

# **ACADEMÍA JAVA XIDERAL MONTERREY, N.L**

## **SEMANA 1 - EJERCICIO 1: POLIMORFISMO**

**Realizado por:  
Luis Miguel Sánchez Flores**

## PROBLEMÁTICA:

Los superhéroes se han convertido en modelos a seguir para millones de personas alrededor del mundo desde su primera aparición en los años 30, retratando la premisa universal del bien contra el mal con el uso de poderes sobrehumanos para combatir el crimen y salvar a la población. Desde entonces, los superhéroes se convirtieron en un fenómeno cultural a través de cómics, libros, series de televisión y películas.

**A los fanáticos de los superhéroes les gusta llevar a cabo un registro de sus héroes favoritos a lo largo de los años**, descubriendo y accediendo información acerca de cada uno de los personajes.

Con tantos superhéroes que existen hoy en día, cada uno de ellos con sus propios nombres, poderes, habilidades, etc., **puede resultar un proceso tedioso y complejo tener en cuenta a cada uno de los superhéroes**, y el usuario es más susceptible de cometer errores al momento de registrarlos.

Sin embargo, **uno puede observar que los superhéroes comparten el mismo tipo de características**: Cada uno de ellos tiene un nombre, un poder único, una historia de origen, debilidades, entre otras cosas.

**Si hubiera una forma de generar y aprovechar una plantilla de superhéroe que pudiera agilizar el proceso de registro y clasificación de una manera más eficaz...**



## SOLUCIÓN:

En el mundo de Java, este tipo de plantillas se pueden implementar gracias al **polimorfismo**, un potente concepto de la programación orientada a objetos que permite que múltiples clases (superhéroes específicos como Flash, Batman, Hulk, etc.) compartan una interfaz o clase padre común (la plantilla Superhéroe), lo que beneficia enormemente el proceso de registro al añadir nuevos superhéroes con mayor facilidad y mejorar la mantenibilidad del código.

El polimorfismo puede adoptar diferentes formas. En este caso, se logra el polimorfismo mediante el uso de la **abstracción**, que consiste en ocultar detalles innecesarios y permite crear una plantilla clara y cohesionada para los superhéroes y simplificar la complejidad del proyecto.

Se empieza con la **clase abstracta** de **Superheroe**, en el cual se incluyen los atributos y funciones que los superhéroes suelen compartir, tales como nombre, lugar de origen y equipo / alianza a la que pertenecen, habilidades, su historia de origen y su mayor debilidad:

```
1 package polimorfismo;
2
3 public abstract class Superheroe {
4
5     // Se definen variables comunes entre los superhéroes:
6     String nombre;
7     String lugarDeOrigen;
8     String equipo;
9
10    // CONSTRUCTOR
11    Superheroe(String nombre, String lugarDeOrigen, String equipo ) {
12        this.nombre = nombre;
13        this.lugarDeOrigen = lugarDeOrigen;
14        this.equipo = equipo;
15    }
16
17    // Imprimir nombre del superhéroe:
18    @Override
19    public String toString() {
20        return "SUPERHEROE : " + nombre;
21    }
22
23    // Imprimir el lugar de origen del superhéroe:
24    void getLugarDeOrigen()
25    {
26        System.out.println("LUGAR DE ORIGEN : " + nombre + " nació en " + lugarDeOrigen + ".");
27    }
28
29    // Imprimir la alianza, facción o equipo que pertenece el superhéroe, si aplica:
30    void getEquipo()
```

```
29    // Imprimir la alianza, facción o equipo que pertenece el superhéroe, si aplica:
30    void getEquipo()
31    {
32        if(equipo != null && !equipo.isEmpty())
33            System.out.println(nombre + " pertenece al equipo de " + equipo);
34        else
35            System.out.println(nombre + " trabaja solo....");
36    }
37
38    // MÉTODOS ABSTRACTOS:
39    // Se declaran métodos / funciones comunes entre los superhéroes:
40    abstract void usarPoder();
41
42    abstract void historiaDeOrigen();
43
44    abstract void getDebilidad();
45
46 }
```

La clase abstracta es después heredada por las subclases **Batman**, **Spiderman** y **Superman**, donde todos ellos contienen las siguientes líneas de código:

- Se incluye un constructor en el que se definen los atributos del superhéroe mediante una llamada al constructor de la clase superior (Superhero) con "super()":
- Con la herencia de la clase abstracta, cada uno de los superhéroes deben implementar los métodos solicitados como se muestra a continuación:

```
1 package polimorfismo;
2 //IS-A
3 public class Superman extends Superhero{
4
5
6     public Superman() {
7         super("Superman", "Krypton", "La Liga de la Justicia");
8     }
9
10    /* Con la clase abstracta Superhero heredada, se tienen que implementar
11       cada uno de los métodos solicitados */
12
13    // Así, cada Superhero implementa los poderes que posee, su historia de origen y su mayor debilidad:
14
15    @Override
16    void usarPoder() {
17        System.out.println("¡" + nombre + " tiene superfuerza para recoger cualquier objeto que te imaginas!");
18    }
19
20    @Override
21    void historiaDeOrigen() {
22        System.out.println("Kal-El nació originalmente en la ciudad de Kryptonopolis, "
23            + "en el planeta Krypton. Su padre, el mejor científico del planeta, "
24            + "hizo muchos estudios desde antes que Kal-El naciera, "
25            + "que probaba que el planeta estaba alcanzando altas temperaturas en su núcleo...");
26    }
27
28
29    @Override
30    void getDebilidad() {
```

Con las subclases desarrolladas, se puede poner a prueba el poder del polimorfismo. En la clase **Principal** que contiene la función **main** a ejecutar, se genera un arreglo de tipo Superhero, incluyendo todas las subclases desarrolladas:

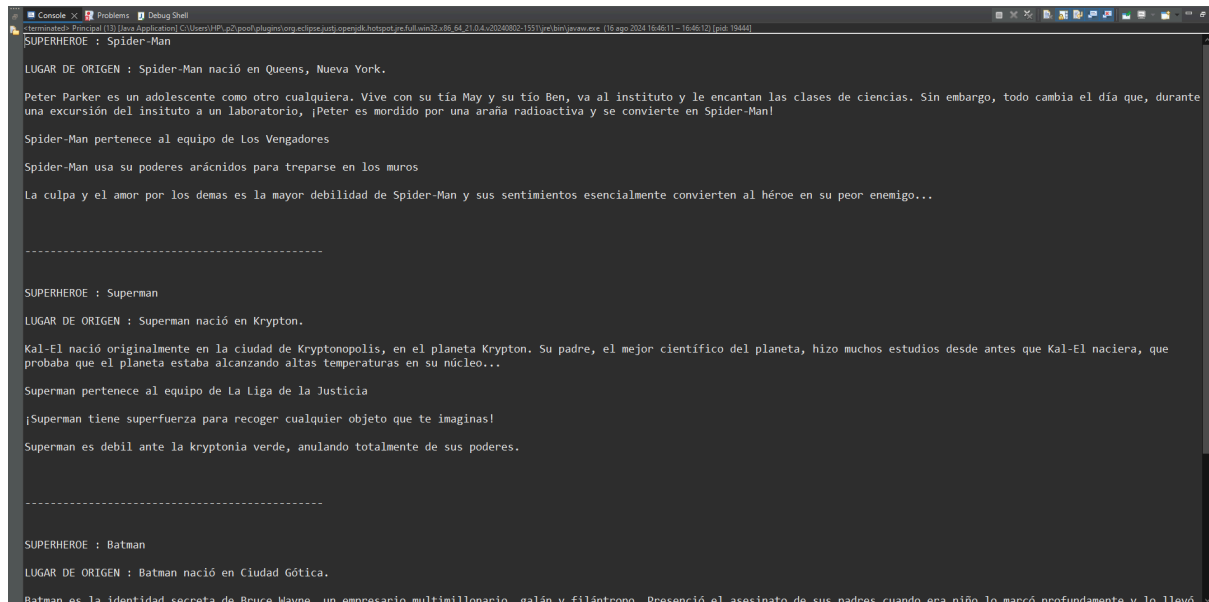
```
10 public class Principal {
11
12     public static void main(String[] args) {
13
14         // Generamos una lista de los superheroes disponibles:
15         Superhero[] superheroes = {
16             new Spiderman(),
17             new Superman(),
18             new Batman(),
19         };
20
```

Se finaliza la función main con un ciclo for que pasa por la lista de superhéroes, imprimiendo la información sobre cada uno de ellos. Dado que cada subclase hereda los atributos y métodos de la clase abstracta, podemos reducir la complejidad del código especificando la variable como tipo Superhero y ejecutando dichos métodos, lo que ayuda a que el código sea más limpio y fácil de mantener:

```
20
21     // Para cada uno de los superheroes incluidos en el listado:
22     for(Superhero superhero : superheroes) {
23
24         // Se imprime información acerca del superheroe correspondiente, incluyendo:
25
26         // Nombre:
27         System.out.println(superhero);
28         System.out.println();
29
30         // Lugar de origen:
31         superhero.getLugarDeOrigen();
32         System.out.println();
33
34         // Su historia de origen:
35         superhero.historiaDeOrigen();
36         System.out.println();
37
38         // El equipo o alianza a la que pertenece:
39         superhero.getEquipo();
40         System.out.println();
41
42         // Los poderes que posee:
43         superhero.usarPoder();
44         System.out.println();
45
46         // Su mayor debilidad:
47         superhero.getDebilidad();
48         System.out.println();

```

El resultado es la descripción de cada uno de los superhéroes incluidos en la lista. Java es capaz de determinar en tiempo de ejecución la implementación apropiada de los métodos, haciendo que el código tenga un diseño flexible y extensible, y permite la creación de otros superhéroes sin afectar al código existente.



```
Console x Problems Debug Shell
C:\Users\HP\p2\psd\ejugm\org.eclipse.jdt.launcher.hotspot-jre-fallback-12-x86_64-21.64-v20240802-1551\rel\bin\javaw.exe (18 ago 2024 16:46:11 - 16:46:12) [pid: 19446]

SUPERHEROE : Spider-Man

LUGAR DE ORIGEN : Spider-Man nació en Queens, Nueva York.

Peter Parker es un adolescente como otro cualquiera. Vive con su tía May y su tío Ben, va al instituto y le encantan las clases de ciencias. Sin embargo, todo cambia el día que, durante una excursión del insituto a un laboratorio, ¡Peter es mordido por una araña radioactiva y se convierte en Spider-Man!

Spider-Man pertenece al equipo de Los Vengadores

Spider-Man usa su poderes arácnidos para treparse en los muros

La culpa y el amor por los demas es la mayor debilidad de Spider-Man y sus sentimientos esencialmente convierten al héroe en su peor enemigo...

-----

SUPERHEROE : Superman

LUGAR DE ORIGEN : Superman nació en Krypton.

Kal-El nació originalmente en la ciudad de Kryptonopolis, en el planeta Krypton. Su padre, el mejor científico del planeta, hizo muchos estudios desde antes que Kal-El naciera, que probaba que el planeta estaba alcanzando altas temperaturas en su núcleo...

Superman pertenece al equipo de La Liga de la Justicia

¡Superman tiene superfuerza para recoger cualquier objeto que te imaginas!

Superman es debil ante la kryptonita verde, anulando totalmente de sus poderes.

-----

SUPERHEROE : Batman

LUGAR DE ORIGEN : Batman nació en Ciudad Gótica.

Batman es la identidad secreta de Bruce Wayne, un empresario multimillonario, galán y filántropo. Presenció el asesinato de sus padres cuando era niño lo marcó profundamente y lo llevó
```

Mediante el uso del polimorfismo, se permite la creación de aplicaciones robustas y escalables como es el registro de superhéroes.