



Patrones de Estrategia en Ingeniería de Software: Mejores Prácticas para el Éxito del Desarrollo

Introducción



Patrón de Estrategia

El **patrón de estrategia** es una técnica de programación que permite separar la implementación de un algoritmo de su uso. Esto permite cambiar el algoritmo en tiempo de ejecución sin modificar el código que lo utiliza. Este patrón también permite crear algoritmos intercambiables y reutilizables.



Ventajas de los Patrones de Estrategia



Ejemplos de Patrones de Estrategia

Existen varios **patrones de estrategia** utilizados en el desarrollo de software, como el patrón de **Estrategia Simple**, el patrón de **Estrategia Compuesta** y el patrón de **Estrategia de Estrategias**. Estos patrones se utilizan para resolver diferentes problemas y ofrecen soluciones flexibles y escalables.

```
public interface PaymentStrategy {  
    double calculateTotalAmount(double[] items);  
}
```

```
public class CreditCardPaymentStrategy implements PaymentStrategy {  
    public double calculateTotalAmount(double[] items) {  
        // Lógica para calcular el monto total usando una tarjeta de crédito  
        // ...  
    }  
}  
  
public class PayPalPaymentStrategy implements PaymentStrategy {  
    public double calculateTotalAmount(double[] items) {  
        // Lógica para calcular el monto total usando PayPal  
        // ...  
    }  
}  
  
public class CashPaymentStrategy implements PaymentStrategy {  
    public double calculateTotalAmount(double[] items) {  
        // Lógica para calcular el monto total en efectivo  
        // ...  
    }  
}
```

Para implementar **patrones de estrategia**, es importante tener en cuenta la estructura del software y definir una interfaz común para las diferentes estrategias. Además, es importante elegir la estrategia correcta para cada situación y asegurarse de que las estrategias sean intercambiables y reutilizables.

Implementación de Patrones de Estrategia

