

# Cinco linguagens para você aprender... e por quê?

Fernando Castor

Centro de Informática – Universidade Federal de Pernambuco

Alguns direitos reservados



# Linguagens de Programação

- Servem para **dizer ao computador o que fazer**
- **Independentemente da máquina** subjacente
- **Legíveis** para humanos



# Linguagens de Programação

- Servem para **dizer ao computador o que fazer**
- **Independentemente da máquina** subjacente
- **Legíveis** para humanos

Hello World!

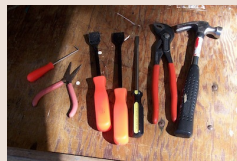
```
C      Hello World in Fortran 77

      PROGRAM HELLO
      PRINT*, 'Hello, World!'
      END
```

# Linguagens de Programação são **Ferramentas**

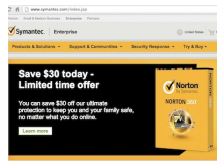
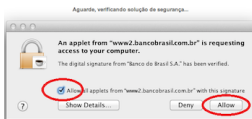
Cada uma se presta a um fim

- **Fortran:** processamento numérico
- **Cobol:** descrição de dados e aplicações comerciais
- **Perl:** processamento de cadeias de caracteres



# Esse fim **pode mudar** com o tempo

## Exemplo: Java





Em 22 de março de 2017, há 738 “LPs” na lista da Wikipédia.

# Certo, são muitas linguagens, mas...

- 1 Quais são as linguagens de programação mais usadas na prática?
- 2 Se você quisesse aprender cinco linguagens para o futuro, quais deveriam ser?



# Parte 1

Quais são as linguagens de programação mais usadas na prática?





# Não há um censo oficial sobre uso de linguagens

- Resposta precisa vir de várias fontes
- Com diferentes contextos
- Dados são **difíceis de comparar**



# O que significam “ser usada” e “na prática”?

- Mais **linhas de código** escritas na linguagem?



# O que significam “ser usada” e “na prática”?

- Mais **linhas de código** escritas na linguagem?
- Número de **commits** de programas que usam a linguagem?



# O que significam “ser usada” e “na prática”?

- Mais **linhas de código** escritas na linguagem?
- Número de **commits** de programas que usam a linguagem?
- Mais **“projetos”** com linhas de código escritas na linguagem?



# O que significam “ser usada” e “na prática”?

- Mais **linhas de código** escritas na linguagem?
- Número de **commits** de programas que usam a linguagem?
- Mais **“projetos”** com linhas de código escritas na linguagem?
  - O que é um “projeto”?
  - **Forks** contam como projetos separados?
  - Se usar apenas **código de terceiros** conta?



# O que significam “ser usada” e “na prática”?

- Mais **linhas de código** escritas na linguagem?
- Número de **commits** de programas que usam a linguagem?
- Mais **“projetos”** com linhas de código escritas na linguagem?
  - O que é um “projeto”?
  - **Forks** contam como projetos separados?
  - Se usar apenas **código de terceiros** conta?
- Onde a **linha** da “prática” é **traçada**?



# O que significam “ser usada” e “na prática”?

- Mais **linhas de código** escritas na linguagem?
- Número de **commits** de programas que usam a linguagem?
- Mais **“projetos”** com linhas de código escritas na linguagem?
  - O que é um “projeto”?
  - **Forks** contam como projetos separados?
  - Se usar apenas **código de terceiros** conta?
- Onde a **linha** da “prática” é **traçada**?
- É relevante saber se desenvolvedores **falam** muito **sobre**?



Em resumo:  
é necessário adotar  
diferentes perspectivas.





# Felizmente...

- Alguém já fez **parte** do trabalho
- TIOBE Programming Community Index

*... is an indicator of the popularity of programming languages [...] based on the **number of skilled engineers** world-wide, **courses** and **third party vendors**. The **popular search engines** Google, Bing, Yahoo!, Wikipedia [...] are used to calculate the ratings. [...] TIOBE index is not about [...] the language in which **most lines of code have been written**.*



De acordo com a TIOBE, o ranking em março de 2017 era este:

Mar 2017	Mar 2016	Change	Programming Language	Ratings	Change
1	1		Java	16.384%	-4.14%
2	2		C	7.742%	-6.86%
3	3		C++	5.184%	-1.54%
4	4		C#	4.409%	+0.14%
5	5		Python	3.919%	-0.34%
6	7	▲	Visual Basic .NET	3.174%	+0.61%
7	6	▼	PHP	3.009%	+0.24%
8	8		JavaScript	2.667%	+0.33%
9	11	▲	Delphi/Object Pascal	2.544%	+0.54%
10	14	▲▲	Swift	2.268%	+0.68%
11	9	▼	Perl	2.261%	+0.01%
12	10	▼	Ruby	2.254%	+0.02%
13	12	▼	Assembly language	2.232%	+0.39%
14	16	▲	R	2.016%	+0.73%
15	13	▼	Visual Basic	2.008%	+0.33%

De acordo com a TIOBE, o ranking em março de 2017 era este:

Mar 2017	Mar 2016	Change	Programming Language	Ratings	Change
1	1		Java	16.384%	-4.14%
2	2		C	7.742%	-6.86%
3	3		C++	5.184%	-1.54%
4	4		C#	4.409%	+0.14%
5	5		Python	3.919%	-0.34%
6	7	⬆	Visual Basic .NET	3.174%	+0.61%
7	6	⬇	PHP	3.009%	+0.24%
8	8		<u>JavaScript</u>	2.667%	+0.33%
9	11	⬆	Delphi/Object Pascal	2.544%	+0.54%
10	14	⬆	Swift	2.268%	+0.68%
11	9	⬇	<u>Perl</u>	2.261%	+0.01%
12	10	⬇	Ruby	2.254%	+0.02%
13	12	⬇	Assembly language	2.232%	+0.39%
14	16	⬆	R	2.016%	+0.73%
15	13	⬇	Visual Basic	2.008%	+0.33%

## Relembrando:

... is an indicator of the popularity of programming languages [...] based on the **number of skilled engineers** world-wide, **courses** and **third party vendors**. The **popular search engines** Google, Bing, Yahoo!, Wikipedia [...] are used to calculate the ratings. [...] TIOBE index is **not** about [...] the language in which **most lines of code have been written**.



## Relembrando:

... is an indicator of the popularity of programming languages [...] based on the **number of skilled engineers** world-wide, **courses** and **third party vendors**. The **popular search engines** Google, Bing, Yahoo!, Wikipedia [...] are used to calculate the ratings. [...] TIOBE index is **not** about [...] the language in which **most lines of code have been written**.

Problema do ranking do índice da TIOBE:  
não menciona diretamente **código** escrito nas linguagens ou **número de repositórios ou projetos** que as usam.

# Obtendo informações sobre código

- Repositórios de software de código aberto (*forges*)
  - Os maiores incluem dezenas de milhares de projetos
  - E já fornecem algumas estatísticas

Examinaremos os dados de dois desses repositórios:



# Dados do SourceForge (de 10/2015)

Programming Language ▾	St
Java (53,572)	
C++ (43,187)	
PHP (33,775)	
C (31,835)	
C# (17,050)	
Python (16,583)	
JavaScript (13,881)	
Perl (10,012)	
Unix Shell (4,775)	
Visual Basic .NET (4,094)	
Delphi/Kylix (4,050)	
Visual Basic (3,684)	
Assembly (2,747)	
JSP (2,173)	
Ruby (2,062)	

**Entre parênteses:** número de projetos que declaram usar a linguagem



# Dados do SourceForge (de 10/2015)

Programming Language ▾	St
Java (53,572)	
C++ (43,187)	
PHP (33,775)	
C (31,835)	
C# (17,050)	
Python (16,583)	
JavaScript (13,881)	
Perl (10,012)	
Unix Shell (4,775)	
Visual Basic .NET (4,094)	
Delphi/Kylix (4,050)	
Visual Basic (3,684)	
Assembly (2,747)	
JSP (2,173)	
Ruby (2,062)	

**Entre parênteses:** número de projetos que declaram usar a linguagem

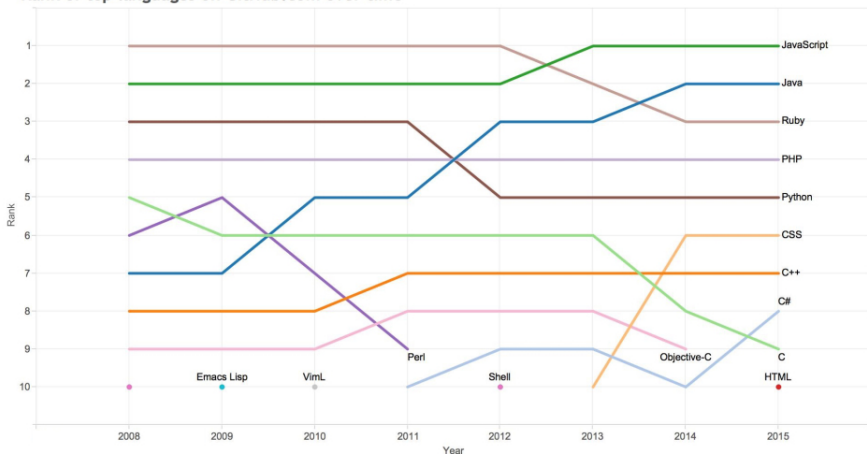
**R, Perl e Swift** aparecem no Top 15 da TIOBE mas **não aqui**





# Dados do Github (10/2015)

Rank of top languages on GitHub.com over time



# E sobre quais os programadores estão falando?

Tags do StackOverflow.com  
em março/2017:

- 1 JavaScript (1347902)
- 2 Java (1229513)
- 3 C# (1073182)
- 4 PHP (1052878)
- 5 Python (720882)
- 6 C++ (503932)
- 7 CSS (457529)
- 8 SQL (377801)
- 9 asp.net (312036)
- 10 Objective-C (274621)
- 11 Ruby (270146)
- 12 C (245446)
- 13 R (175524)
- 14 Swift (140342)

**Entre parênteses:** número de  
posts com aquela tag



# E sobre quais os programadores estão falando?

Tags do StackOverflow.com  
em março/2017:

- 1 JavaScript (1347902)
- 2 Java (1229513)
- 3 C# (1073182)
- 4 PHP (1052878)
- 5 Python (720882)
- 6 C++ (503932)
- 7 CSS (457529)
- 8 SQL (377801)
- 9 asp.net (312036)
- 10 Objective-C (274621)
- 11 Ruby (270146)
- 12 C (245446)
- 13 R (175524)
- 14 Swift (140342)

Entre parênteses: número de  
posts com aquela tag

Isso não é tudo!

JQuery, AJAX, Node.js,  
AngularJS ⇒ **JavaScript**

Django ⇒ **Python**

Rails ⇒ **Ruby**

Android ⇒ **Java**

iPhone, iOS, Xcode1 ⇒  
**Objective-C e Swift.**



# The RedMonk Programming Language Rankings (01/2017)



# Top 10 a partir de diversas fontes

Github	SourceForge	TIOBE	StackOverflow	Redmonk
JavaScript	Java	Java	JavaScript	JavaScript
Java	C++	C	Java	Java
Ruby	PHP	C++	C#	PHP
PHP	C	C#	PHP	Python
Python	C#	Python	Python	C++
CSS	Python	VB.NET	C++	C#
C++	JavaScript	PHP	CSS	CSS
C#	Perl	JavaScript	SQL	C
C	Shell	Delphi	asp.net	Objective-C
HTML (???)	VB.NET	Swift	Objective-C	Ruby

# Top 10 a partir de diversas fontes

Github	SourceForge	TIOBE	StackOverflow	Redmonk
JavaScript	Java	Java	JavaScript	JavaScript
Java	C++	C	Java	Java
Ruby	PHP	C++	C#	PHP
PHP	C	C#	PHP	Python
Python	C#	Python	Python	C++
CSS	Python	VB.NET	C++	C#
C++	JavaScript	PHP	CSS	CSS
C#	Perl	JavaScript	SQL	C
C	Shell	Delphi	asp.net	Objective-C
HTML (???)	VB.NET	Swift	Objective-C	Ruby

**Importante:** não representam a indústria **como um todo**.

# Parte 2

Se você quisesse aprender cinco linguagens para o futuro, quais deveriam ser?



Cinco linguagens de programação para que? Quais os critérios para escolhê-las?





# Meus critérios

Nada de linguagens “acadêmicas”

Não devem servir apenas como prova de conceito para uma ideia



# Meus critérios

Nada de linguagens “acadêmicas”

Não devem servir apenas como prova de conceito para uma ideia

Linguagens que forneçam **lições amplas**

- linguagens que se possa empregar diretamente
- e linguagens que ajudem a usar melhor outras linguagens

# Meus critérios

Nada de linguagens “acadêmicas”

Não devem servir apenas como prova de conceito para uma ideia

Linguagens que forneçam **lições amplas**

- linguagens que se possa empregar diretamente
- e linguagens que ajudem a usar melhor outras linguagens

Linguagens de programação para você **aprender**



# Meus critérios

Levando em conta certos nichos de aplicação e cenários de uso

- 1 Primeira linguagem de programação para aprender
- 2 Desenvolvimento Móvel
- 3 Programação multi-núcleo e sistemas confiáveis
- 4 Desenvolvimento para a Web
- 5 Software de baixo nível e de alto desempenho

# 1. Primeira Linguagem de Programação para Aprender



# Olá, Mundo!

```
class HelloWorld {  
    static public void main( String args[] ) {  
        System.out.println(" Hello World!");  
    }  
}
```

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

```
int main(void) {  
    puts(" Hello World!");  
}
```

```
10 PRINT " Hello World!"
```

```
(defun helloworld ()  
  (print " Hello World!")  
)
```

```
print " Hello World!"
```

```
package main  
import "fmt"  
func main() {  
    fmt.Printf(" Hello World!")  
}
```

# Linguagem 1



# Linguagem 1







## Porquês

- Sintaxe amigável
- Dinamicamente tipificada
- Multiparadigma
- Muito popular na prática
- Usada como primeira linguagem em vários cursos de computação

# Alternativas

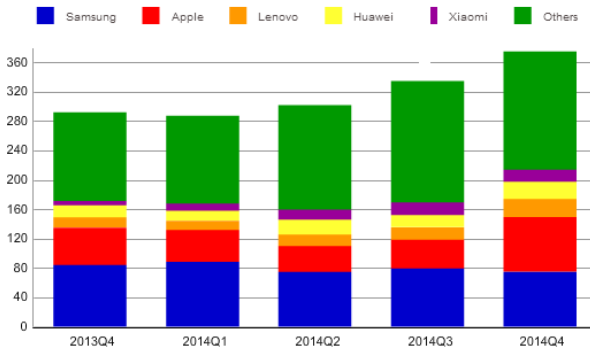


## 2. Desenvolvimento Móvel



# Vendas de Smartphones no mundo

Top 5 WW Smartphone Vendors, 2014Q4  
Unit Shipments (Millions)



Fonte: SmartphoneMarketResearch.com



# Para Android: Java



# Para Android: Java



Também é possível usar JavaScript



Mas...



Mas...





# Para iOS: Objective-C



**Objective-C**

# Para iOS: Objective-C

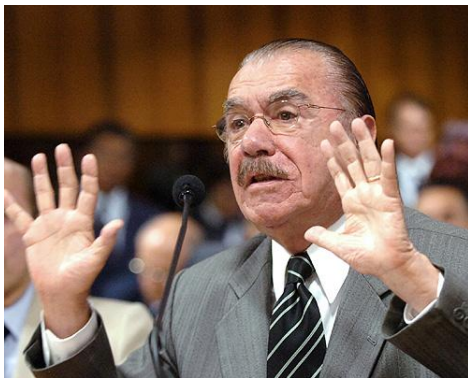


**Objective-C**

Também é possível usar JavaScript<sup>2</sup>

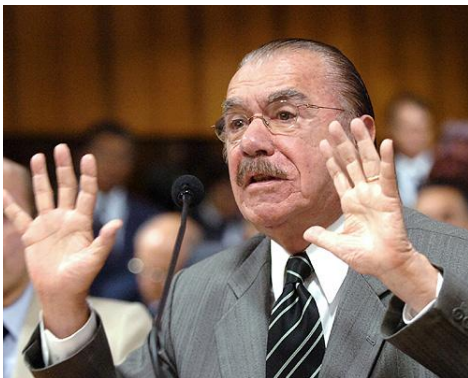


Mas...



Objective-C...

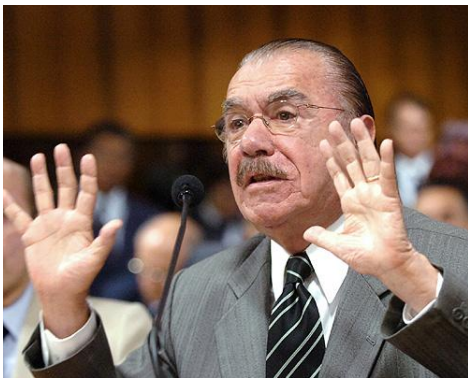
# Mas...



## Objective-C...

- tem uma aparência desagradável

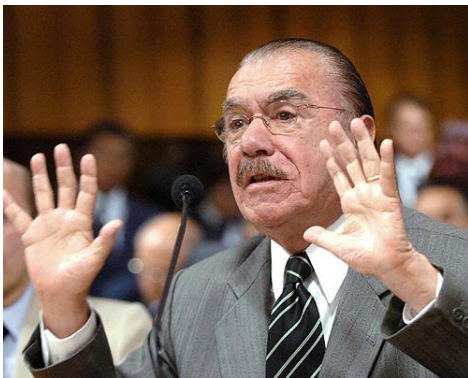
# Mas...



## Objective-C...

- tem uma aparência desagradável
- tem uma propensão a causar corrupção

# Mas...



## Objective-C...

- tem uma aparência desagradável
- tem uma propensão a causar corrupção
- está por aí há tempo demais (alguns dizem que é **imortal**)

# Linguagem 2



*The growth that Swift experienced is essentially unprecedented in the history of these rankings.*

<http://redmonk.com/sogradys/2015/01/14/language-rankings-1-15/>





# TIOBE Index for March 2017

## March Headline: Swift enters the top 10 for the first time

This month Swift enters the TIOBE index top 10 for the first time. About 3 years ago Apple replaced its app development language Objective-C by its newborn language called Swift. The expectation was that Swift would rise quickly, but it has been slow. That is for good reasons by the way because the installed base of Objective-C code is very large. Old apps that are written in Objective-C are not actively migrated to Swift. This is also what makes Swift a good choice to write applications in Apple's ecosystem, it is expected that it won't rise much further.



# Por que Swift?

Orientada a Objetos + Funcional

Estaticamente tipificada, mas sem inchaço

*Playgrounds* interativos

Código aberto: ecossistema Apple + Linux



# Ainda não dá...

para fazer apps Android em Swift.





Georgie J

Follow

Jul 8, 2016 · 8 min read

## How we put an app in the Android Play Store using Swift

*This is the second (more technical) part of our story about putting what was possibly the first app on the Play Store with a significant Swift component. [See here](#) for part one.*

As I described in the [first part](#) of this series, we had time and business pressure to release a [flowkey](#) Android app, with no proven technology path to achieve it. In summary, we decided to take our chances that Swift would become available to us on Android to run our performance-critical Pitch Detection code once Swift went open source (at a then-unknown point in time) *without* having to rewrite that code in C(++). This would accompany our hybrid largely HTML5-based app which had already proven itself performant and compatible running on Android up until then.



# Alternativas (obviamente)



# Alternativas (obviamente)



E tem também

JavaScript



### 3. Programação Multi-Núcleo e Sistemas Confiáveis



<http://is1.minnesotacomputers.net/images/productimages/M605.gif>

[http://www.ps3-systems.com/images/playstation\\_3.jpg](http://www.ps3-systems.com/images/playstation_3.jpg)

<http://www.zeldauniverse.net/wp-content/uploads/2010/07/oot3dsblack.png>

[http://www.tudocelular.com/new\\_files/images/global/Samsung-Galaxy-Note\\_53032\\_1.jpg](http://www.tudocelular.com/new_files/images/global/Samsung-Galaxy-Note_53032_1.jpg)

<http://maxcdn.liewcf.com/blog/wp-content/uploads/ipad-2-wifi.jpg>



Multi-núcleo  $\Rightarrow$   
concorrência e paralelismo



# Problemas oriundos de programação multi-núcleo

- Condições de corrida
- Falhas de atomicidade
- Deadlocks, livelocks, starvation
- Morte prematura de threads
- Custo de criação e chaveamento entre threads
- Quebra de trabalho entre threads, sincronização



## Algumas causas fundamentais

- **Estado compartilhado mutável** é a raiz do mal.
- **Falta** de mecanismos para **tratamento de erros** em cenários concorrentes e distribuídos.
- **Threads do sistema operacional** como unidades de quebra de trabalho.

# Linguagem 3

# Linguagem 3





## Porquês

- **Filosofia Let It Crash**
- Puramente funcional e sem estado compartilhado
- Processos leves – unidades finas de quebra de trabalho
- Concorrência e distribuição são **inerentes** à linguagem
  - É impossível construir sistemas não-triviais ignorando isso

Nenhuma.



# 4. Desenvolvimento para a Web



# Qual o atual estado da prática de desenvolvimento para a Web?

(de acordo com a Wikipédia)



**Lado servidor:** ASP, CSP, Server-Side ANSI C, ColdFusion, CGI, Groovy (Grails), Java, Lotus Domino, Perl, PHP, Python (Django), Real Studio Web Edition, Ruby (Rails), Node.js, Server-Side JavaScript, Mozilla Rhino, Websphere, .NET, (...)

**Lado cliente:** Ajax, Flash, JavaScript, jQuery, Silverlight, HTML5



# JavaScript é um padrão de fato, mas...

- $[] + [] =$



# JavaScript é um padrão de fato, mas...

- `[] + [] = string vazio!`
- `[] + {} =`



# JavaScript é um padrão de fato, mas...

- `[] + [] = string vazio!`
- `[] + {} = [object Object]?!`
- `{ } + [] =`

# JavaScript é um padrão de fato, mas...

- `[] + [] = string vazio!`
- `[] + {} = [object Object]?!`
- `{ } + [] = 0!!!`
- `{ } + { } =`

# JavaScript é um padrão de fato, mas...

- $[] + [] = \text{string vazio!}$
- $[] + \{\} = [\text{object Object}]?!$
- $\{\} + [] = 0!!!$
- $\{\} + \{\} = \text{NaN!}$
- $\{\} - \{\} =$

# JavaScript é um padrão de fato, mas...

- $[] + [] = \text{string vazio!}$
- $[] + \{\} = [\text{object Object}]?!$
- $\{\} + [] = 0!!!$
- $\{\} + \{\} = \text{NaN!}$
- $\{\} - \{\} = \text{NaN!}$
- $\{\} * \{\} =$



# JavaScript é um padrão de fato, mas...

- $[] + [] = \text{string vazio!}$
- $[] + \{\} = [\text{object Object}]?!$
- $\{\} + [] = 0!!!$
- $\{\} + \{\} = \text{NaN!}$
- $\{\} - \{\} = \text{NaN!}$
- $\{\} * \{\} = \text{syntax error!}$
- $\text{Array}(16) + 5 =$

# JavaScript é um padrão de fato, mas...

- $[] + [] = \text{string vazio!}$
- $[] + \{\} = [\text{object Object}]?!$
- $\{\} + [] = 0!!!$
- $\{\} + \{\} = \text{NaN!}$
- $\{\} - \{\} = \text{NaN!}$
- $\{\} * \{\} = \text{syntax error!}$
- $\text{Array}(16) + 5 = \text{,,,,,,,,,,,,,5}$
- $\text{Array}(16) - 5 =$

# JavaScript é um padrão de fato, mas...

- $[] + [] = \text{string vazio!}$
- $[] + \{\} = [\text{object Object}]?!$
- $\{\} + [] = 0!!!$
- $\{\} + \{\} = \text{NaN!}$
- $\{\} - \{\} = \text{NaN!}$
- $\{\} * \{\} = \text{syntax error!}$
- $\text{Array}(16) + 5 = \text{,,,,,,,,,,,,,5}$
- $\text{Array}(16) - 5 = \text{NaN!}$

## E o escopo de variáveis?

*“Javascript programmers are practically ranked by how well they understand scope.”*

<http://stackoverflow.com/questions/500431/javascript-variable-scope>

- Melhorou com a introdução de **let**.
- Cuidado com o significado de **this**!



# Linguagens 4



# Linguagens 4



**DART**

e

TypeScript





# DART

e

# TypeScript



## Porquês

- Menos caos
  - Possibilidade de usar **anotações de tipo**
  - Compiláveis para JavaScript
  - Ambas de código aberto
  - atributos adicionados dinamicamente mais controlados

# Alternativas





## 5. Software de Baixo Nível e de Alto Desempenho



Que linguagem você acha que  
seria boa para construir um  
*shooter* 3d?



E um escalonador de processos?



# Um programa para processar imagens de satélites?



# E um driver de dispositivo?

# Sendo um pouco mais focado

Tem algo que nenhuma linguagem discutida anteriormente lhe dá  
**Controle** sobre a máquina



# Sendo um pouco mais focado

Tem algo que nenhuma linguagem discutida anteriormente lhe dá  
**Controle** sobre a máquina

Que linguagem usar se eu quiser...

- acesso direto à RAM
  - inclusive para gerenciar memória manualmente
- guardar valores explicitamente em registradores
  - em alguns casos, especificando o registrador
- produzir código extremamente eficiente
- e (possivelmente) portátil?

# Linguagem 5





# Linguagem 5



# Linguagem 5



THE

C



PROGRAMMING  
LANGUAGE

BRIAN W. KERNIGHAN  
DENNIS M. RITCHIE





THE  
C  
PROGRAMMING  
LANGUAGE

BRIAN W. KERNIGHAN  
DENNIS M. RITCHIE

## Porquês

- Porque permite **tudo mencionado no slide anterior**
- Porque é **menos complexa** que C++
- Porque Assembly só é uma opção quando é a **única opção**

# Mas há outras razões para aprender C

## Outros porquês

- Gerenciamento de memória manual
- Tipificação fraca
- Manipulação de ponteiros é
  - fundamental mesmo para as operações mais básicas
  - complexa
  - insegura
- Difícil para novatos

**Resumindo:** é importante aprender C porque:



**Resumindo:** é importante aprender C porque:

- É uma linguagem muito usada na prática



**Resumindo:** é importante aprender C porque:

- É uma linguagem **muito usada na prática**
- Para vários tipos de aplicação, é uma **boa ferramenta**



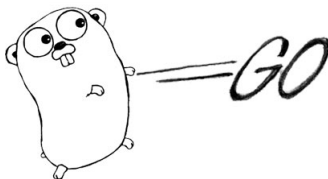
**Resumindo:** é importante aprender C porque:

- É uma linguagem **muito usada na prática**
- Para vários tipos de aplicação, é uma **boa ferramenta**
- As **características negativas** da linguagem fornecem lições importantes





# Alternativas



E sempre haverá **C++**, claro. E Assembly.





## Linguagens

- 1 Python
- 2 Swift
- 3 Erlang
- 4 Dart e TypeScript
- 5 C



## Linguagens

- 1 Python
- 2 Swift
- 3 Erlang
- 4 Dart e TypeScript
- 5 C

**Contato:** castor no domínio cin.ufpe.br

**Slides:** <http://github.com/fernandocastor>

**Vídeo:** <http://www.youtube.com/watch?v=MsWmSCCVj5Y>

(Busca no YouTube ou Google: **cpre1 linguagens**)