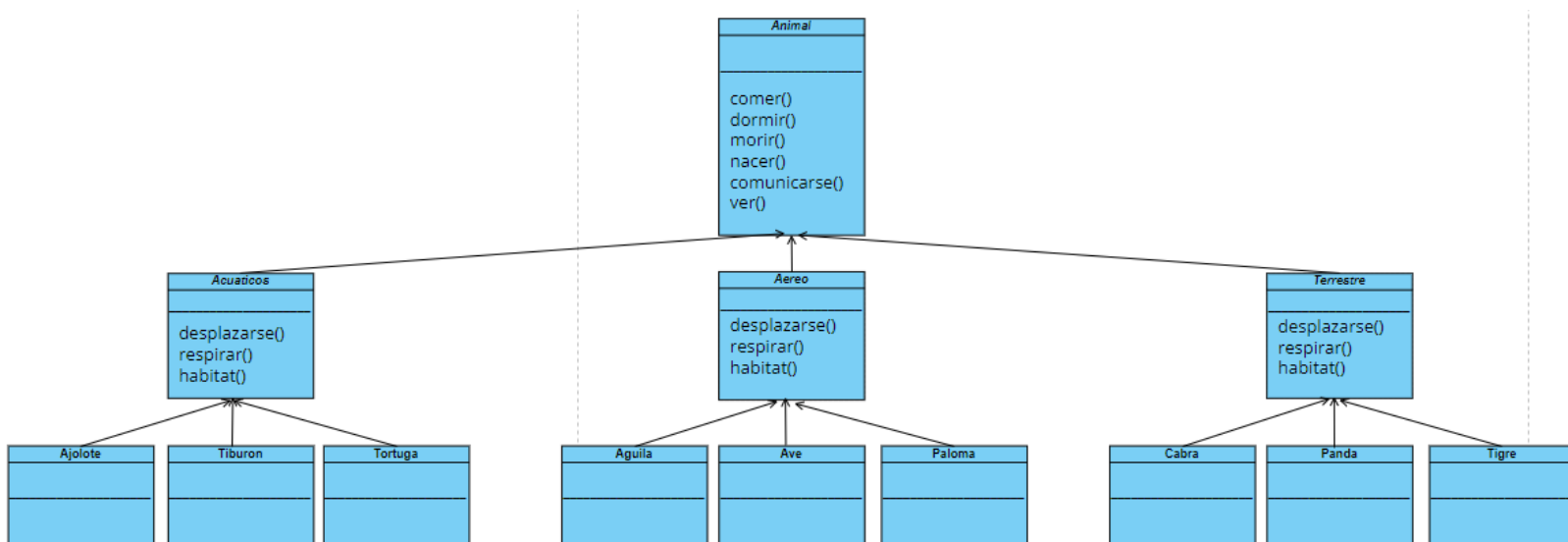
```
3
Mostrar todos los animales guardados

Panda
Ajolote
Aguila
Tortuga
```

Si se desea finalizar la ejecución del programa, solo es necesario ingresar la opción 4 y el programa se finaliza.

```
4
Saliendo del programa
```

El diagrama UML de los programas es el siguiente.



La clase abstracta **Animal** contiene 6 métodos que se ocuparan en todas las demás clases. Esta clase hereda a las clases abstractas **Acuáticos**, **Aéreo** y **Terrestre**. Cada una de estas clases contiene 3 métodos que contienen el mismo nombre pero realizan funciones distintas.

De la clase abstracta **Acuáticos** hereda a las clases **Ajolote**, **Tiburón** y **Tortuga**. Estas 3 clases utilizan los 6 métodos de la clase **Animal** pero realizan funciones distintas.

De la clase abstracta **Aéreo** hereda a las clases **Aguila**, **Ave** y **Paloma**. Estas 3 clases utilizan los 6 métodos de la clase **Animal** pero realizan funciones distintas.

De la clase abstracta **Terrestre** hereda a las clases **Cabra**, **Panda** y **Tigre**. Estas 3 clases utilizan los 6 métodos de la clase **Animal** pero realizan funciones distintas.

## Conclusiones

Se completó la practica en su totalidad resolviendo los ejercicios propuestos por el profesor, así mismo se cumplen los objetivos de la práctica. Los programas están documentados.

El uso de las clases abstractas e interfaces facilita el flujo del trabajo de los programas pues se obtiene un orden, la construcción de las clases es más limpio y se obtiene una mayor comprensión de los programas.

Lo que más se me complicó del programa es que en el ejercicio 1, al guardar los dispositivos de la tienda de tecnología en un HashMap tuve muchos errores de sintaxis para acceder a través de un número (índice) el elemento de esa posición.

En el ejercicio 2 encontré la manera de obtener el nombre de la clase y esto se me hizo de mucha utilidad ya que en el caso del ejercicio, los animales guardados por el usuario se guardaron en un ArrayList y al iterarlo, se obtuvo el nombre de la clase de cada elemento y así obtener la información de que animales estaban registrados.