



Grupo: Bruno Luiz, Gabriel Gomes, Luiz Guimarães, Luiza Ávila, Pedro Achilles.

Professor: Marco Rodrigo Costa

Nome Swift



História

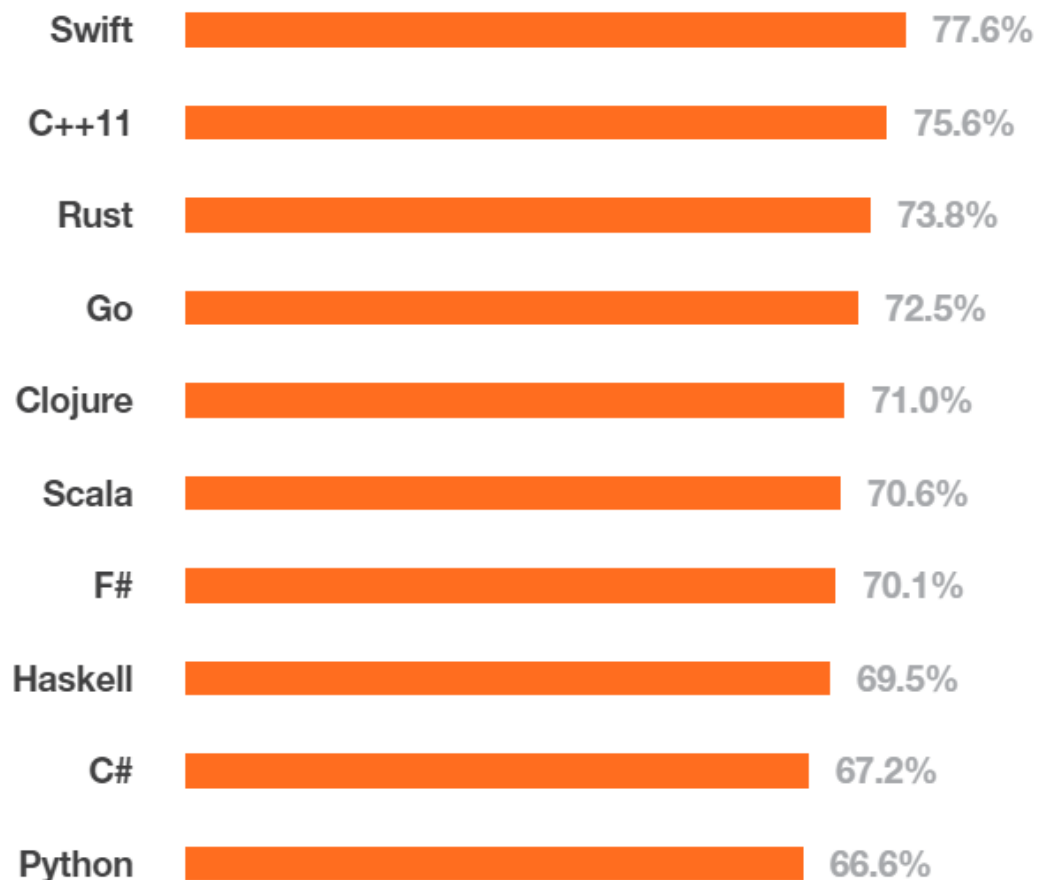
- Desenvolvimento inicializado em Julho de 2010, por Chris Lattner.
- Disponibilizada aos desenvolvedores em 2014.
- Ganhou primeiro lugar na categoria *Linguagem de Programação Mais Amada em 2015*.
- Torna-se open source em 3 de Dezembro de 2015.
- É lançado Swift Playgrounds em 2016.



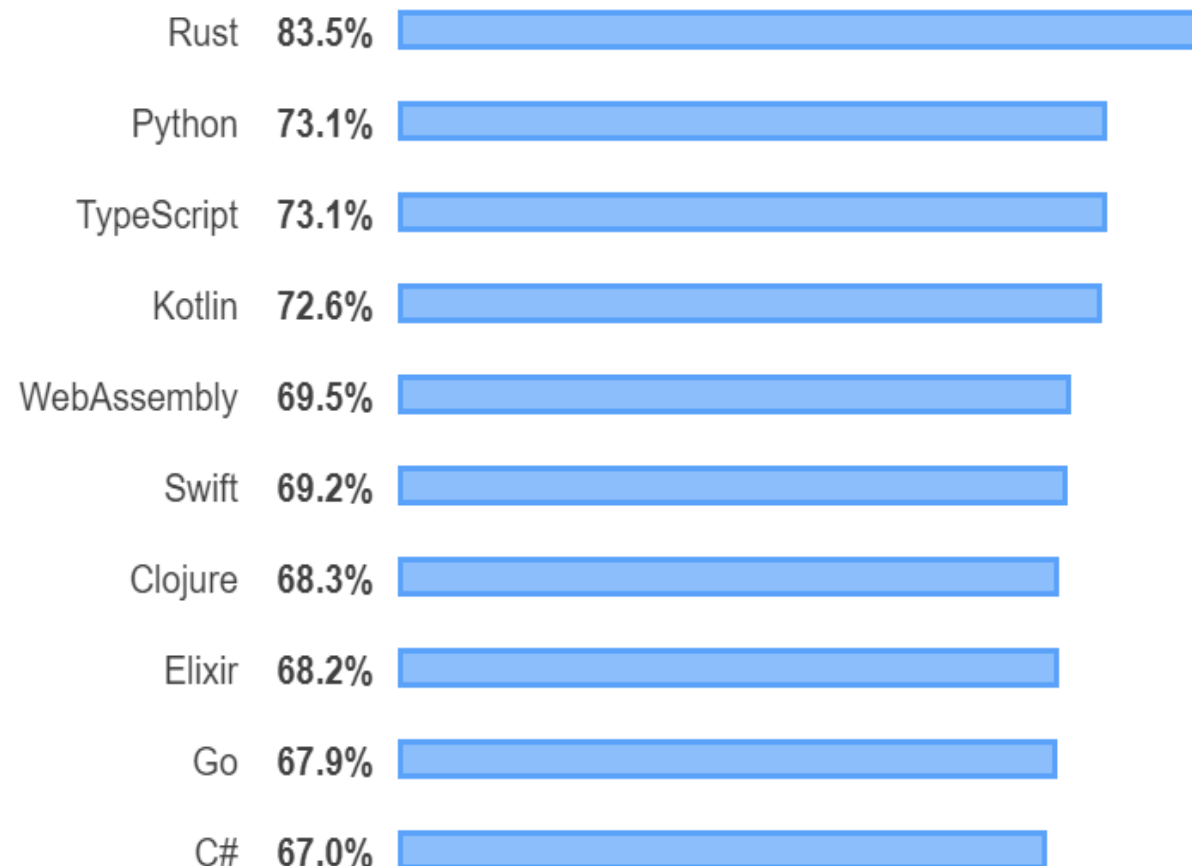
Histórico de Versões

Data de Lançamento	Versão
09/09/2014	Swift 1.0
22/10/2014	Swift 1.1
08/04/2015	Swift 1.2
21/09/2015	Swift 2.0
13/09/2016	Swift 3.0
19/09/2017	Swift 4.0
29/03/2018	Swift 4.1
17/09/2018	Swift 4.2
25/03/2019	Swift 5.0

Linguagem de Programação Mais Amada



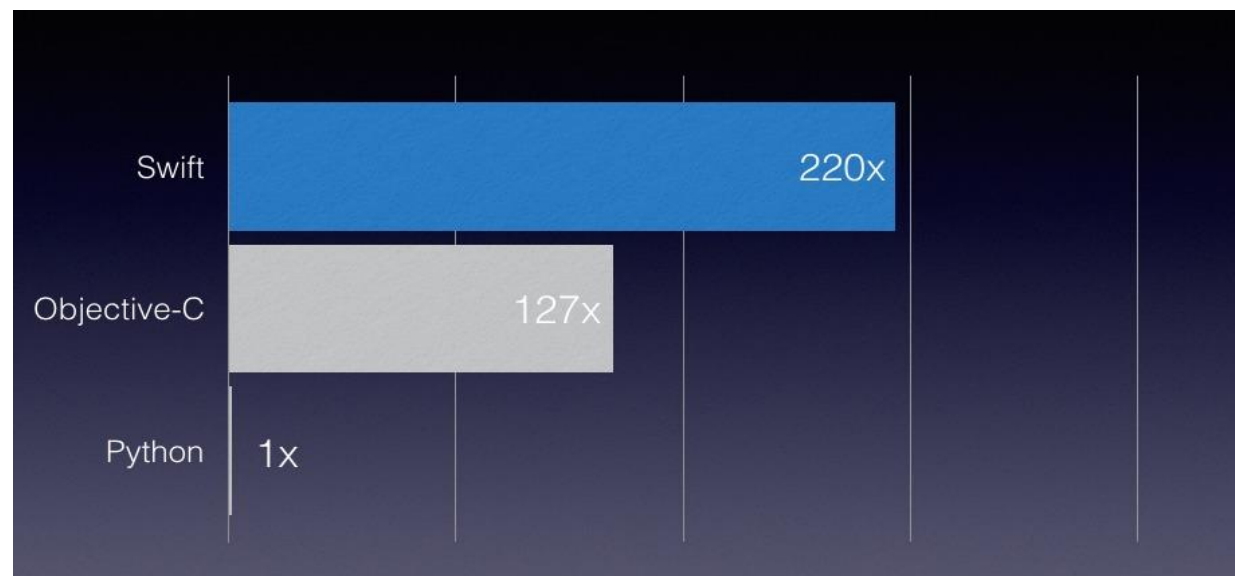
Lista de linguagens mais amadas 2015



Lista de linguagens mais amadas 2019

Características

- Moderna.
- Rápida.
- Segura.
- interativa
- Sintaxe simples.
- Fácil aprendizado.
- Tipagem Estática e forte.



Características gerais

- A declaração de variáveis em Swift é feito por meio dos marcadores *var* e *let*. Usamos *var* para declarar uma variável, e *let* para declarar uma constante.

```
var linguagem: String = "Swift"  
let materia = "LP"
```

- Funções em Swift são declaradas pela palavra-chave *func*.
- Por padrão, todos os parâmetros reais de uma função devem ser rotulados. Podemos contornar isso usando um traço baixo antes da amarração do parâmetro formal.

```
func pow(_ base: Double, exp: Double = 2.0) -> Double {  
    if exp == 0 {  
        return 1  
    } else {  
        return base * pow(base, exp: exp-1);  
    }  
}
```

Optionals

- Em Swift, tipos que podem não conter algum valor (se anulam), recebem um tratamento especial, são chamados opcionais.
- Um tipo opcional não pode ser usado onde se espera uma variável não-nula, sendo assim devemos tratar esses casos.

```
var opt: String? = nil

print(opt) // nil
print(opt!) // ERRO de execução
print(opt ?? "Valor Nulo") // "Valor Nulo"
```

Closures e Generics

- Através do uso de generics, podemos receber em nossas classes e funções tipos desconhecidos. Assim podemos escrever um unico procedimento capaz de operar sobre vários tipos diferentes.
- Com o uso das closures, podes passar funções como parâmetro para outras funções de forma a modificar seu comportamento.

```
func maior<T>(_ lista: [T], f: (T, T) -> Bool) -> T {  
    var maior = lista[0];  
    for i in 1..  
        <lista.count {  
        if f(maior, lista[i]) {  
            maior = lista[i]  
        }  
    }  
  
    return maior  
}
```


Multiparadigma

**Orientada a
objetos**

Estruturada

**Orientada a
protocolos**

Funcional

Imperativa

Protocolos

```
protocol Vegetal {  
    var especie: String { get }  
    var comeCarne: Bool { get }  
}  
  
protocol Carnivoro {  
    var kilogramasNecessariosPorDia: Double { get }  
}  
  
extension Vegetal {  
    var comeCarne: Bool { return self is Carnivoro }  
}
```

```
struct Ventricosa: Vegetal, Carnivoro {  
    let especie: String = "Nepenthes Ventricosa"  
  
    let kilogramasNecessariosPorDia: Double = 0.02  
}  
  
struct OrquideaQualquer: Vegetal {  
    let especie: String = "Cattleya Labiata"  
}
```

```
enum DroseraSulAmericana: String, Vegetal, Carnivoro {  
    case tentaculata = "tentaculata"  
    case sessilifolia = "sessilifolia"  
    case roraimae = "roraimae"  
    case desconhecido  
  
    var especie: String {  
        return (self != .desconhecido) ? "Drosera \(self.rawValue)" : "Desconhecido"  
    }  
  
    var kilogramasNecessariosPorDia: Double { return 0.03 }  
}
```

Switch Case

Fonte: IOS App Development

```
switch (some value to consider) {  
  case value 1 :  
    respond to value 1  
  case value 2 ,  
    value 3 :  
    respond to value 2 or 3  
  default:  
    otherwise, do something else  
}
```

Relacionamento com outras linguagens

Linguagens semelhantes

- Objective-C, C++, Ruby, Haskell, Python, Rust e muitas outras.
 - Ruby e Python por usar poucas linhas de programação
 - C++, Java e C# pelo uso de Generics
 - C# por tipos Optionals
 - Scala e Opa por inferência de tipos
- É importante também pontuar que a linguagem com que mais se assemelha é aquela que deseja substituir: Objective-C.

Linguagens contraditórias

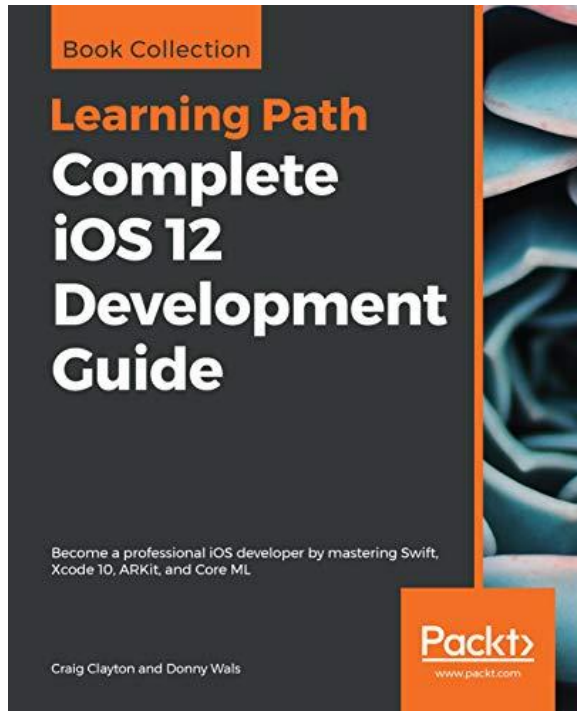
- Objective-C pois Swift preenche as lacunas e falhas de Objective-C
- Python por uma conseguir preencher a alternativa pra lacuna da outra.

```
137 //access camera//access camera//access camera//access camera//
138
139 color1Button.isHidden=true //hide colors
140 ///processing review
141 if((userDefaults.object(forKey: "firstrun") == nil))
142 {
143     userDefaults.set(true, forKey: "firstrun")
144     debugPrint("FIRST RUN")
145 }
146 else // if not first run
147 {
148     if(userDefaults.bool(forKey: "firstrun") == true)
149     {
150         userDefaults.set(
151
```

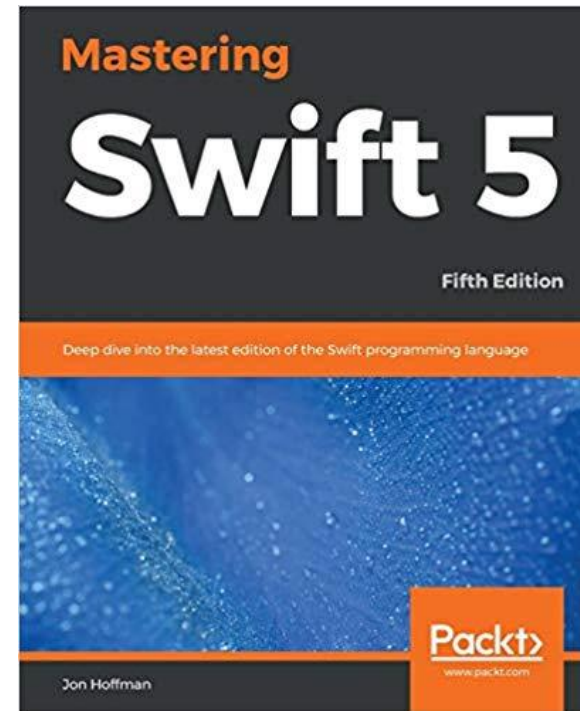
Exemplos

```
main.asyncAfter(deadline: now)
self?.color1Butt
let
```

Bibliografia



CLAITON, Craig e WALS, Donny. *Complete iOS 12 Development Guide: Become a professional iOS developer by mastering Swift, Xcode 10, ARKit, and Core ML.*

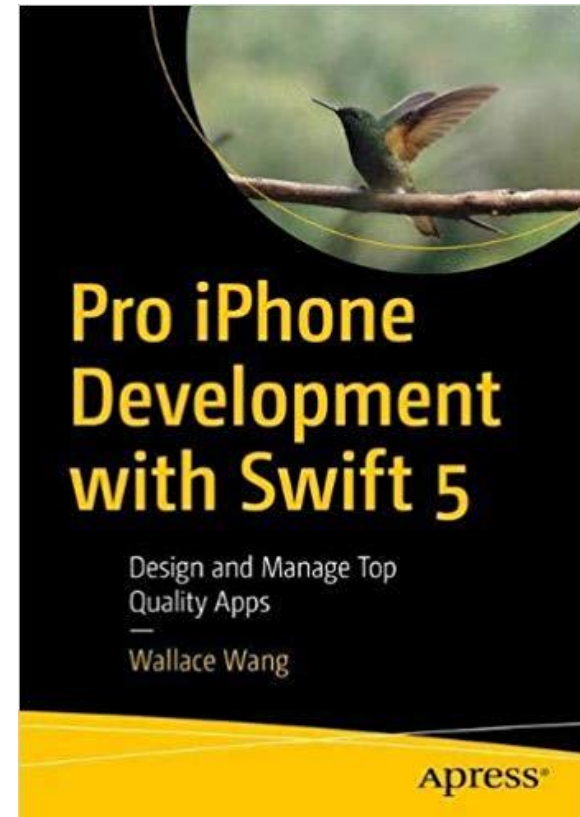


HOFFMAN, Jon. *Mastering Swift 5 (5a Edição)*

Bibliografia



Apple Inc. *The Swift Programming Language (Swift 5 Edition)*. Disponível em:
<<https://swift.org/documentation/>>



WANG, Wallace. *Pro iPhone Development with Swift 5: Design and Manage Top Quality Apps*.

Referência

STACK OVERFLOW. **2015 Developer Survey**. Disponível em: <<https://insights.stackoverflow.com/survey/2015>> Acesso em 28 de maio de 2019

STACK OVERFLOW. **Developer Survey Results 2019**. Disponível em: <<https://insights.stackoverflow.com/survey/2019>> Acesso em 28 de maio de 2019

EDUCATION ECOSYSTEM. **Swift History**. Disponível em: <<https://www.education-ecosystem.com/guides/programming/swift/history>> Acesso em 28 de maio de 2019

CLEVERISM. **Swift**. Disponível em: <<https://www.cleverism.com/skills-and-tools/swift/>> Acesso em 28 de maio de 2019

DATTA, Sourav. **Which programming language is Swift similar to, and how do they compare to each other?** 14 jun. 2014. Disponível em: <<https://www.quora.com/Which-programming-language-is-Swift-similar-to-and-how-do-they-compare-to-each-other>> Acesso em 28 de maio de 2019

RUBENS, Paul. **10 things you should know about Apple's Swift**. 21 jul. 2014 Disponível em: <<https://www.javaworld.com/article/2456964/10-things-you-should-know-about-apples-swift.html>> Acesso em 28 de maio de 2019

THIAGO. **Desenvolvimento IOS: conheça a linguagem swift**. 2014. Disponível em: <<https://www.devmedia.com.br/desenvolvimento-ios-conheca-a-linguagem-swift/31860>> Acesso em 28 de maio de 2019

ALTExSOFT. **Swift vs Objective-C: Out with the Old, In with the New**. 28 jun. 2018. Disponível em: <<https://www.altexsoft.com/blog/engineering/swift-vs-objective-c-out-with-the-old-in-with-the-new/>> Acesso em 28 de maio de 2019

POPKO, Aleksander. **Swift vs Objective C: iOS' Programming Languages Compared**. 14 Feb.2018. Disponível em: <<https://www.netguru.com/blog/swift-vs-objective-c-ios-programming-languages-compared>> Acesso em 28 de maio de 2019

REDBYTES. **Importance of the Programming Language ‘Swift’ in iOS Development.** 21 mar. 2017. Disponível em: <<https://www.redbytes.in/importance-programming-language-swift-mobile-app-development/>> Acesso em 28 de maio de 2019

PATIL, Ritesh. **Similarities And Differences Between Objective-C And Swift Programming.** 1 Mar. 2017. Disponível em:<<https://medium.com/@ritesh.patil732/similarities-and-differences-between-objective-c-and-swift-programming-ab15d215b136>> Acesso em 28 de maio de 2019

JAGTAP, Shashikant. **Mobile Developers Takeaways From Stack Overflow Developer Survey 2019.** 11 abr. 2019. Disponível em: <<https://medium.com/xcblog/mobile-developers-takeaways-from-stack-overflow-developer-survey-2019-c0d04f6eb902>> Acesso em 28 de maio de 2019

MEDIUM. **The Evolution of Swift.** Disponível em:
<<https://medium.com/@mindfiresolutions.usa/the-evolution-of-swift-29e7a89f1a0c>> Acesso em 28 de maio de 2019

WIKIPEDIA. **Swift (linguagem de programação).** Disponível em:
<[https://pt.wikipedia.org/wiki/Swift_\(linguagem_de_programa%C3%A7%C3%A3o\)](https://pt.wikipedia.org/wiki/Swift_(linguagem_de_programa%C3%A7%C3%A3o))> Acesso em 28 de maio de 2019

TUTSPLUS. **Swift from Scratch: Parâmetros, tipos e encadeamento de funções.** Disponível em:
<<https://code.tutsplus.com/pt/tutorials/swift-from-scratch-function-parameters-types-and-nesting--cms-23056>> Acesso em 28 de maio de 2019

IMASTERS. **Falando sobre Swift – Funções.** Disponível em:
<<https://imasters.com.br/back-end/falando-sobre-swift-funcoes>> Acesso em 28 de maio de 2019

ARSTECHNICA. **A fast look at Swift, Apple’s new programming language.** Disponível em:
<<https://arstechnica.com/gadgets/2014/06/a-fast-look-at-swift-apples-new-programming-language/>> Acesso em 28 de maio de 2019

TECNOBLOG. **Surpresa na WWDC: Apple anuncia Swift, sua nova linguagem de programação.** Disponível em:
<<https://tecnoblog.net/157711/apple-linguagem-programacao-swift/>>

GITHUB. Disponível em: <<https://github.com/raywenderlich/swift-algorithm-club/tree/master/Quicksort>> Acesso em 28 de maio de 2019