

Ejemplo de Patrón Strategy

DigitalHouse >
Coding School



**Certified Tech
Developer**
The Ultimate Degree

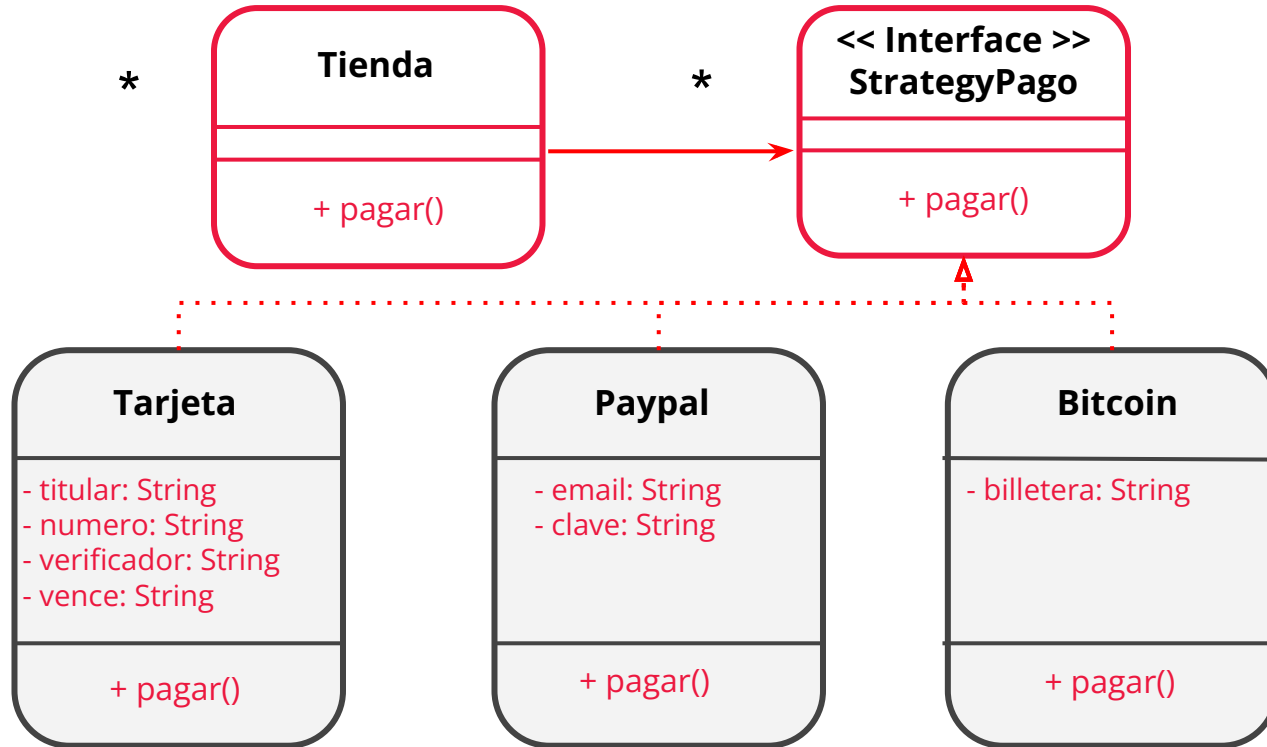
Ejemplo de Patrón Strategy

Supongamos que estamos diseñando una tienda *online* y queremos que acepte distintos medios de pago (tarjeta de crédito, Paypal, Bitcoin, etc). Un ejemplo de arquitectura con Strategy sería el siguiente.

Veamos el diagrama UML a seguir (no vamos a escribir los constructores de las clases).



Solución



1

Implementación de las clases del diagrama UML

```
public interface StrategyPago {  
    void pago();  
}
```

Código de la
interface
StrategyPago

Implementación de las clases del diagrama UML

Para los pagos vamos a definir clases para tarjeta, Paypal y Bitcoin:

```
public class Tarjeta implements StrategyPago {  
  
    private String titular;  
  
    private String numero;  
  
    private String verificador;  
  
    private String vence;  
  
    public void pago() {  
  
        System.out.println("pagado con tarjeta");  
  
    }  
}
```

Código de la
clase
Tarjeta

Implementación de las clases del diagrama UML

```
public class Paypal implements StrategyPago {  
    private String email;  
    private String clave;  
    public void pago(){  
        System.out.println("pagado con paypal");  
    }  
}
```

Código de la
clase
Paypal

Implementación de las clases del diagrama UML

```
public class Bitcoin implements StrategyPago {  
    private String billetera;  
    public void pago() {  
        System.out.println("pagado con bitcoin");  
    }  
}
```

Código de la
clase
Bitcoin

Implementación de las clases del diagrama UML

```
public class Tienda {  
    private StrategyPago formaPago;  
  
    public void pago() {  
        formaPago.pago();  
    }  
  
    public void setPago(StrategyPago nuevoPago) {  
        formaPago = nuevoPago;  
    }  
}
```

Código de la
clase
Tienda

Conclusión



Vemos que en este ejemplo simplificado podemos utilizar un objeto **Tienda**, setear la forma de pago (de algunas de las clases que tenemos que implementan **StrategyPago**) y efectuar el pago.

La lógica del pago estará, entonces, en cada forma de pago y no en la tienda, separando y facilitando el algoritmo.

Para agregar una nueva forma de pago debería solamente hacer una nueva clase que implemente **StrategyPago**.

DigitalHouse>
Coding School