



Recibe una cálida:

¡Bienvenida!

Te estábamos esperando 😊 

➤ Polimorfismo y principios básicos de diseño

Plan formativo: Desarrollo de Aplicaciones Full Stack Java Trainee V2.0

HOJA DE RUTA

¿Cuáles **skill** conforman el programa?



REPASO CLASE ANTERIOR

En la clase anterior trabajamos :

✓ Interfaces



LEARNING PATHWAY

4.6

Start! 🏁

Polimorfismo y
principios básicos de
diseño

Polimorfismo mediante
Interfaces

Polimorfismo mediante
Interfaces

Veterinaria 4.0

OBJETIVOS DE APRENDIZAJE

¿Qué aprenderemos?



**Comprender la implementación de
polimorfismo a través de interfaces en Java**





Rompehielo 🧊



Respondan en el chat o levantando la mano: 🙋

- ¿Qué profesionales ven en la imagen?
- ¿Qué función cumple cada uno?
- ¿Cómo interactúan con los pacientes?

Podríamos decir que, aunque de forma diferente, todos realizan funciones de atención médica.

- ¿De qué manera podríamos representar esto en programación?



› Polimorfismo mediante Interfaz



Polimorfismo mediante Interfaces



¿Cómo se implementa?:

Igual que con las clases, se puede conseguir el polimorfismo con las interfaces porque al implementar una interfaz podemos decir que **el objeto de la clase es un objeto de la interfaz**.



El polimorfismo entra en juego cuando utilizamos una interfaz como **tipo de referencia** en lugar de una clase concreta. Esto nos permite escribir código genérico que puede trabajar con diferentes clases que implementan la misma interfaz.





Polimorfismo mediante Interfaces



Veamos un ejemplo en código:

Las interfaces en Java nos permiten aplicar polimorfismo de una manera flexible.

La clave está en declarar referencias de la interfaz en lugar de la clase concreta.

Por ejemplo, podríamos tener una interfaz Figura con un método obtenerArea():



```
public interface Figura {  
    public double obtenerArea();  
}
```



Polimorfismo mediante Interfaces



Luego hacemos que diferentes clases implementen la interfaz Figura y por consiguiente el método a implementar.

En este caso serán las clases Circulo y Rectangulo.



```
public class Circulo implements Figura {  
    @Override  
    public double obtenerArea() {
```

```
public class Rectangulo implements Figura{  
    @Override  
    public double obtenerArea() {
```



Polimorfismo mediante Interfaces



En la clase Main, vamos a crear una referencia a la interfaz para asignarle objetos Circulo o Rectangulo según sea necesario.



```
public static void main(String[] args) {  
  
    Figura figura; //referencia a la interfaz  
  
    figura = new Circulo();  
  
    figura = new Rectangulo();  
}
```



Polimorfismo mediante Interfaces



Al llamar a **figura.obtenerArea()** se ejecutará el método correspondiente al tipo de objeto real (Círculo o Rectángulo).

Esto es **polimorfismo mediante interfaces**, nos permite tratar objetos de distintas clases de manera uniforme a través de la interfaz en común.

La ventaja es que podemos agregar nuevas clases que implementen Figura y nuestro código seguirá funcionando sin necesidad de modificación.

```
public static void main(String[] args) {  
  
    Figura figura; //referencia a la interfaz  
  
    figura = new Circulo();  
    //llamamos al método de la clase Circulo  
    figura.obtenerArea();  
  
    figura = new Rectangulo();  
    //llamamos al método de la clase Rectangulo  
    figura.obtenerArea();  
}
```

Evaluación Integradora ✨

¿Listos para un nuevo desafío? En esta clase comenzamos a construir nuestro...

Trabajo Integrador del Módulo 💪

Iremos completándolo progresivamente clase a clase.



LIVE CODING

Ejemplo en vivo

Formas de Pago:

Un buen ejemplo de interfaz podría ser la forma de pago en una billetera electrónica. Esta forma de pago se aplica a cualquier medio de pago que se elija utilizar. Veamos un ejemplo:

*1- Crear una interfaz **FormaPago** con método **procesarPago()** e implementarla en las clases **TarjetaCredito**, **PayPal** y **Efectivo**. Probar polimorfismo pagando de distintas formas.*

 **Tiempo: 25 minutos**





Ejercicio N° 1

Veterinaria 4.0



Veterinaria 4.0

Aplicando polimorfismo: 🙌

Como hemos visto, el polimorfismo se puede aplicar tanto mediante herencia como mediante interfaces. En este caso vamos a modificar el código de los ejercicios anteriores para practicar la implementación de Interfaces.

Consigna: ✍️

1. Crear una interfaz **Animal** con métodos hacerRuido(), comer() y moverse().
2. **Implementar** en clases Perro, Pez, Gato y Ave.
3. Crear un **arreglo** de tipo “Animal” en el main y utilizar **polimorfismo** para mostrar la implementación de cada método.

Tiempo 🕒: 30 minutos

○

¿Alguna consulta?

+



RESUMEN

¿Qué logramos en esta clase?

- ✓ **Comprender la implementación del polimorfismo mediante interfaces**
- ✓ **Reconocer los beneficios de aplicar polimorfismo mediante interfaces**



#WorkingTime

Continuemos ejercitando

¡Antes de cerrar la clase! Te invitamos a: 🙌 🙌 🙌

1. Repasar nuevamente la grabación de esta clase
2. Revisar el material compartido en la plataforma de Moodle (lo que se vio en clase y algún ejercicio adicional)
 - a. Material 1 (Foro)
 - b. *Lectura Módulo 4, Lección 6: páginas 6 - 7*
3. Traer al próximo encuentro, todas tus dudas y consultas para verlas antes de iniciar nuevo tema.

¡Muchas Gracias!

Nos vemos en la próxima clase 🙌



Momento: ✚

Time-out!

🕒 5 min.

