



Estácio

Paradigmas de Linguagem Utilizando Python

Prof. Cristiano Soares de Aguiar

Uma Breve Apresentação?

Cristiano Soares de Aguiar



Graduado em Ciência da Computação (2002)

Mestrado em Computação (2005)

Especialista em Computação Forense (2023)

Especialização em Inteligência Artificial (cursando, 2024)


Doutorando em Propriedade Intelectual e Inovação (cursando)

Professor Universitário (Rio de Janeiro, Brasília, Goiânia) desde 2004

Contato:  crsaguiar2@gmail.com

 www.linkedin.com/in/cristianoAguiar02

 Lattes: <http://lattes.cnpq.br/6477870357668952>

 (61) 9.8195-0210

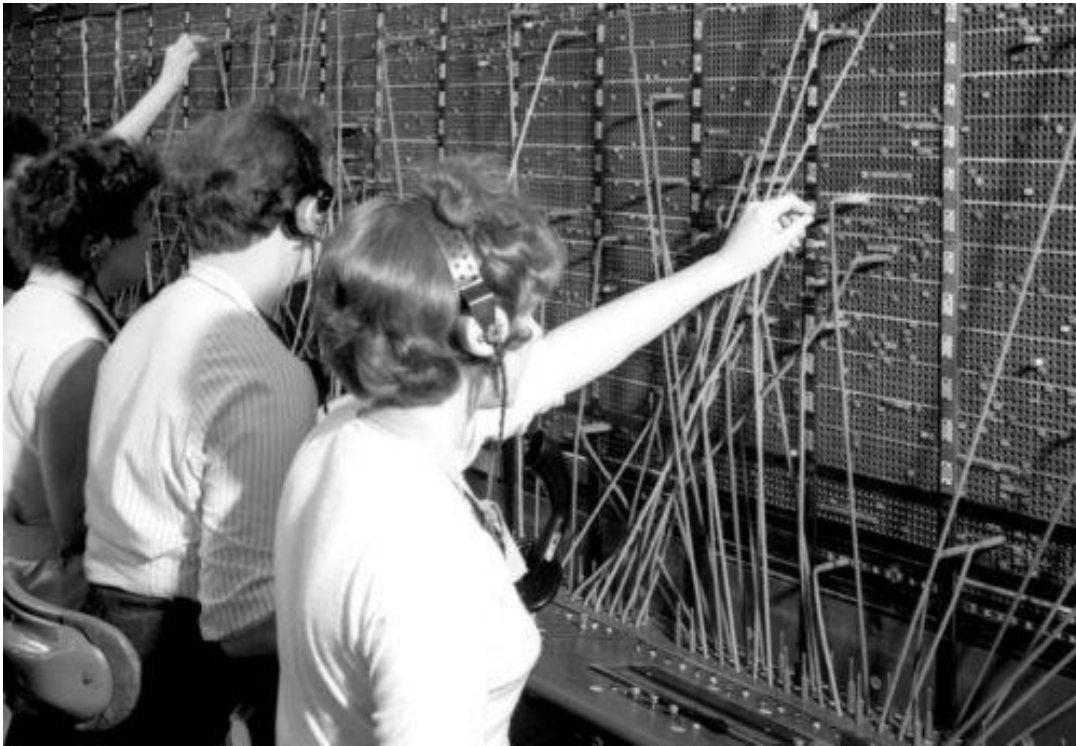
O que Veremos?

- 1. O que são Paradigmas de Linguagem?**
- 2. Evolução dos Paradigmas de Linguagem.**
- 3. Porque utilizar a Linguagem Python?**
- 4. Como as Linguagens Funcionam?**
- 5. E o Futuro dos Paradigmas?**
- 6. Bibliografia;**

Níveis de Abstração?

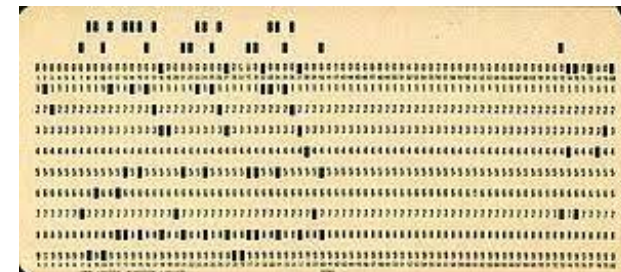
Paradigmas de Linguagem:

- Paradigmas de Linguagens estão relacionados aos níveis de abstração de construção de um Programa de Computador.



- No início da computação, os computadores eram programados em nível de **hardware**. Por óbvio, algo bastante trabalhoso!

→ Circuitos Eletrônicos



Níveis de Abstração?

Evolução dos Paradigmas:

Instrução	Instrução
mov eax,5	mov eax,5
mark 1	add eax,1
emark 1	mov ebx,3
mov ebx,3	add eax,ebx
add eax,ebx	cmp eax,8
cmp eax,8	je label1
je label1	add eax,1
add eax,1	label1:
label1:	adapt 1,dado
adapt 1,dado	

Programa Não Estruturado

- **Paradigma Não Estruturado.** Exemplo: Assembly.
→ Economia de hardware: memória e processador.



Níveis de Abstração?

Evolução dos Paradigmas:

```
#include <stdio.h>
#include <iso646.h>

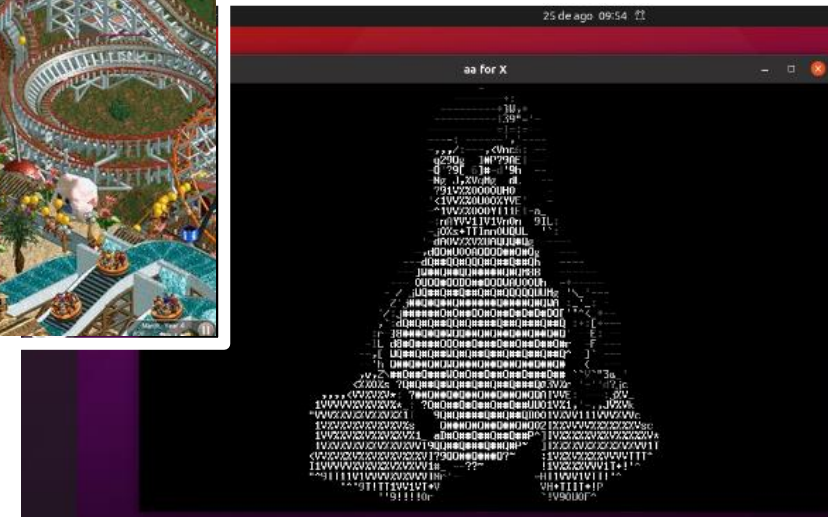
void main(){

    int a = 1, b=2;

    if(a==0 and b==0)
        printf("A e B sao zero");
    else if (a==1 or b==1)
        printf("Pelo menos um valor eh um");
    else if (a==2 && b==2)
        printf("A e B sao iguais a dois");
}
```

Programa Estruturado

- **Paradigma Estruturado.** Exemplo: C.
→ Modularidade e Legibilidade.



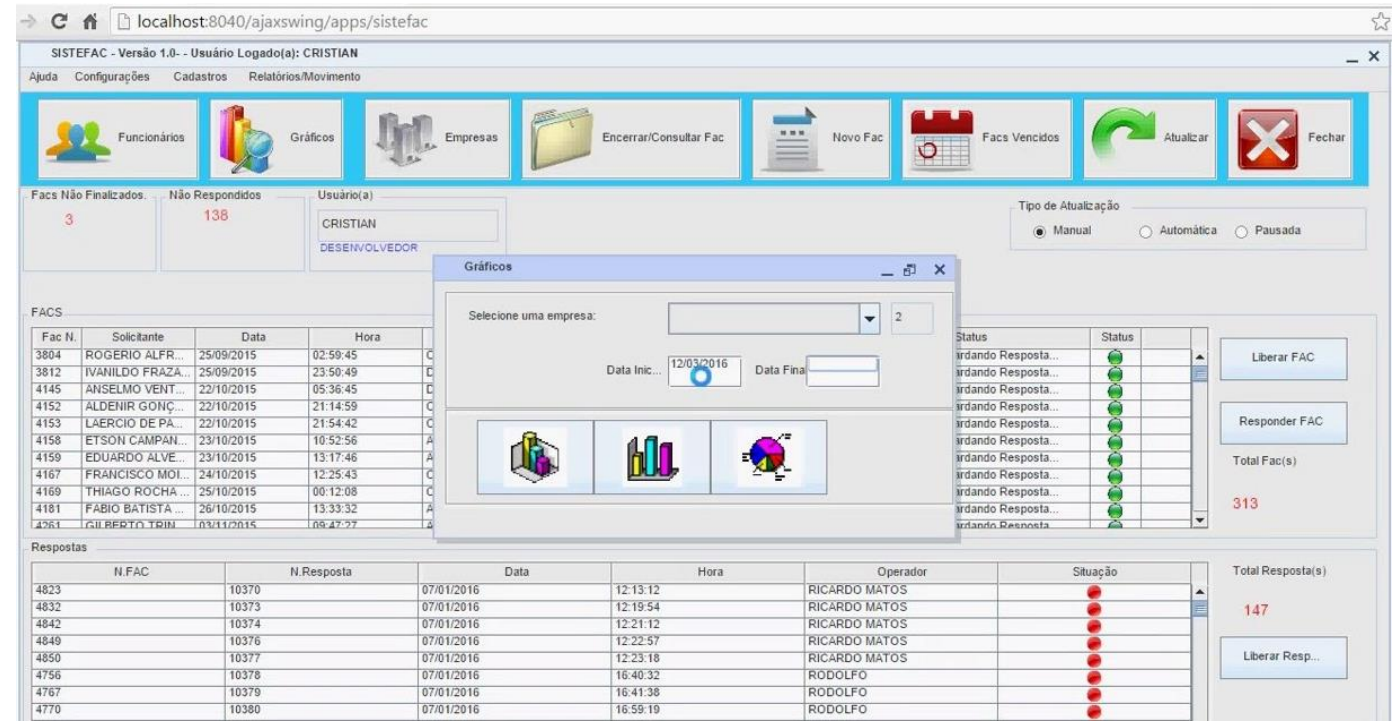
Níveis de Abstração?

Evolução dos Paradigmas:

```
public class ProdutoDAO {  
  
    public List<Produto> lista(){  
  
        // Implementação qualquer que devolve uma lista de produtos  
  
    }  
  
    public void produtosResumido(){  
        List<Produto> prs = lista();  
        for (Produto p : prs) {  
            String dse = p.getDescricao().trim();  
            StringBuilder sb = new StringBuilder();  
            for (int i = 0; i < dse.length(); i++) {  
                if(i > 15) {  
                    sb.append("...");  
                    break;  
                }  
                sb.append(dse.charAt(i));  
            }  
  
            String da = sb.toString();  
            String vf = p.getValor().toString().replace(".", ",");  
            String df = p.getId() + " - " + da + " - R$ " + vf;  
            System.out.println(df);  
        }  
    }  
}
```

Programa OO

- Paradigma OO. Exemplo: Java.
→ Maior Abstração Mundo Real: Encapsulamento.



Níveis de Abstração?

Evolução dos Paradigmas:

```
# Defines how many layers to freeze during training.
# Layers in the convolutional base are switched from trainable to non-trainable
# depending on the size of the fine-tuning parameter.
if fine_tune > 0:
    for layer in conv_base.layers[:-fine_tune]:
        layer.trainable = False
else:
    for layer in conv_base.layers:
        layer.trainable = False

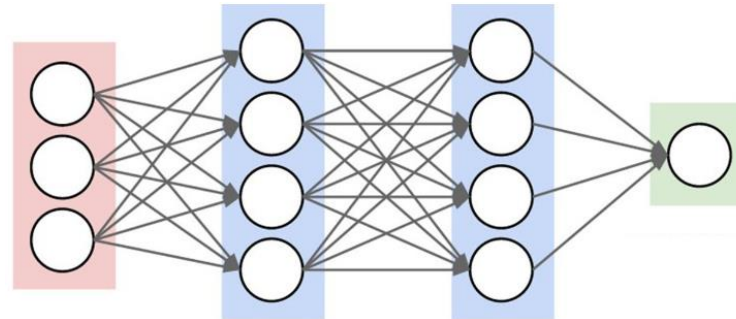
# Create a new 'top' of the model (i.e. fully-connected layers).
# This is 'bootstrapping' a new top_model onto the pretrained layers.
top_model = conv_base.output
top_model = Flatten(name="flatten")(top_model)
top_model = Dense(4096, activation='relu')(top_model)
top_model = Dropout(0.2)(top_model)
top_model = Dense(2048, activation='relu')(top_model)
top_model = Dropout(0.2)(top_model)
top_model = Dense(1024, activation='relu')(top_model)
top_model = Dropout(0.2)(top_model)
output_layer = Dense(n_classes, activation='softmax')(top_model)

# Group the convolutional base and new fully-connected layers into a Model object.
model = Model(inputs=conv_base.input, outputs=output_layer)

# Compiles the model for training.
model.compile(optimizer=optimizer,
              loss='categorical_crossentropy',
              metrics=['accuracy'])
```

Programa ML

- **Paradigma ML.** Exemplo: Python.
→ Ausência de Programação de Regras.



Por que Python?



Python suporta quase todos os paradigmas!

→ Linguagem década de 90

```
31 def __init__(self, settings):
32     self.file = None
33     self.fingerprints = set()
34     self.logdups = True
35     self.debug = debug
36     self.logger = logging.getLogger(__name__)
37     if path:
38         self.file = open(os.path.join(path, "requests.log"),
39                         "a")
40         self.file.seek(0)
41         self.fingerprints.update(s.request for s in self.requests)
42
43 @classmethod
44 def from_settings(cls, settings):
45     debug = settings.getbool("SUPERDEBUG")
46     return cls(job_dir(settings), debug)
47
48 def request_seen(self, request):
49     fp = self.request_fingerprint(request)
50     if fp in self.fingerprints:
51         return True
52     self.fingerprints.add(fp)
53     if self.file:
54         self.file.write(fp + os.linesep)
55
56 def request_fingerprint(self, request):
57     return request_fingerprint(request)
```

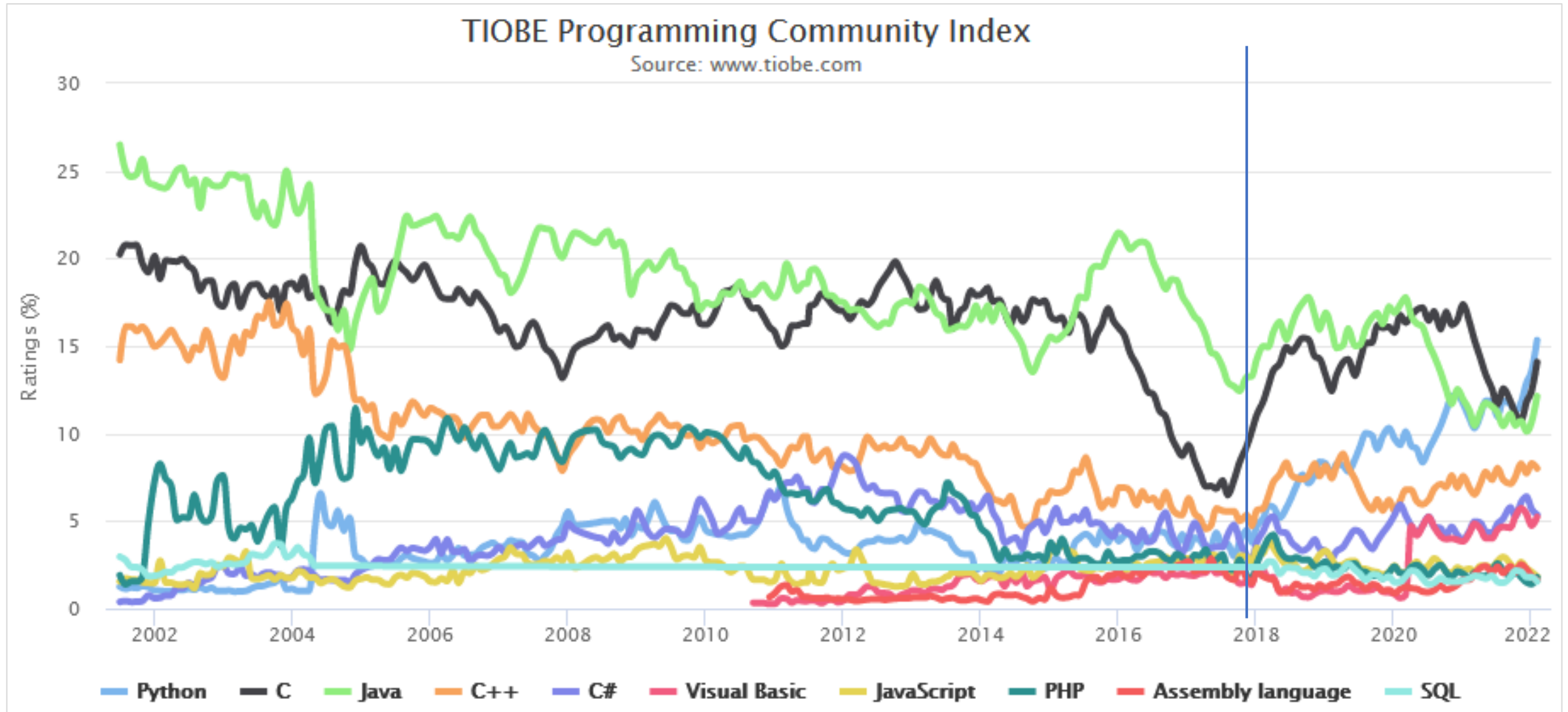


Monty Python



Estácio

Como o Mundo vê a Linguagem Python?



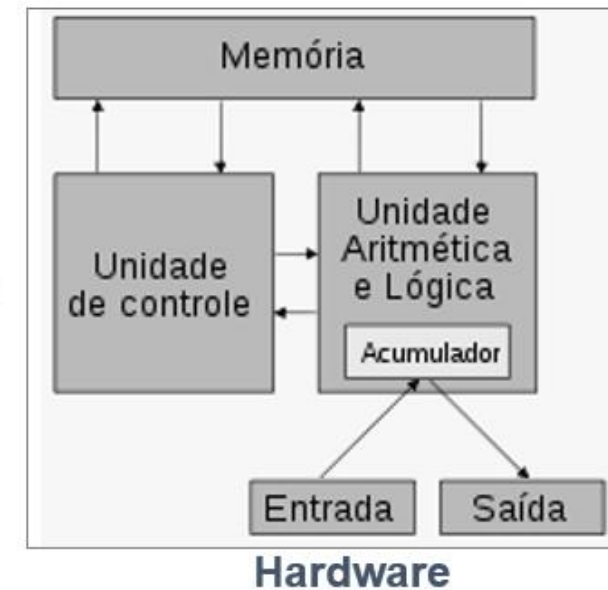
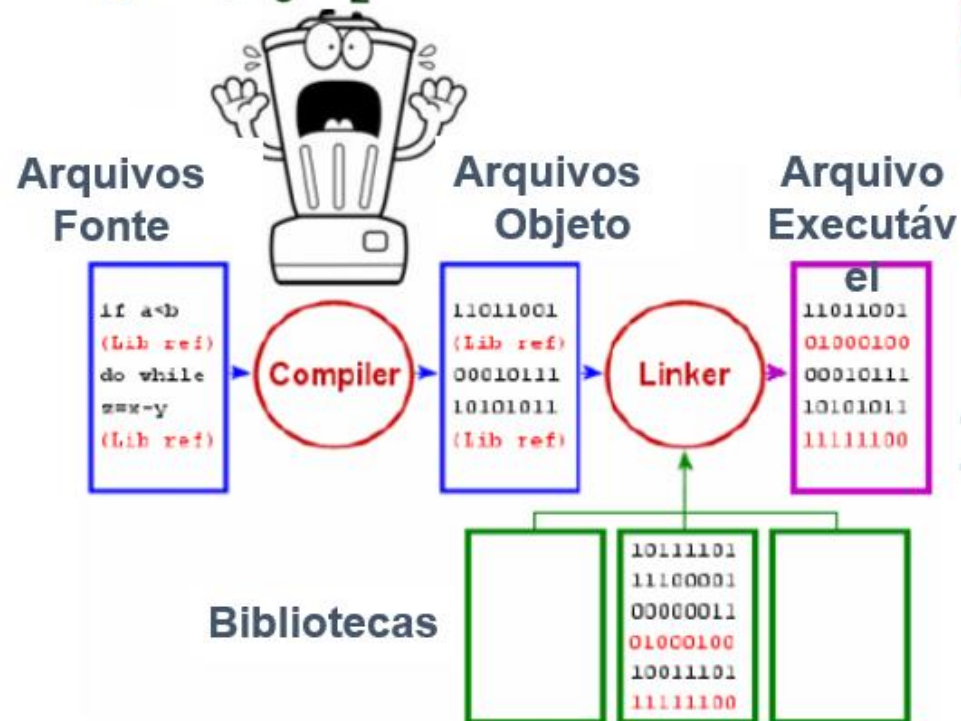
O que é Possível Fazer com Python?



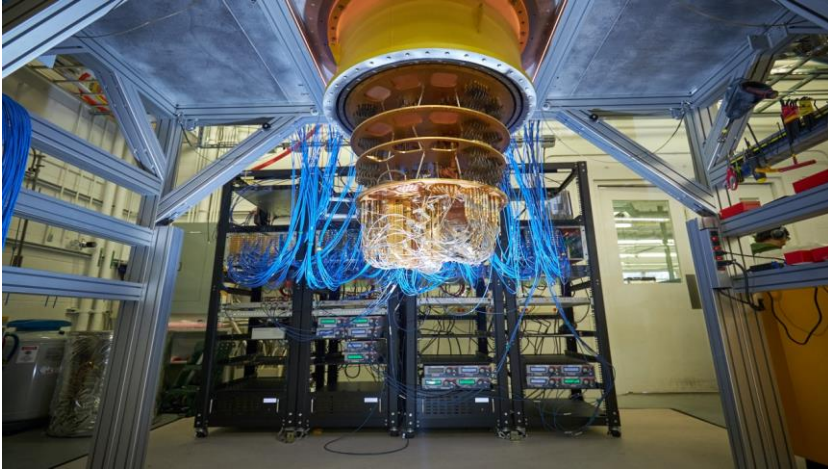
Como as Linguagens Funcionam?



Software Embarcado



E o Futuro dos Paradigmas?



▪ **Computação Quântica e Big Data**



Minority Report - A Nova Lei
2002 ▪ Ficção científica/Ação ▪ 2h 25m



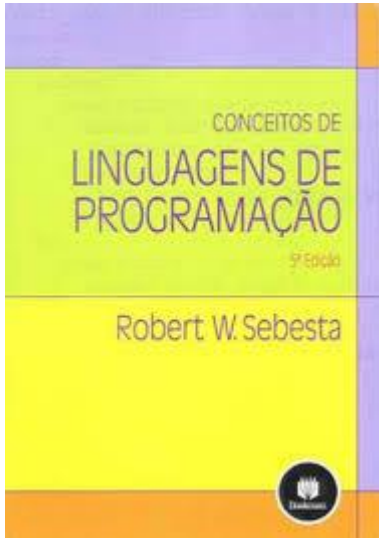
Para quem Gosta de Ciência e Tecnologia



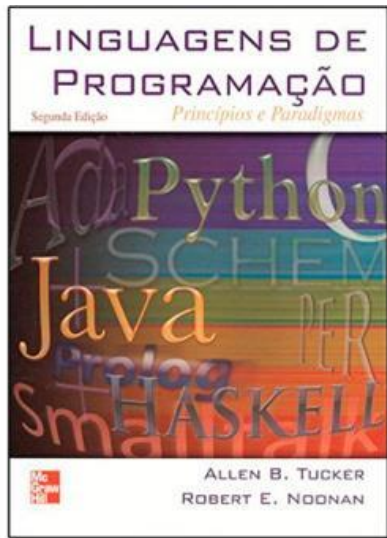
O Jogo da Imitação

2014 ▪ Drama/Guerra ▪ 1h 54m

Referências Bibliográficas



Sebesta, R. W., **Conceitos de Linguagens de Programação**; Editora Bookman. 2003.



Tucker, A. B. et al., **Linguagens de Programação: Princípios e Paradigmas**; Editora McGraw Hill. 2009.



MENEZES, N. N. C. **Introdução à Programação com Python: Algoritmos e Lógica de Programação Para Iniciantes**. Novatec. 2019.



Estácio