



Swift



Design Pattern ou Padrões de projeto



O que é Design Pattern?

O que é Design Pattern?

É uma solução geral para um problema que ocorre com frequência dentro de um determinado contexto no projeto de software.

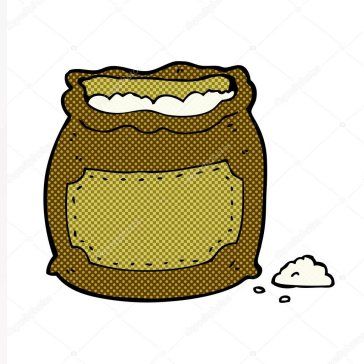
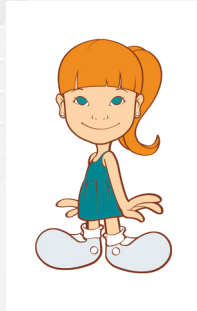
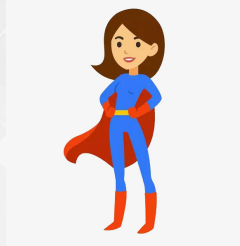
Por que usamos Design Patterns

O valor dos design patterns reside no fato que eles são soluções que foram utilizadas e testadas, o que nos dá confiança em sua eficácia.

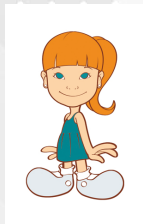
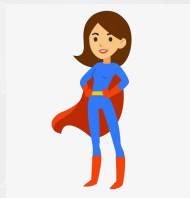
Alguns Patterns

- Abstract Factory
- Builder
- Factory Method
- Prototype
- Singleton
- Bridge
- Façade
- **Observer**
- Interpreter
- Etc

Observer



Observer



...



Estou aguardando minha filha voltar....(Observer)



Como podemos usar o observer no swift?

- Protocol (delegate, listener)
- Closures
- Notification Center

Vamos falar sobre delegate



```
protocol Voavel {  
  func voar()  
}
```




```
class Aeroporto {  
    private var delegate: Voavel  
    init(delegate: Voavel) {  
        self.delegate = delegate  
    }  
    func autorizarVoo() {  
        delegate.voar()  
    }  
}
```



```
class Aviao: Voavel {  
    func voar() {  
        print("Estou voando, que maravilha")  
    }  
}
```






```
var aeroporto = Aeroporto(delegate: Aviao())  
aeroporto.autorizarVoo()
```


Exercício do Jornal

Vamos falar de Closures



```
var closureName: (ParameterTypes) -> ReturnType
```



```
func comprarFarinha(tortaFinalizada: (_  
farinha: String) -> Void) {  
  
}
```

```
comprarFarinha { (farinha) in  
  
}
```

Exercício do Jornal com closure

Links



[Sobre Patterns](#)

[Patterns usados em swift](#)

[Sobre delegates](#)

[Fucking closures yntax](#)

