



Exemplo Padrão Strategy



**Certified
Developer**
The Ultimate Tech Degree

DigitalHouse >
Coding School

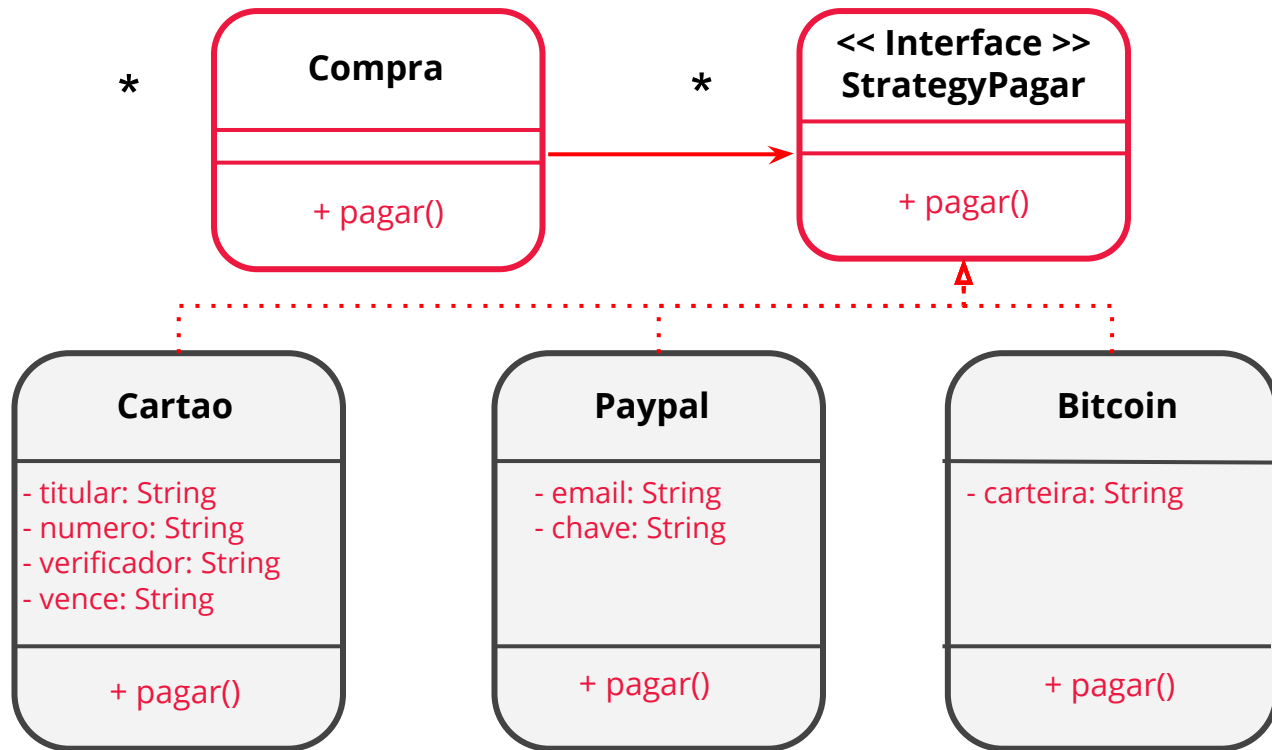
Exemplo de Padrão Strategy

Suponha que estejamos projetando uma loja online e queiramos aceitar diferentes meios de pagamento (cartão de crédito, Paypal, Bitcoin, etc). Um exemplo de arquitetura com Estratégia seria o seguinte.

Vamos ver o diagrama UML a seguir (não vamos escrever os construtores das classes).



Solução



1

Implementação das classes do diagrama UML

```
public interface StrategyPagar {  
    void pagar();  
}
```

Código da
interface
StrategyPagar

Implementação das classes do diagrama UML

Para os tipos de pagamento vamos definir as classes para cartão, Paypal e Bitcoin:

```
public class Cartao implements StrategyPago {  
  
    private String titular;  
  
    private String numero;  
  
    private String verificador;  
  
    private String vence;  
  
    public void pagar() {  
  
        System.out.println("pago com cartão");  
  
    }  
}
```

Código da
classe
Cartao

Implementação das classes do diagrama UML

```
public class Paypal implements StrategyPagar {  
    private String email;  
    private String chave;  
    public void pagar() {  
        System.out.println("pago com paypal");  
    }  
}
```

Código da
classe
Paypal

Implementação das classes do diagrama UML

```
public class Bitcoin implements StrategyPagar {  
    private String carteira;  
  
    public void pagar() {  
        System.out.println("pago com bitcoin");  
    }  
}
```

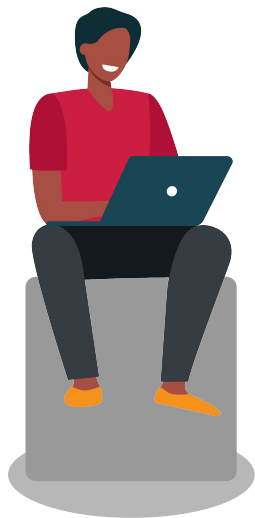
Código da
classe
Bitcoin

Implementação das classes do diagrama UML

```
public class Compra {  
    private StrategyPagar formaPagamento;  
  
    public void pagar() {  
        formaPagamento.pagar();  
    }  
  
    public void setPagar(StrategyPagar novoPagamento)  
    {  
        formaPagamento= novoPagamento;  
    }  
}
```

Código da
classe
Compra

Conclusão



Vemos que neste exemplo simplificado podemos usar um objeto Compra, definir a forma de pagamento (de algumas das classes que temos que implementam StrategyPagar) e fazer o pagamento.

A lógica do pagamento estará, então, em cada forma de pagamento e não na loja, separando e facilitando o algoritmo.

Para adicionar um novo método de pagamento você deve apenas fazer uma nova classe que implemente o StrategyPagar.

DigitalHouse>
Coding School