



Python^{Live de}

#121 - State

Obrigado!

apoia.se/livedepython

```
dunossauro at unknown in ~/git/apoiase on master*  
$ python apoiadores.py  
Alexandre Possebom   Alexandre Tsuno       Alysson Oliveira     Amaziles Carvalho  
And Past             Andre Machado         Andre Rodrigues      Athayr Athayr  
Bernardo Fontes     Bruno Oliveira        Bruno Gaffuri        Bruno Guizi  
Bruno Rocha         Carlos Augusto        Cleber Santos        Cleiton Mittmann  
DYLAN MAKER         David Reis            David Silva          Dayham Soares  
Diego Ubirajara     Edimar Fardim        Eliabe Silva         Eliakim Moraes  
Elias Soares       Eliel Lima           Emerson Lara         Eugenio Mazzini  
Fabiano Silos       Fabiano Teichmann    Fabiano Gomes       Fausto Caldeira  
Fernando Furtado    FlavKaze FlavKaze     Franklin Silva       Fábio Serrão  
Gleison Oliveira    Guilherme Ramos       Hemilio Lauro        Humberto Rocha  
Hélio Neto          JONATHAN DOMINGOS    Jean Vetorello       Johnny Tardin  
Jonatas Oliveira    Jonatas Simões       José Prado           João Lugão  
João Coelho         Juan Gutierrez        Jucélio Silva        Júlia Kastrup  
Kauan Alves         Leon Teixeira         Lucas Nascimento     Magno Malkut  
Marcello Benigno    Marcus Salgues       Maria Boladona       Matheus Francisco  
Nilo Pereira        Nídio Dolfini        Pablo Henrique       Patrick Corrêa  
Paulo Tadei         Pedro Alves          Rafael Galleani      Regis Santos  
Renan Gomes         Renan Moura          Renato Santos        Rennan Almeida  
Renne Rocha        Rhenan Bartels       Rodrigo Ferreira     Rodrigo Vaccari  
Sérgio Passos      Thais Viana          Thiago Araujo        Tiago Cordeiro  
Tyrone Damasceno    Vergil Valverde      Vicente Marcal       Vinícius Ferreira  
Wander Silva        Wellington Carlos     Wellington Camargo   Welton Souza  
William Oliveira    Willian Gl           Yros Aguiar          Falta você
```

Roteiro

- Estate

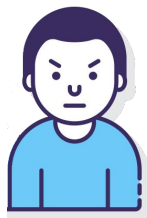
Revisar Depois:

- Live #64 - Composição de objetos
- Live #120 - Strategy

Disclaimer



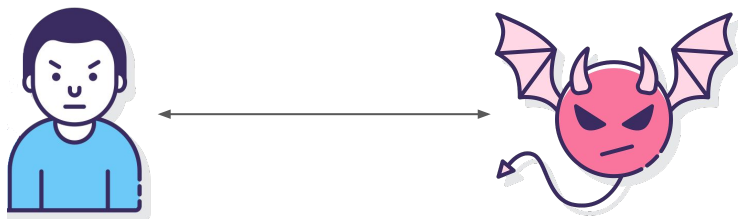
Fausto



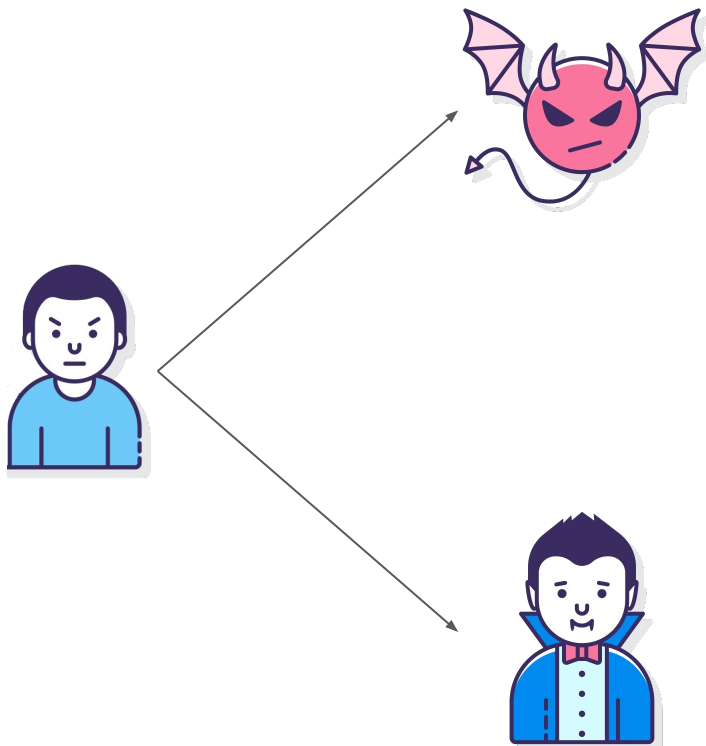
Vlad era um sujeito bem incomum. Às vezes até incompreendido.

Ele tinha a capacidade de mudar o estado de todas as suas habilidades. Tanto que uma vez chegou a ser chamado de polimorfo.

Mas Vlad era um vampiro.

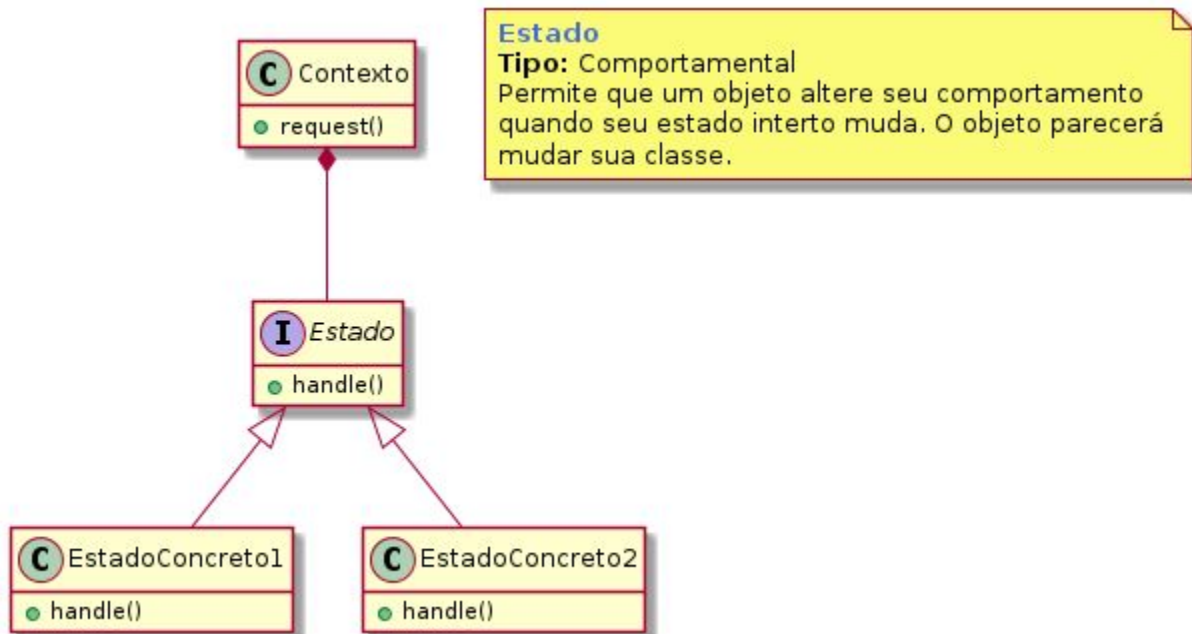


Vlad podia, quando queria, alterar seu estado e sair voando por aí. Com a forma de um morcego.

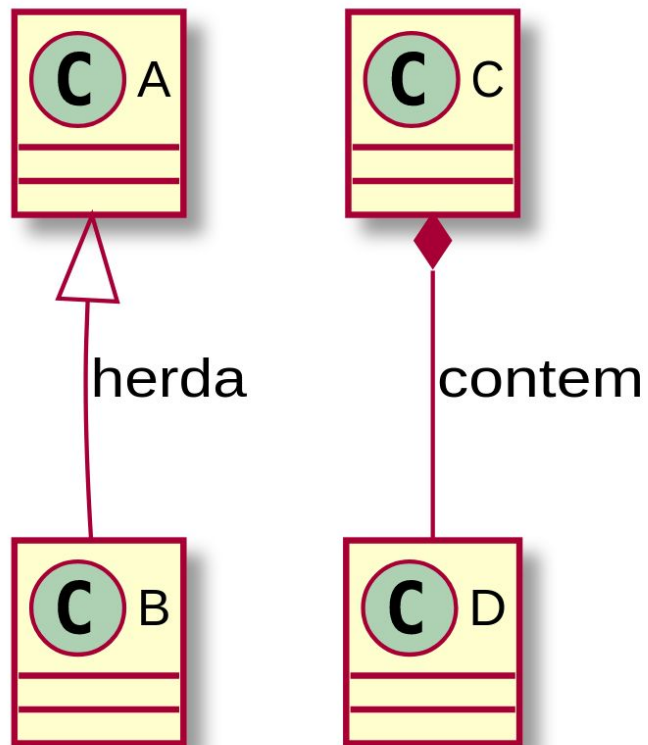


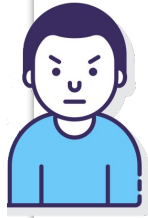
Em casa, sozinho a noite, ele vestia suas belas roupas de couro sintético e virava um vampirão.

Implementação em UML

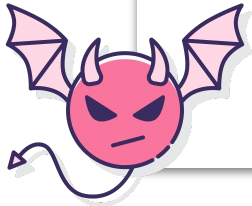


Ponto de atenção sobre UML

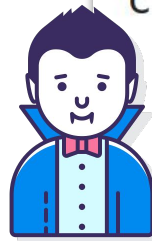




```
class Vlad:  
    """Classe de contexto."""  
    def __init__(self):  
        self.state = None  
  
    def se_mover(self):  
        self.state.se_mover()
```



```
class Morcego:  
    def se_mover(self):  
        print('Voando...')
```



```
class Vampiro:  
    def se_mover(self):  
        print('Andando...')
```

```
dracula = Vlad() # Estado inicial: None
```

```
dracula.state = Vampiro()  
dracula.se_mover()
```

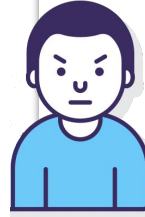
```
dracula.state = Morcego()  
dracula.se_mover()
```

```
class Vlad:
```

```
    """Classe de contexto."""
```

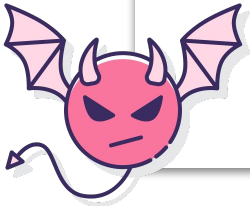
```
    def __init__(self):  
        self.state = None
```

```
    def se_mover(self):  
        self.state.se_mover()
```



```
class Morcego:
```

```
    def se_mover(self):  
        print('Voando...')
```



```
class Vampiro:
```

```
    def se_mover(self):  
        print('Andando...')
```



```
dracula = Vlad() # Estado inicial: None
```

```
dracula.state = Vampiro()  
dracula.se_mover()
```

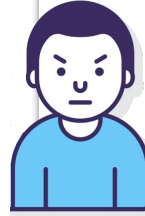
```
dracula.state = Morcego()  
dracula.se_mover()
```

```
class Vlad:
```

```
    """Classe de contexto."""
```

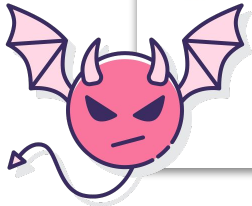
```
    def __init__(self):  
        self.state = None
```

```
    def se_mover(self):  
        self.state.se_mover()
```



```
class Morcego:
```

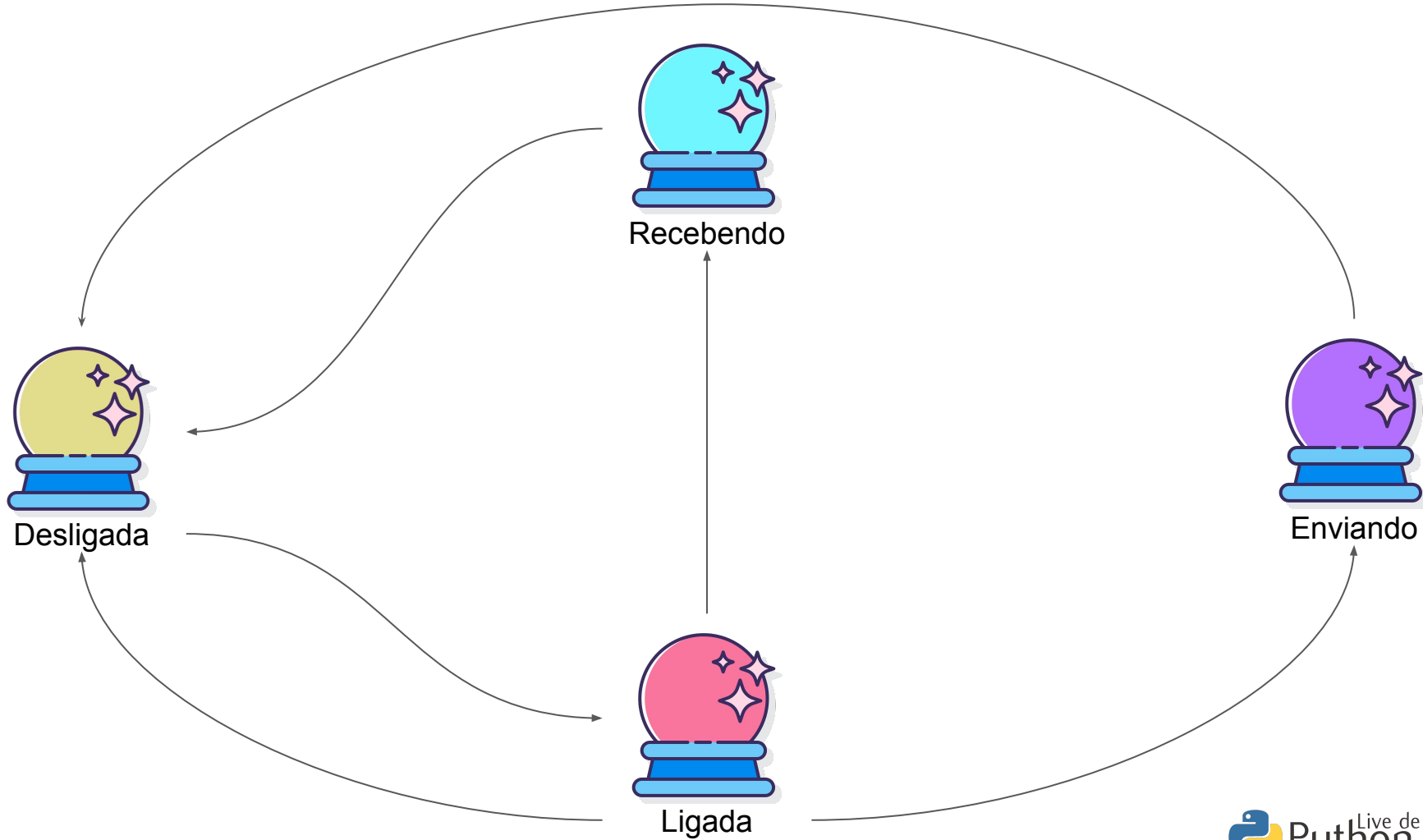
```
    def se_mover(self):  
        print('Voando...')
```



```
class Vampiro:
```

```
    def se_mover(self):  
        print('Andando...')
```





Vamos codar

Contras

1. Se você possui apenas alguns estados e eles raramente mudam, não há motivo real para complicar demais o programa com novas classes e interfaces que acompanham o padrão.
2. Dá pra simplificar com HOFs

Pontos de atenção

Design Patterns in Dynamic Programming

Peter Norvig
Chief Designer, Adaptive Systems
Harlequin Inc.

<http://www.norvig.com/design-patterns/design-patterns.pdf>

Are Design Patterns Missing Language Features

A list of `DesignPatterns` and a language feature that (largely) replaces them:

```
VisitorPattern ..... GenericFunctions (MultipleDispatch)
FactoryPattern ..... MetaClasses, closures
SingletonPattern ..... MetaClasses
IteratorPattern ..... AnonymousFunctions
                        (used with HigherOrderFunctions,
                        MapFunction, FilterFunction, etc.)
InterpreterPattern ..... Macros (extending the language)
                        EvalFunction, MetaCircularInterpreter
                        Support for parser generation (for differing syntax)
CommandPattern ..... Closures, LexicalScope,
                        AnonymousFunctions, FirstClassFunctions
HandleBodyPattern ..... Delegation, Macros, MetaClasses
RunAndReturnSuccessor ..... TailCallOptimization
Abstract-Factory,
Flyweight,
Factory-Method,
State, Proxy,
Chain-of-Responsibility ..... FirstClass types (Norvig)
Mediator, Observer ..... Method combination (Norvig)
BuilderPattern ..... Multi Methods (Norvig)
FacadePattern ..... Modules (Norvig)
StrategyPattern ..... higher order functions (Gene Michael
Stover?), ControlTable
AssociationList ..... Dictionaries, maps, HashTables
                        (these go by numerous names in different languages)
```

wiki.c2.com/?AreDesignPatternsMissingLanguageFeatures

Obrigado!

apoia.se/livedepython

```
dunossauro at unknown in ~/git/apoiase on master*  
$ python apoiadores.py  
Alexandre Possebom   Alexandre Tsuno       Alysson Oliveira      Amaziles Carvalho  
And Past             Andre Machado         Andre Rodrigues       Athayr Athayr  
Bernardo Fontes     Bruno Oliveira        Bruno Gaffuri         Bruno Guizi  
Bruno Rocha         Carlos Augusto        Cleber Santos        Cleiton Mittmann  
DYLAN MAKER         David Reis            David Silva           Dayham Soares  
Diego Ubirajara     Edimar Fardim        Eliabe Silva         Eliakim Moraes  
Elias Soares       Eliel Lima           Emerson Lara         Eugenio Mazzini  
Fabiano Silos       Fabiano Teichmann    Fabiano Gomes        Fausto Caldeira  
Fernando Furtado    FlavKaze FlavKaze     Franklin Silva        Fábio Serrão  
Gleison Oliveira    Guilherme Ramos       Hemilio Lauro        Humberto Rocha  
Hélio Neto          JONATHAN DOMINGOS    Jean Vetorello       Johnny Tardin  
Jonatas Oliveira    Jonatas Simões       José Prado           João Lugão  
João Coelho         Juan Gutierrez        Jucélio Silva        Júlia Kastrup  
Kauan Alves         Leon Teixeira         Lucas Nascimento     Magno Malkut  
Marcello Benigno    Marcus Salgues       Maria Boladona       Matheus Francisco  
Nilo Pereira        Nídio Dolfini        Pablo Henrique       Patrick Corrêa  
Paulo Tadei         Pedro Alves          Rafael Galleani      Regis Santos  
Renan Gomes         Renan Moura          Renato Santos        Rennan Almeida  
Renne Rocha        Rhenan Bartels       Rodrigo Ferreira     Rodrigo Vaccari  
Sérgio Passos       Thais Viana          Thiago Araujo        Tiago Cordeiro  
Tyrone Damasceno    Vergil Valverde      Vicente Marcal       Vinícius Ferreira  
Wander Silva        Wellington Carlos     Wellington Camargo    Welton Souza  
William Oliveira    Willian Gl           Yros Aguiar          Falta você
```