



UNIVERSIDADE
FEDERAL DO CEARÁ
CAMPUS QUIXADÁ

Kotlin - Fundamentos Básicos

QXD0102 - Desenvolvimento para Dispositivos Móveis

Prof. Bruno Góis Mateus (brunomateus@ufc.br)

Conteúdo

Conteúdo

- Introdução

Conteúdo

- Introdução
- História

Conteúdo

- Introdução
- História
- Características

Conteúdo

- Introdução
- História
- Características
- Kotlin e Android

Conteúdo

- Introdução
- História
- Características
- Kotlin e Android
- Hands on

Conteúdo

- Introdução
- História
- Características
- Kotlin e Android
- Hands on
- Tarefa de casa

Introdução

**Quantas linguagens de
programação existem ?**

The aim of this list of programming languages is to include all notable programming languages in existence, both those in current use and historical ones, in alphabetical order. *Dialects of BASIC, esoteric programming languages, and markup languages* are not included.

[D](#) · [S](#) · [A](#) · [B](#) · [C](#) · [D](#) · [E](#) · [F](#) · [G](#) · [H](#) · [I](#) · [J](#) · [K](#) · [L](#) · [M](#) · [N](#) · [O](#) · [P](#) · [Q](#) · [R](#) · [S](#) · [T](#) · [U](#) · [V](#) · [W](#) · [X](#) · [Y](#) · [Z](#)
Contents: See also

Programming language lists

- [Alphabetical](#)
- [Categorical](#)
- [Chronological](#)
- [Generational](#)

1119

A [\[edit \]](#)

- | | | | |
|--|---|---|---|
| <ul style="list-style-type: none">• A.NET• A-0 System• A+• ABAP• ABC• ABC ALGOL• ACC• Accent• Ace DASL (Distributed Application Specification Language)• Action!• ActionScript | <ul style="list-style-type: none">• Actor• Ada• Adenine• Agda• Agilent VEE• Agora• AIMMS• Aldor• Alef• ALF• ALGOL 58• ALGOL 60 | <ul style="list-style-type: none">• ALGOL 68• ALGOL W• Alice• Alma-0• AmbientTalk• Amiga E• AMOS• AMPL• AngelScript• Apex• APL• App Inventor for Android's visual block language | <ul style="list-style-type: none">• AppleScript• APT• Arc• ARexx• Argus• Assembly language• AutoHotkey• AutoIt• AutoLISP / Visual LISP• Avenest• AWK• Axum |
|--|---|---|---|

B [\[edit \]](#)

- | | | | |
|--|---|--|--|
| <ul style="list-style-type: none">• B• Babbage• Ballerina• Bash• BASIC | <ul style="list-style-type: none">• Batch file (Windows/MS-DOS)• bc• BCPL• BeanShell | <ul style="list-style-type: none">• Bertrand• BETA• BLISS• Blocky• BlooP | <ul style="list-style-type: none">• Boo• Boomerang• Bosque• Bourne shell (including bash and ksh) |
|--|---|--|--|

C [\[edit \]](#)

- | | | | |
|--|--|---|---|
| <ul style="list-style-type: none">• C• C- (C minus minus)• C++ (C plus plus) – ISO/IEC 14882• C*• C# (C sharp) – ISO/IEC 23270 | <ul style="list-style-type: none">• Chicken• Chapel• Chem• CHILL• CHIP-8 | <ul style="list-style-type: none">• CLU• CMS-2• COBOL – ISO/IEC 1989• CobolScript – COBOL Scripting language• Cobra | <ul style="list-style-type: none">• Coral 66• CorVision• COWSEL• CPL• Cryptol |
|--|--|---|---|

The aim of this list of programming languages is to include all notable programming languages in existence, both those in current use and historical ones, in alphabetical order. *Dialects* of BASIC, esoteric programming languages, and markup languages are not included.

[D](#) · [S](#) · [A](#) · [B](#) · [C](#) · [D](#) · [E](#) · [F](#) · [G](#) · [H](#) · [I](#) · [J](#) · [K](#) · [L](#) · [M](#) · [N](#) · [O](#) · [P](#) · [Q](#) · [R](#) · [S](#) · [T](#) · [U](#) · [V](#) · [W](#) · [X](#) · [Y](#) · [Z](#)
Contents: See also

Programming language lists

- [Alphabetical](#)
- [Categorical](#)
- [Chronological](#)
- [Generational](#)

1119

A [\[edit \]](#)

- | | | | |
|--|---|---|---|
| <ul style="list-style-type: none">• A.NET• A-0 System• A+• ABAP• ABC• ABC ALGOL• ACC• Accent• Ace DASL (Distributed Application Specification Language)• Action!• ActionScript | <ul style="list-style-type: none">• Actor• Ada• Adenine• Agda• Agilent VEE• Agora• AIMMS• Aldor• Alef• ALF• ALGOL 58• ALGOL 60 | <ul style="list-style-type: none">• ALGOL 68• ALGOL W• Alice• Alma-0• AmbientTalk• Amiga E• AMOS• AMPL• AngelScript• Apex• APL• App Inventor for Android's visual block language | <ul style="list-style-type: none">• AppleScript• APT• Arc• ARexx• Argus• Assembly language• AutoHotkey• AutoIt• AutoLISP / Visual LISP• Avenest• AWK• Axum |
|--|---|---|---|

B [\[edit \]](#)

- | | | | |
|--|---|--|--|
| <ul style="list-style-type: none">• B• Babbage• Ballerina• Bash• BASIC | <ul style="list-style-type: none">• Batch file (Windows/MS-DOS)• bc• BCPL• BeanShell | <ul style="list-style-type: none">• Bertrand• BETA• BLISS• Blocky• BlooP | <ul style="list-style-type: none">• Boo• Boomerang• Bosque• Bourne shell (including bash and ksh) |
|--|---|--|--|

C [\[edit \]](#)

- | | | | |
|--|--|---|---|
| <ul style="list-style-type: none">• C• C- (C minus minus)• C++ (C plus plus) – ISO/IEC 14882• C*• C# (C sharp) – ISO/IEC 23270 | <ul style="list-style-type: none">• Chicken• Chapel• Chem• CHILL• CHIP-8 | <ul style="list-style-type: none">• CLU• CMS-2• COBOL – ISO/IEC 1989• CobolScript – COBOL Scripting language• Cobra | <ul style="list-style-type: none">• Coral 66• CorVision• COWSEL• CPL• Cryptol |
|--|--|---|---|

Que tal programar em WhiteSpace?

Que tal programar em WhiteSpace?

```
S S S T S S T S S S L
T L
S S S S S T T S S T S T L
T L
S S S S S T T S T T S S L
T L
S S S S S T T S T T S S L
T L
S S S S S T T S T T T T L
T L
S S S S S T S T T S S L
T L
S S S S S T S S S S L
T L
S S S S S T T T S T T T L
T L
S S S S S T T S T T T T L
T L
S S S S S T T T S S T S L
T L
S S S S S T T S T T S S L
T L
S S S S S T S S S S T L
T L
S S L
L
L
```

Que tal programar em WhiteSpace?

```

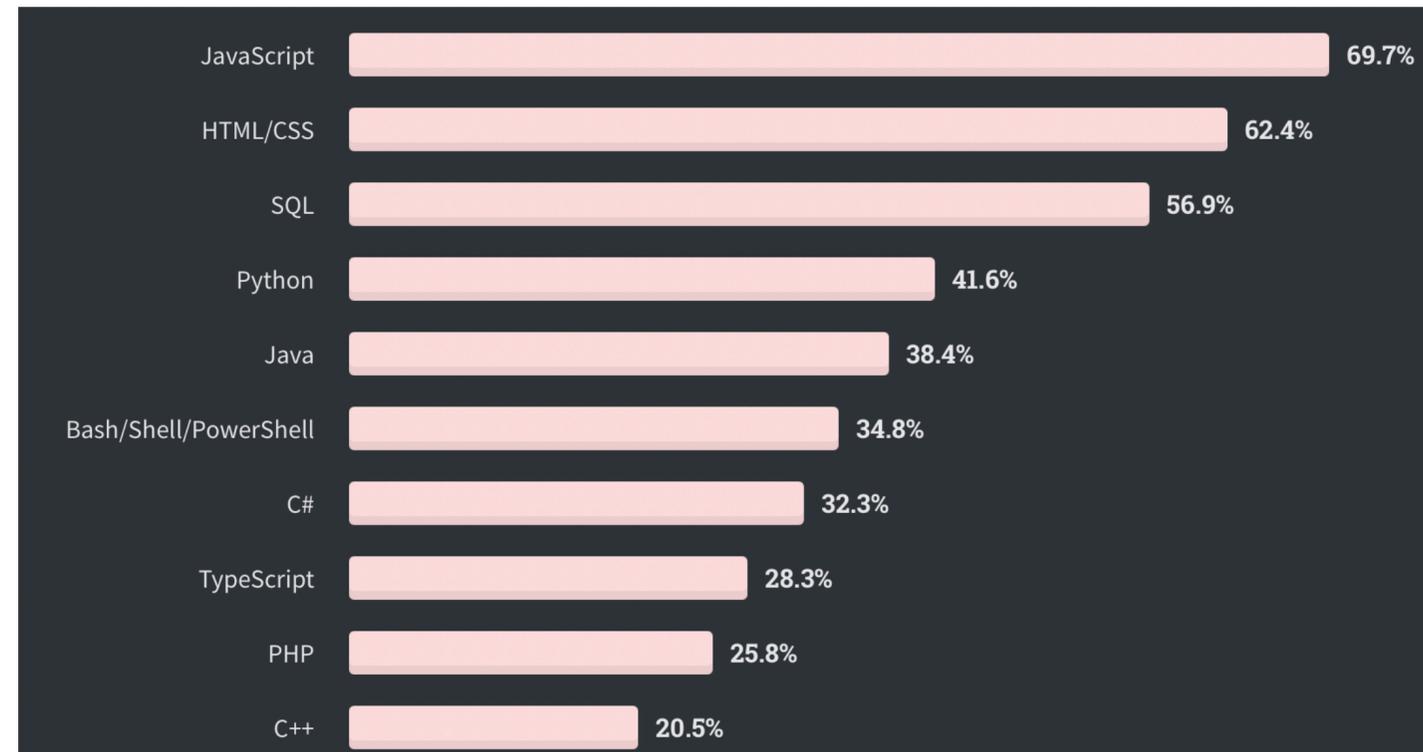
S S S T S S T S S S L
T L
S S S S S T T S S T S T L
T L
S S S S S T T S T T S S L
T L
S S S S S T T S T T S S L
T L
S S S S S T T S T T T T L
T L
S S S S S T S T T S S L
T L
S S S S S T S S S S S L
T L
S S S S S T T T S T T T L
T L
S S S S S T T S T T T T L
T L
S S S S S T T S S T S S L
T L
S S S S S T S S S S T L
T L
S S L
L
L

```



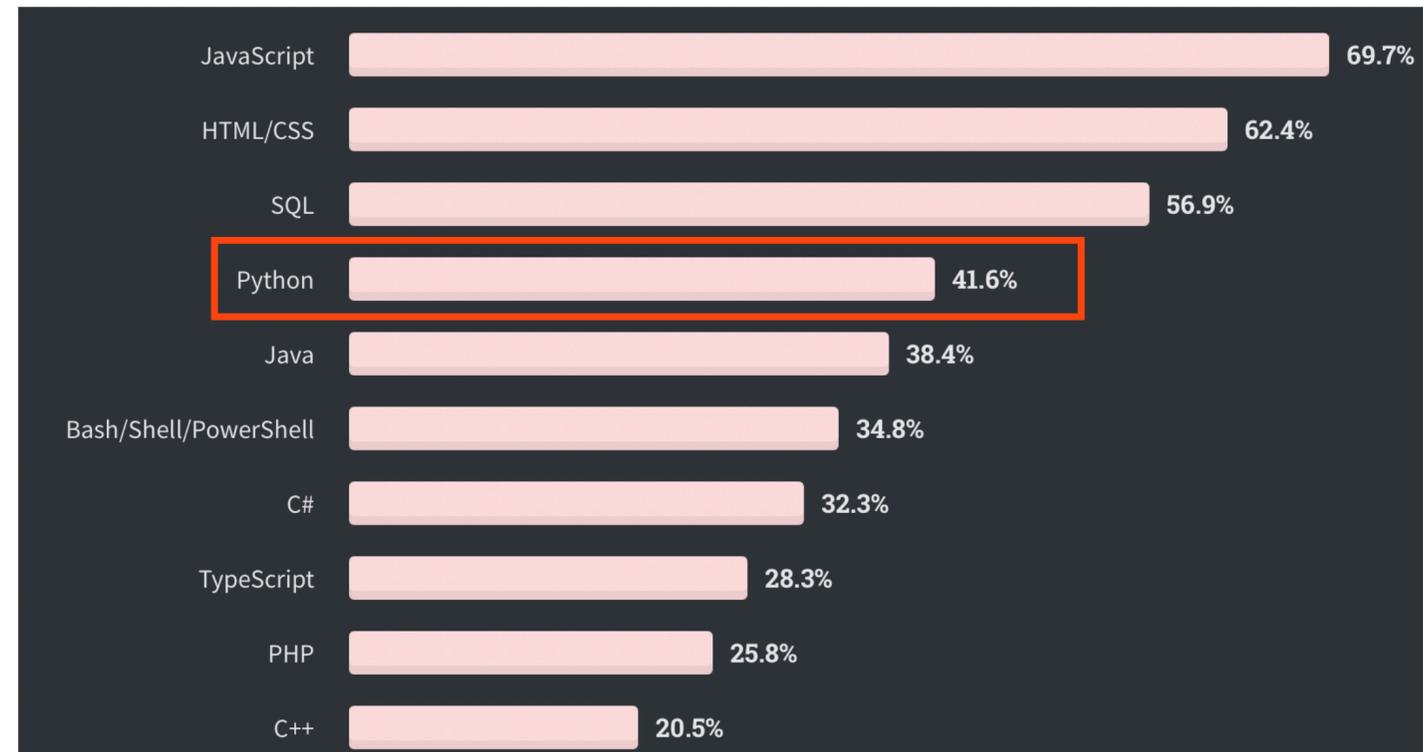
As mais populares

StackOverFlow - 2020 Developer Survey



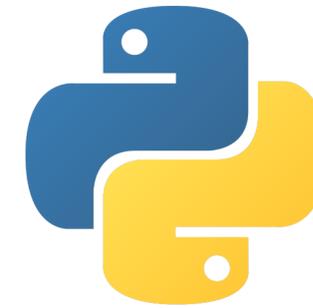
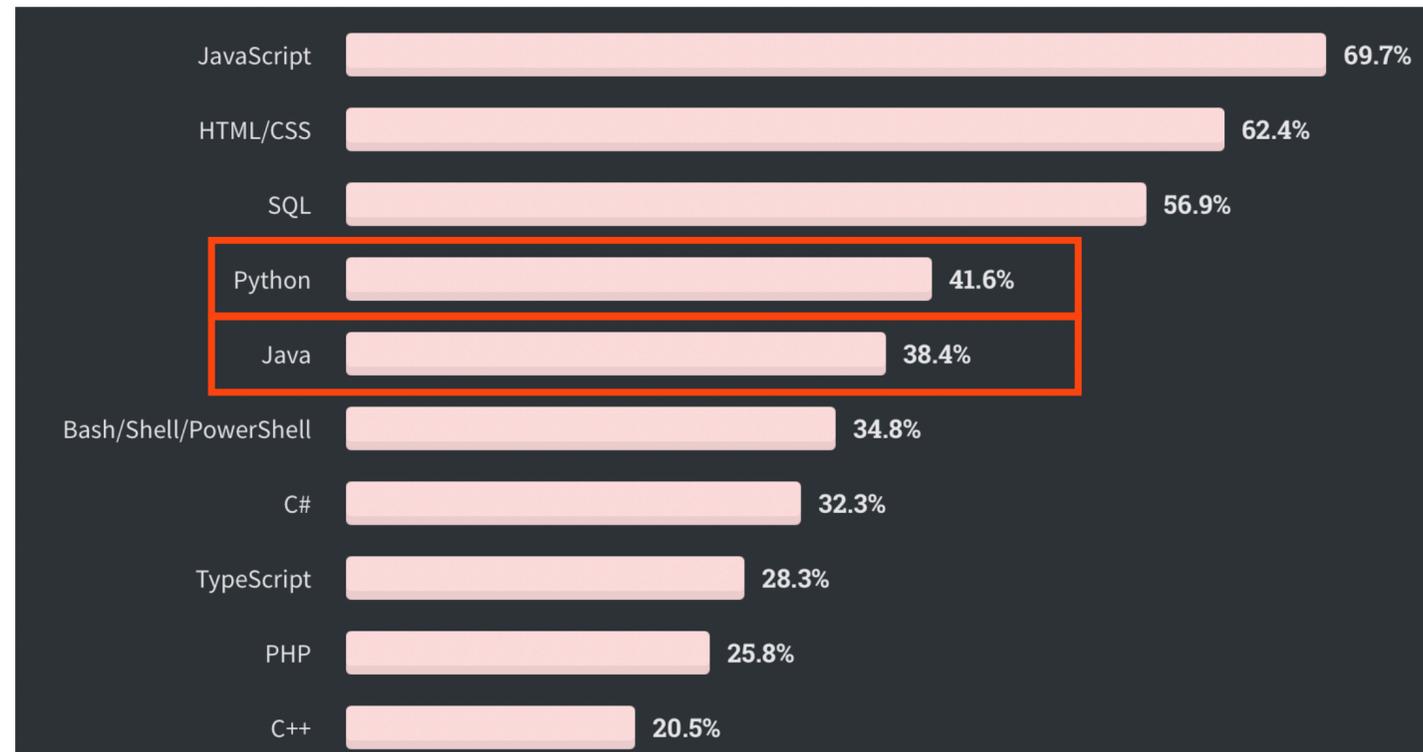
As mais populares

StackOverFlow - 2020 Developer Survey



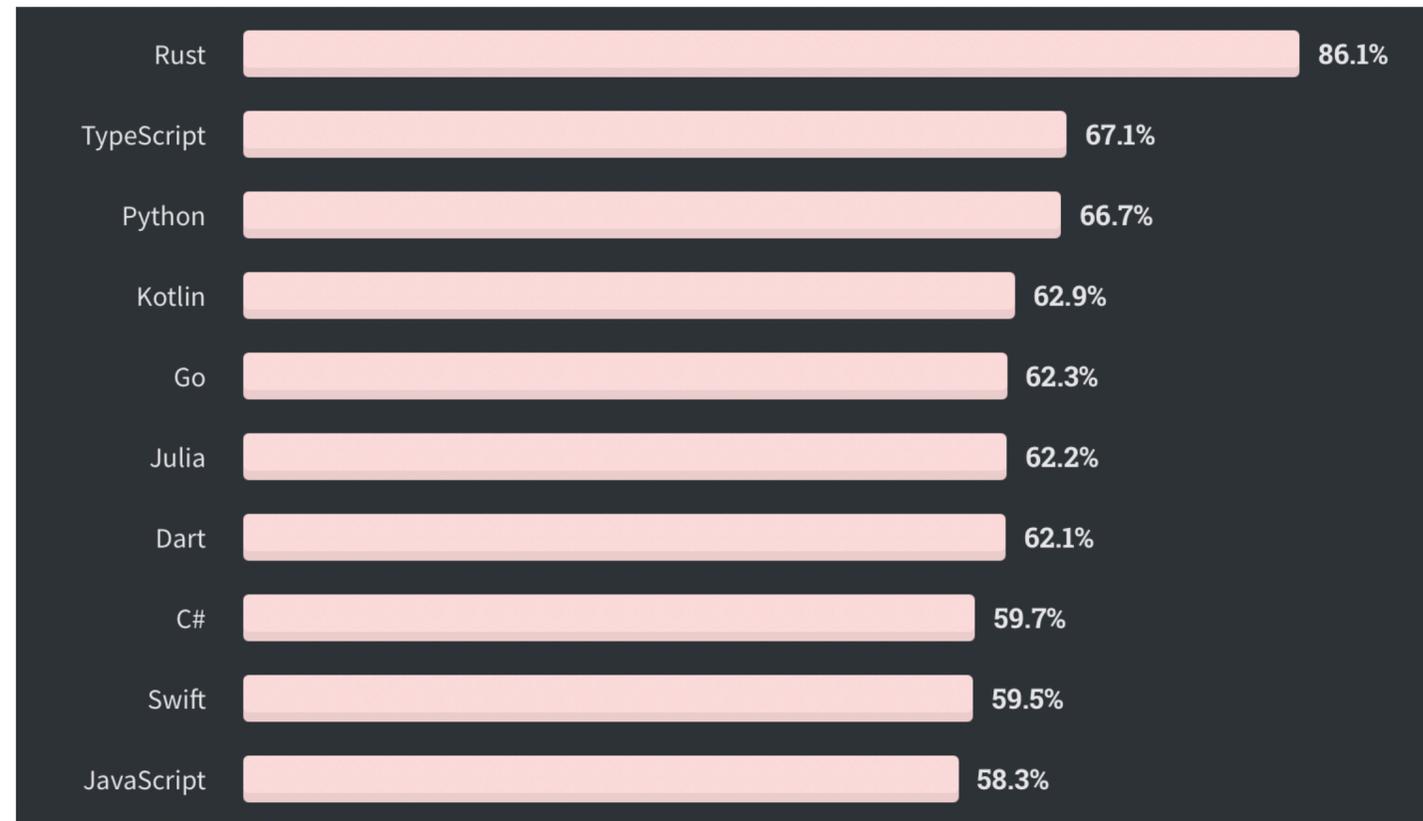
As mais populares

StackOverFlow - 2020 Developer Survey



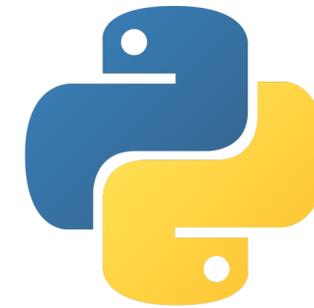
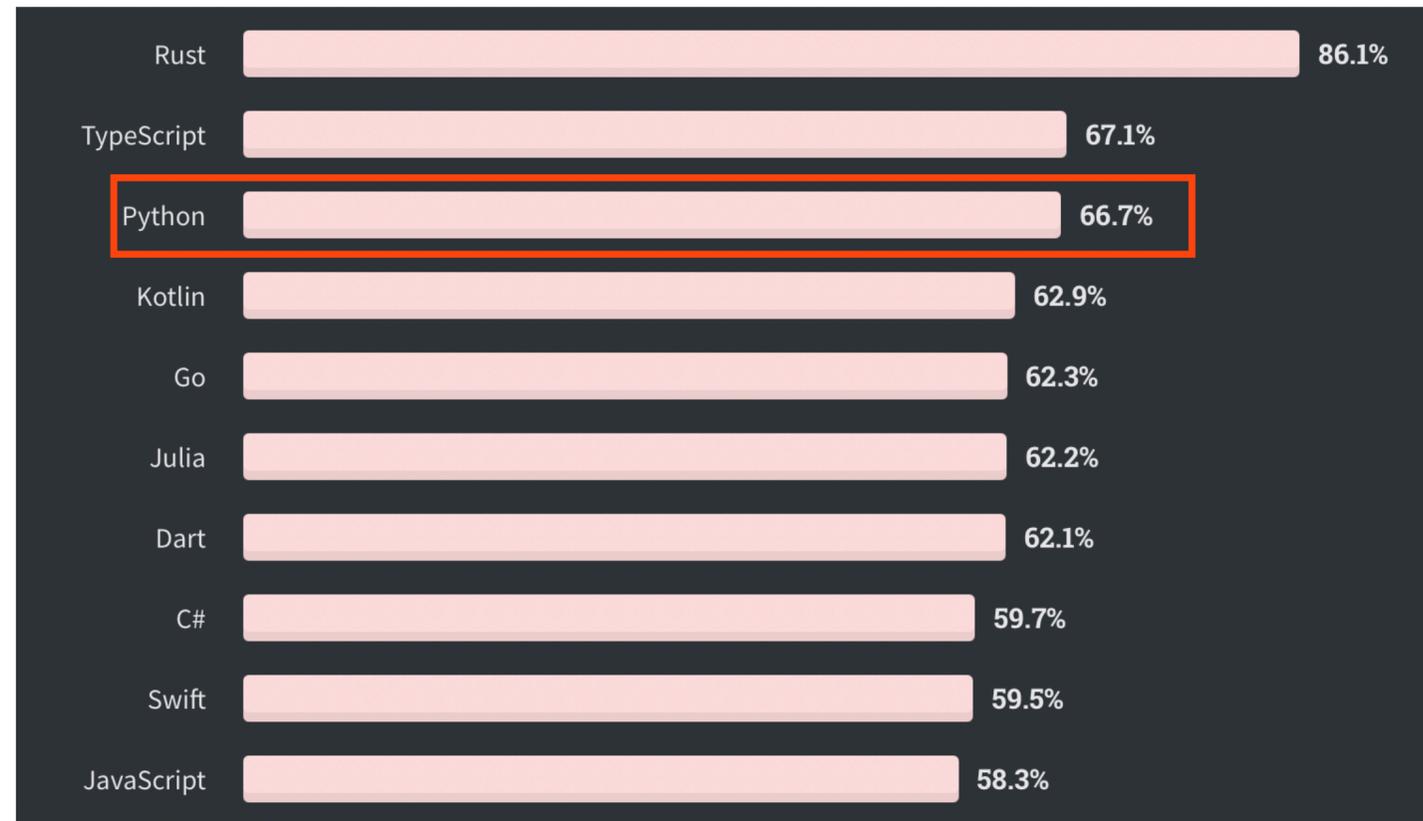
As mais amadas

StackOverFlow - 2020 Developer Survey



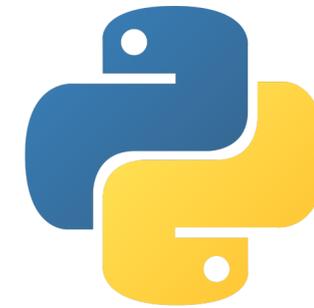
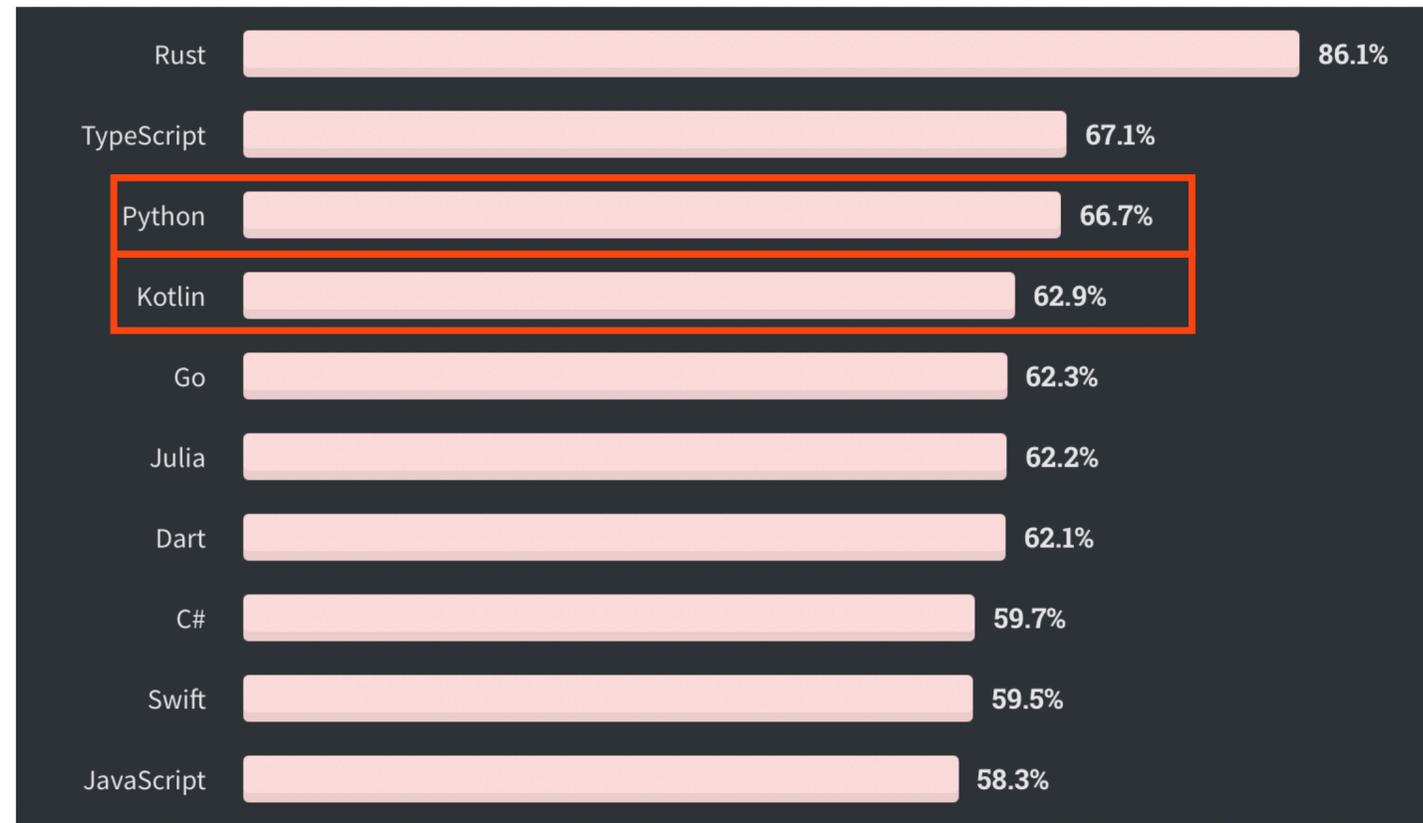
As mais amadas

StackOverFlow - 2020 Developer Survey



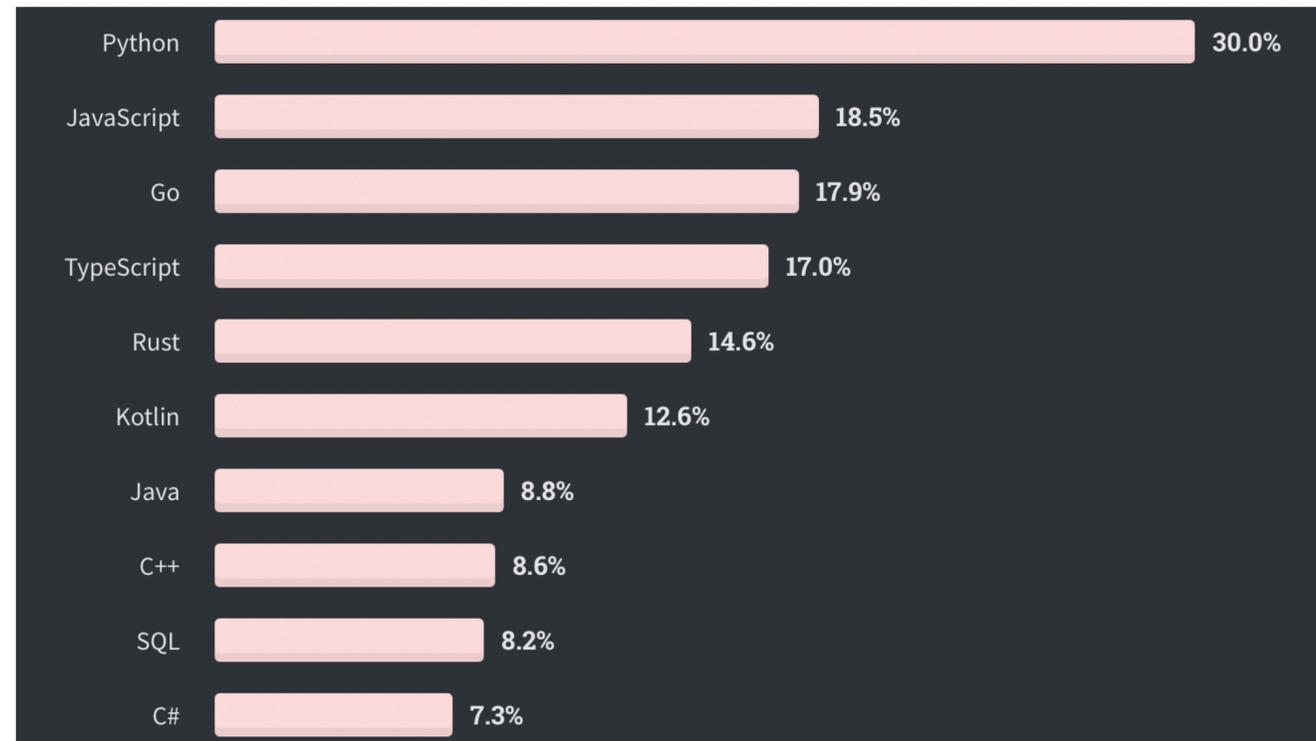
As mais amadas

StackOverFlow - 2020 Developer Survey



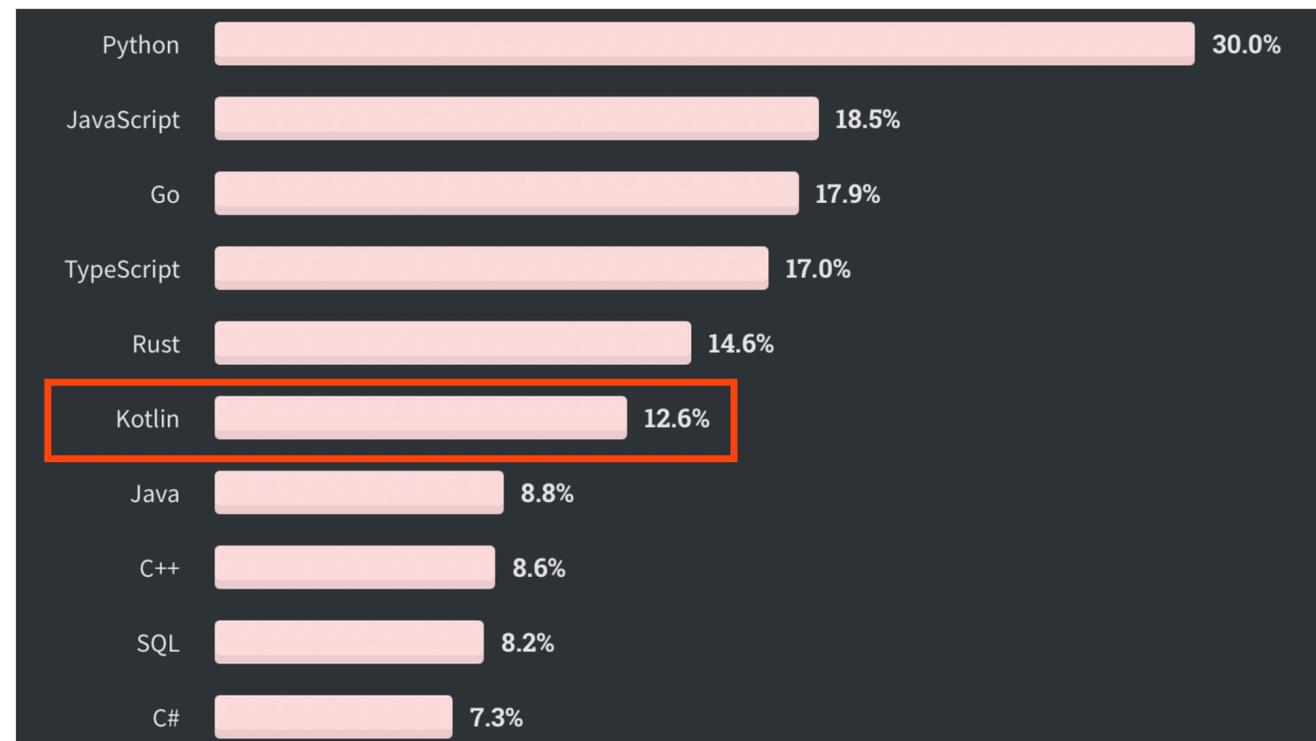
As mais desejadas

StackOverFlow - 2020 Developer Survey



As mais desejadas

StackOverFlow - 2020 Developer Survey



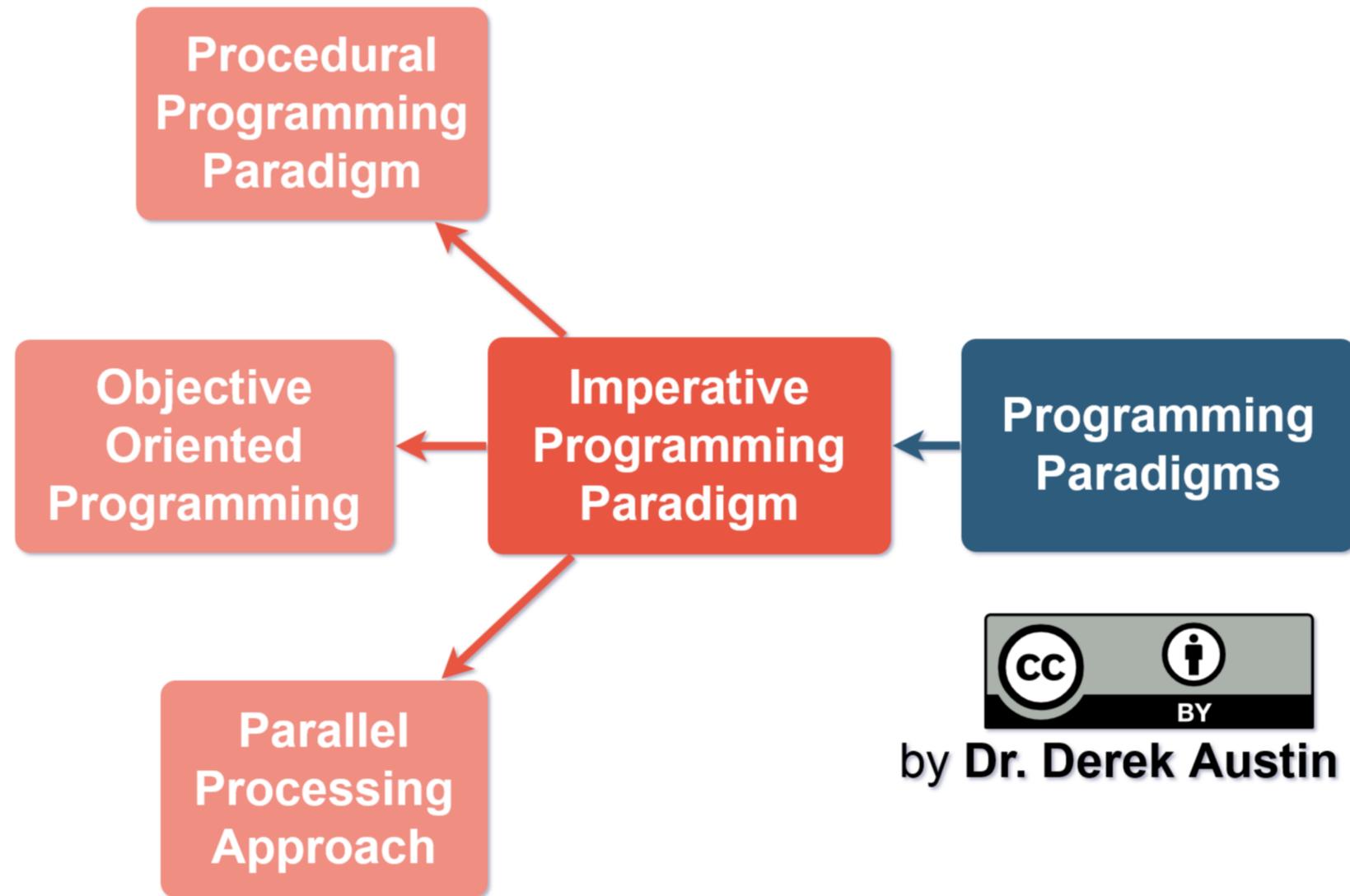
Paradigmas de Programação

Programming
Paradigms

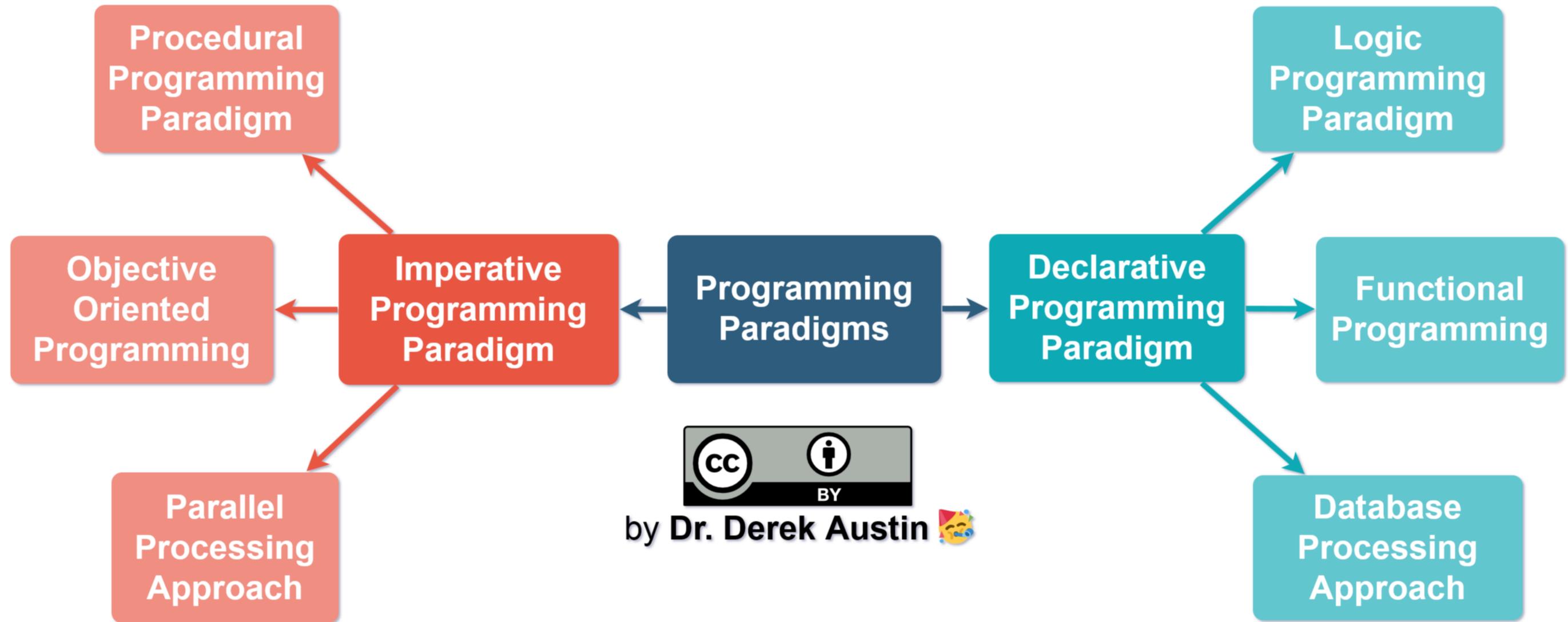


Dr. Derek Austin |

Paradigmas de Programação

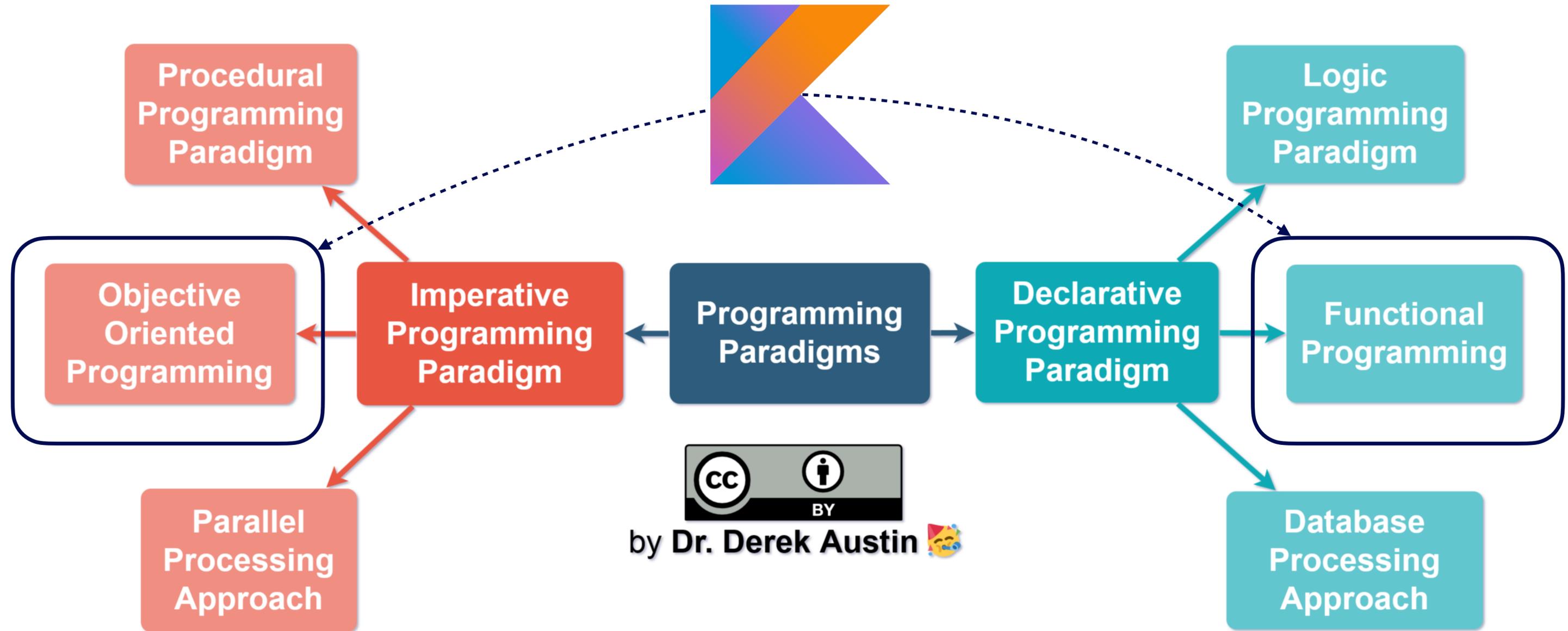


Paradigmas de Programação



by Dr. Derek Austin 🧑🏻‍🎓

Paradigmas de Programação



História

História

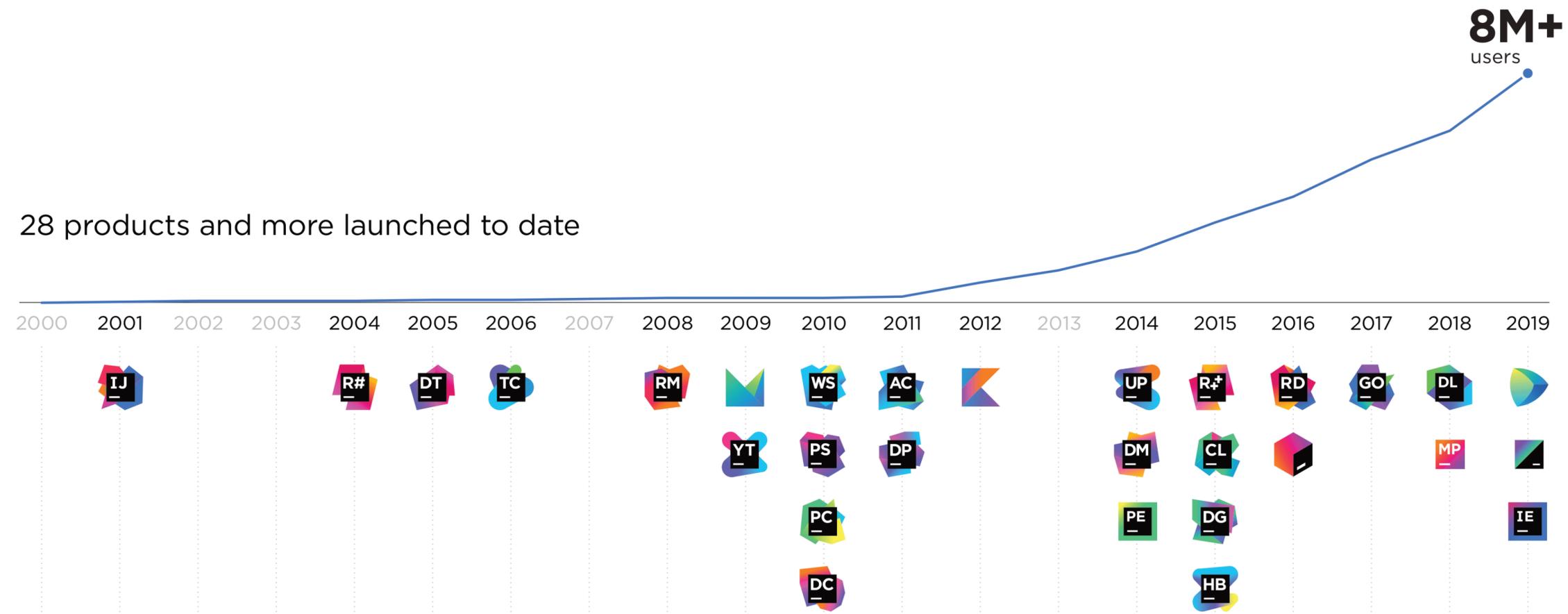
História

- Desenvolvida pela **JetBrains**
 - Empresa Checa



História

- Desenvolvida pela **JetBrains**
- Empresa Checa



História

Tudo começou em 2011

História

Tudo começou em 2011

Andrey Breslav, líder do projeto Kotlin

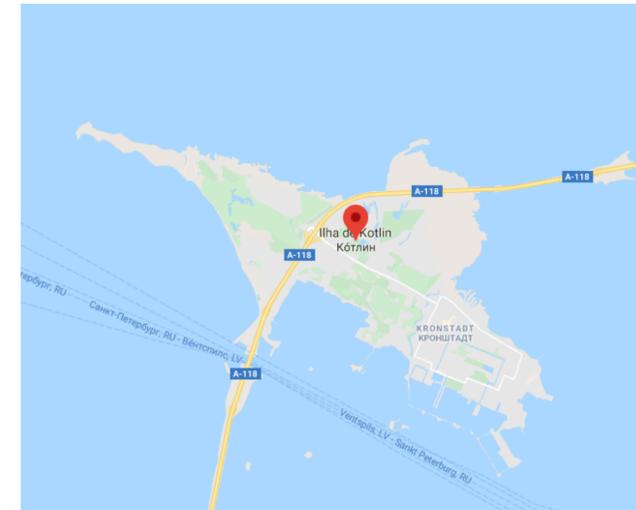
"Kotlin é projetada para ser uma linguagem orientada a objeto de força industrial, **e melhor do que Java**, mas ainda ser totalmente **interoperável** com código Java, permitindo que as empresas possam **fazer uma migração gradual de Java para Kotlin.**"

História

Tudo começou em 2011

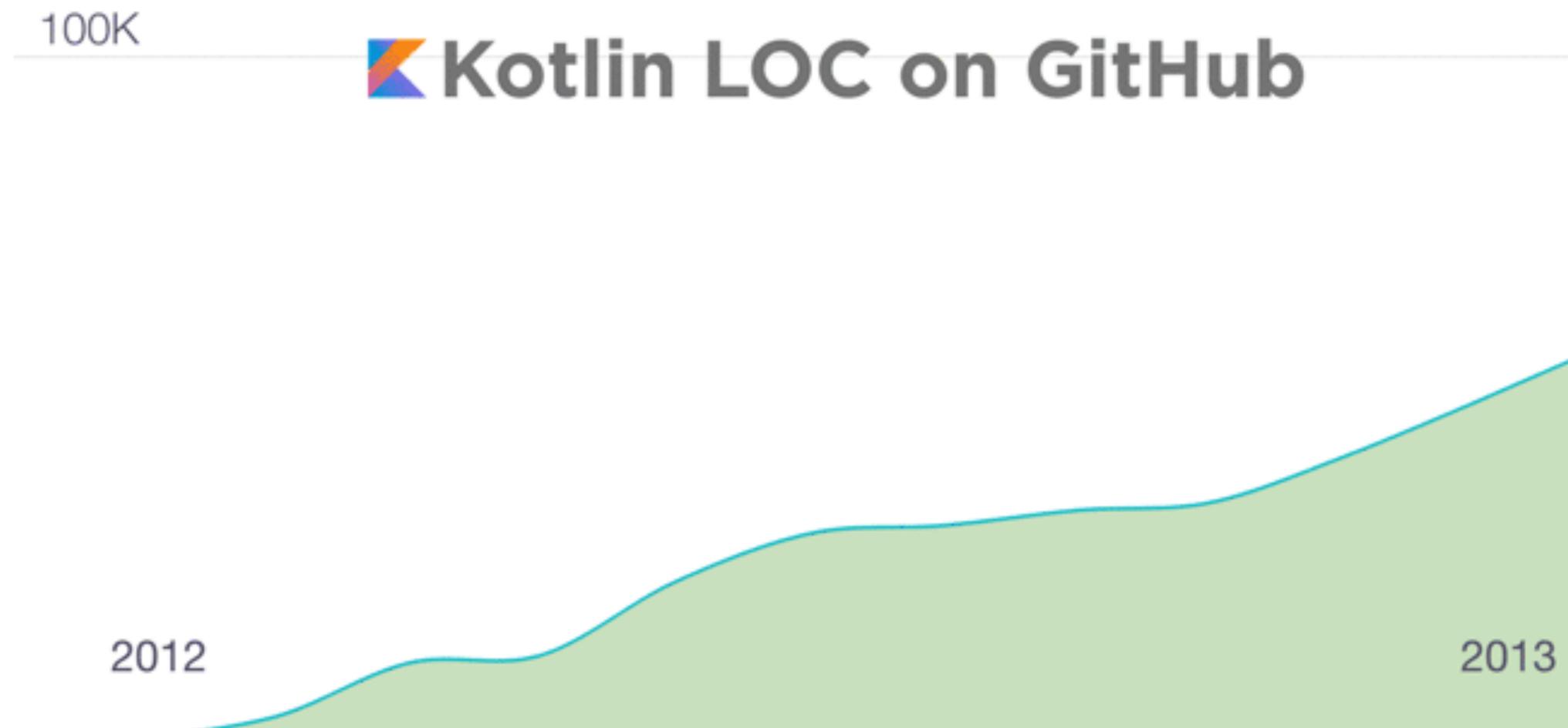
Andrey Breslav, líder do projeto Kotlin

"Kotlin é projetada para ser uma linguagem orientada a objeto de força industrial, **e melhor do que Java**, mas ainda ser totalmente **interoperável** com código Java, permitindo que as empresas possam **fazer uma migração gradual de Java para Kotlin.**"



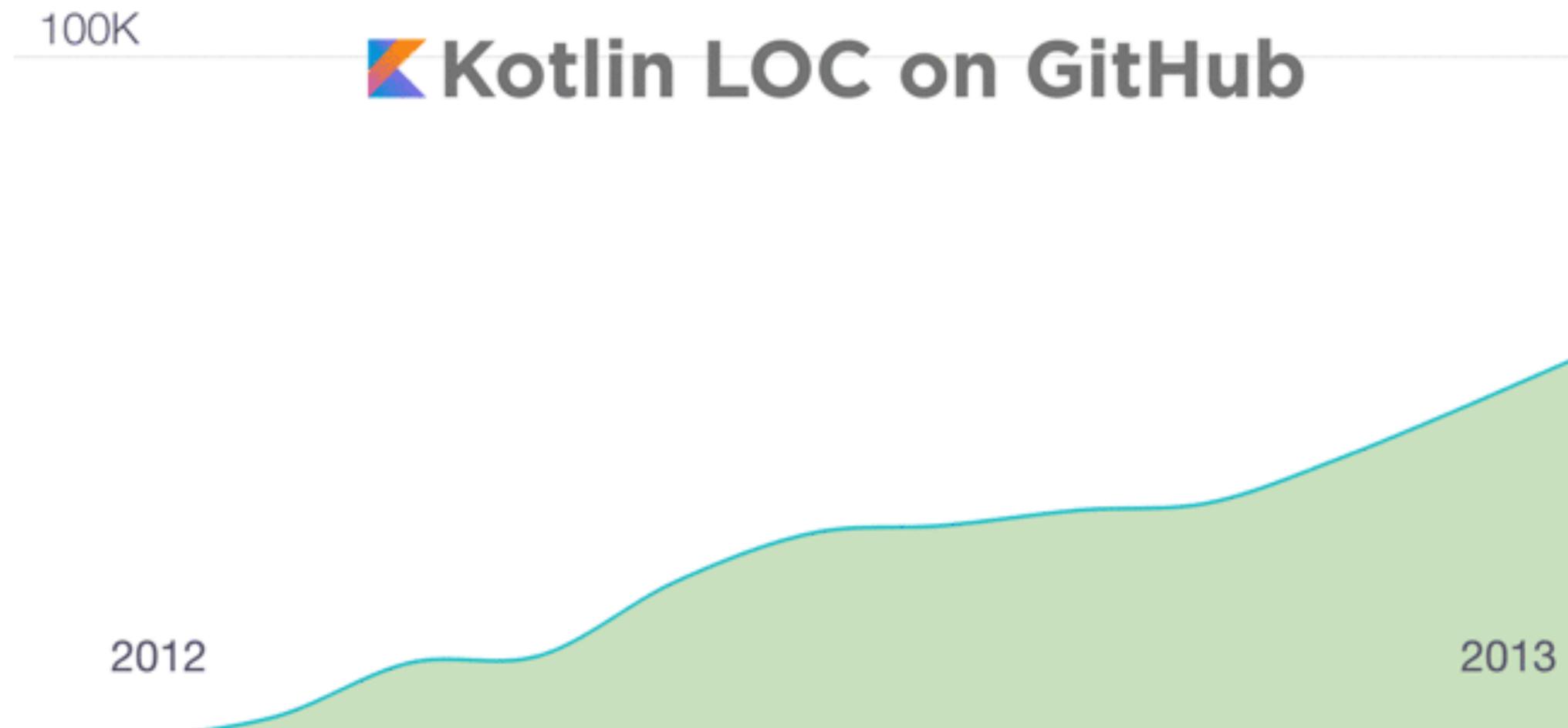
História

Tudo começou em 2011



História

Tudo começou em 2011



História

Versões

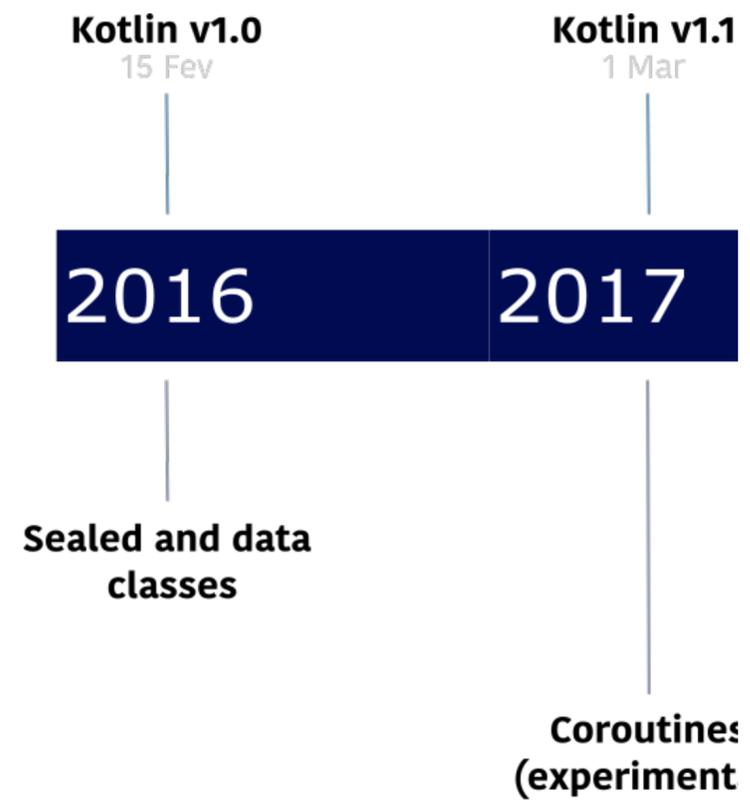
História

Versões



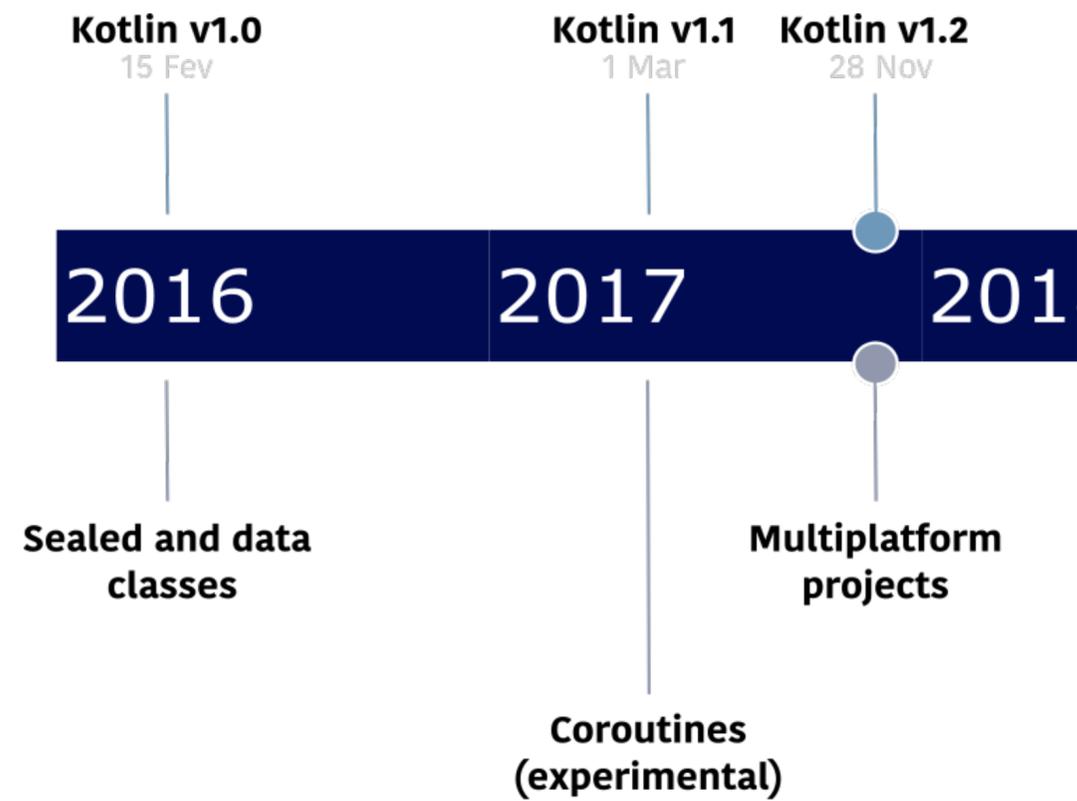
História

Versões



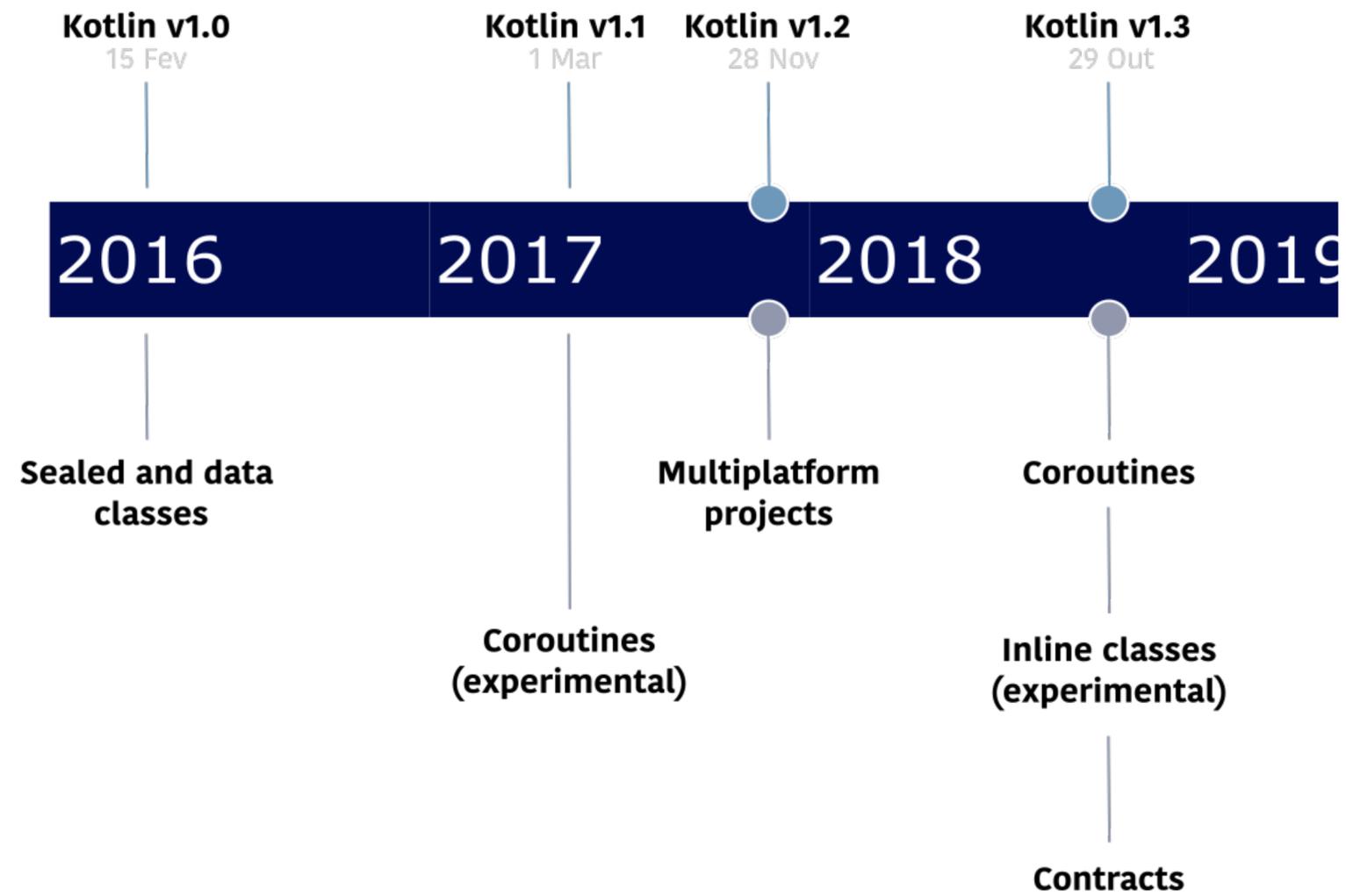
História

Versões



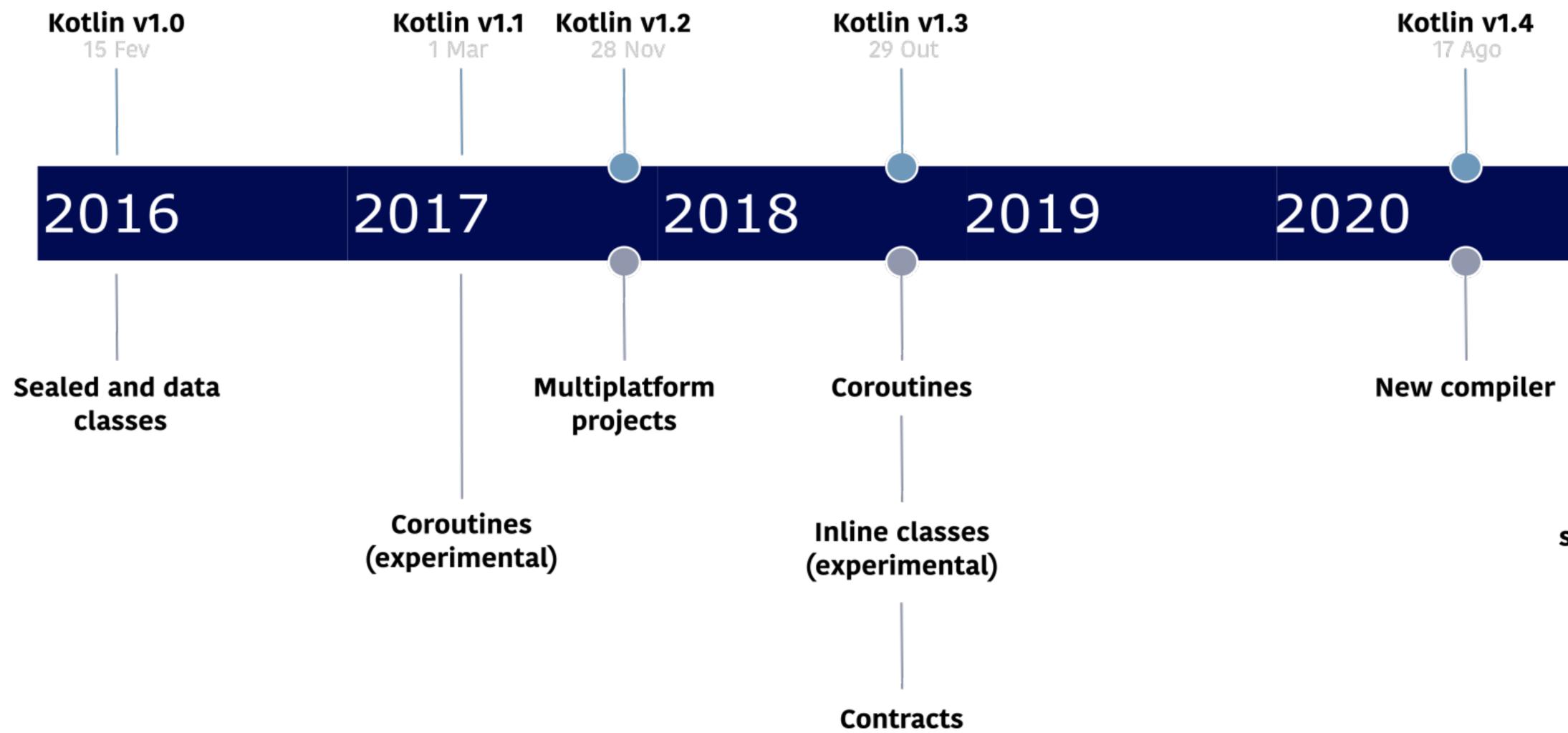
História

Versões



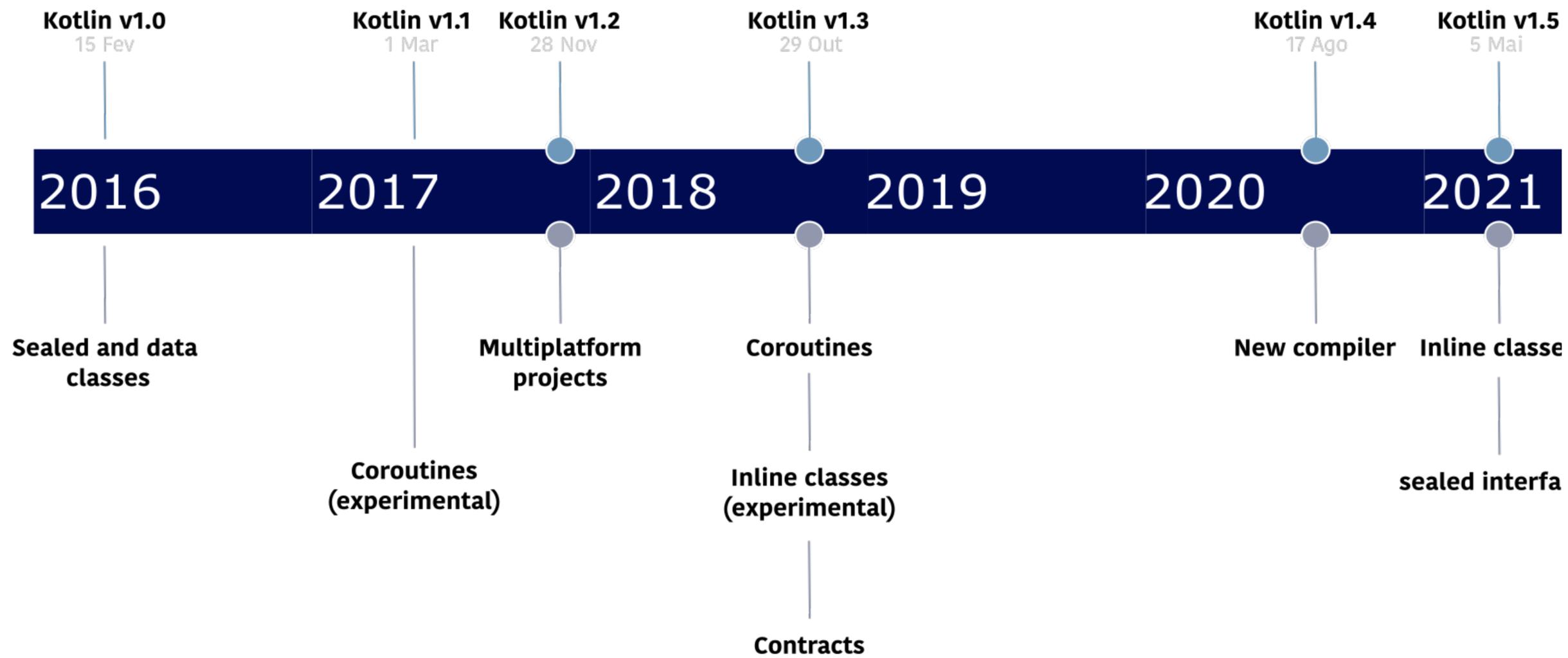
História

Versões



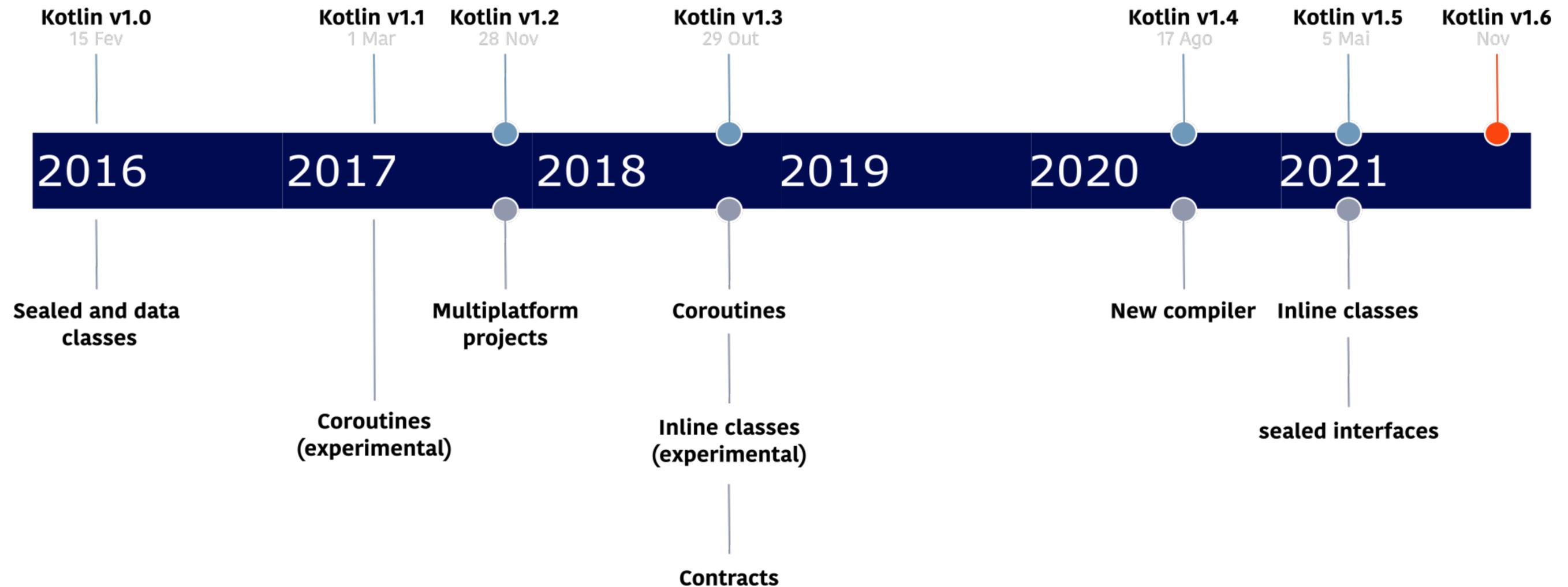
História

Versões



História

Versões



Características

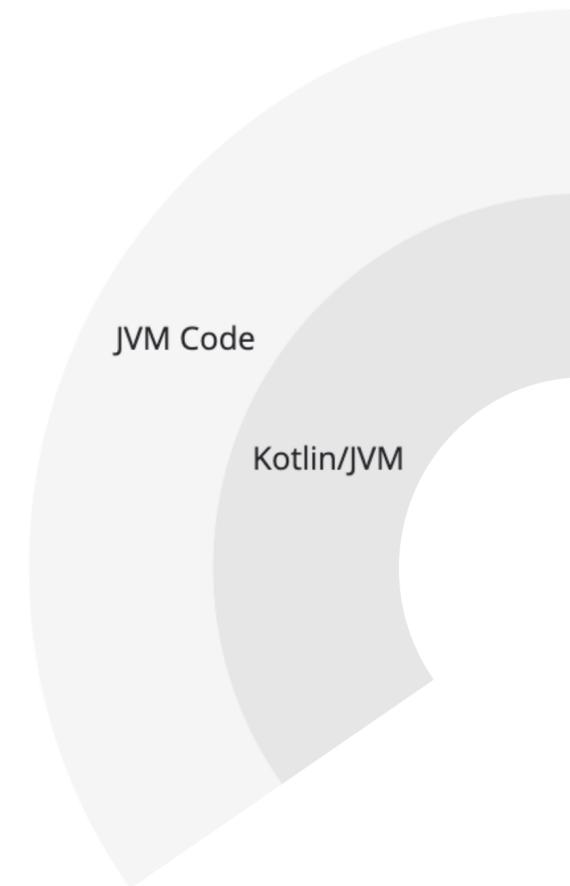
Características

Características

- Multiplataforma

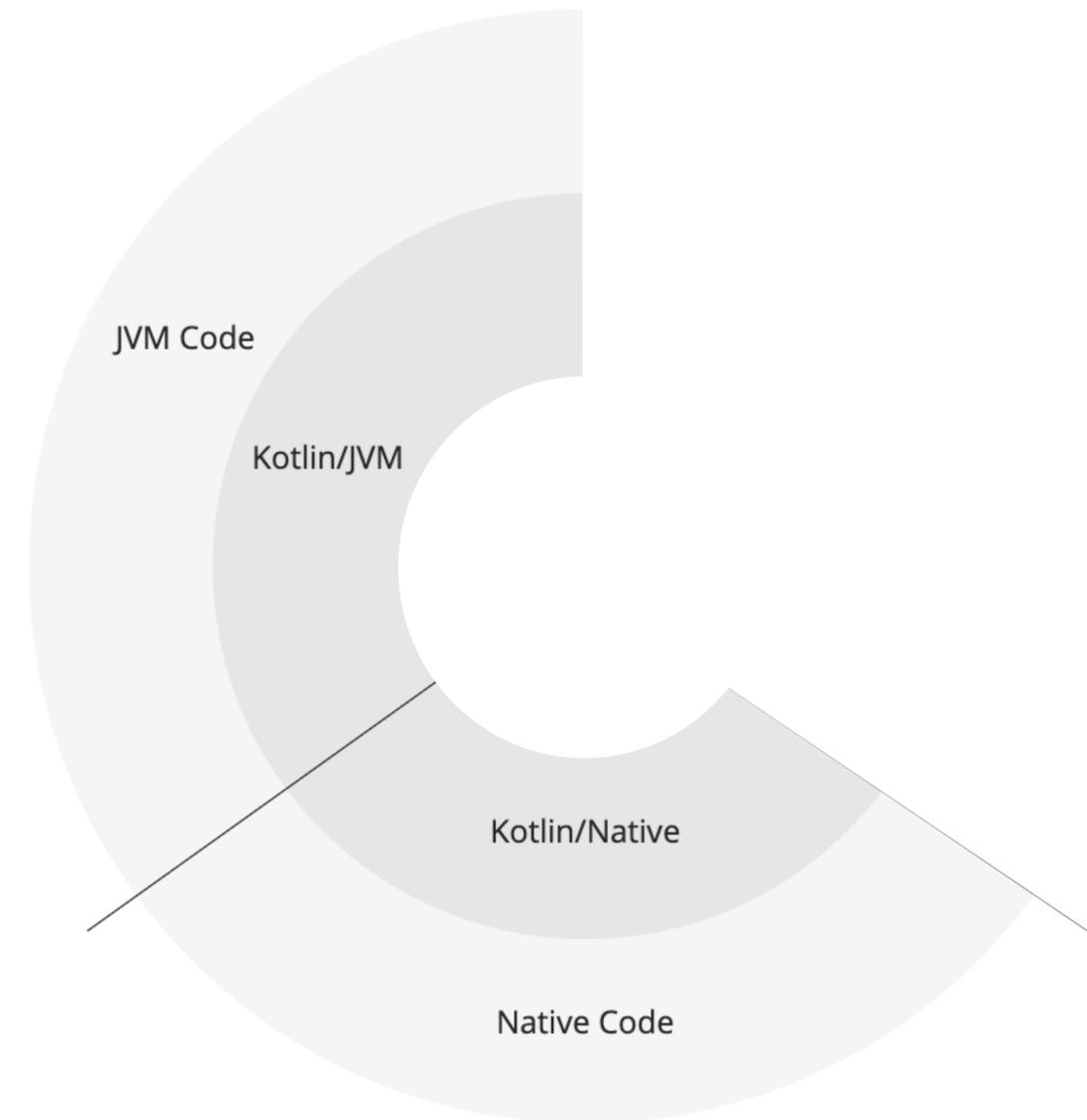
Características

- Multiplataforma
 - JVM



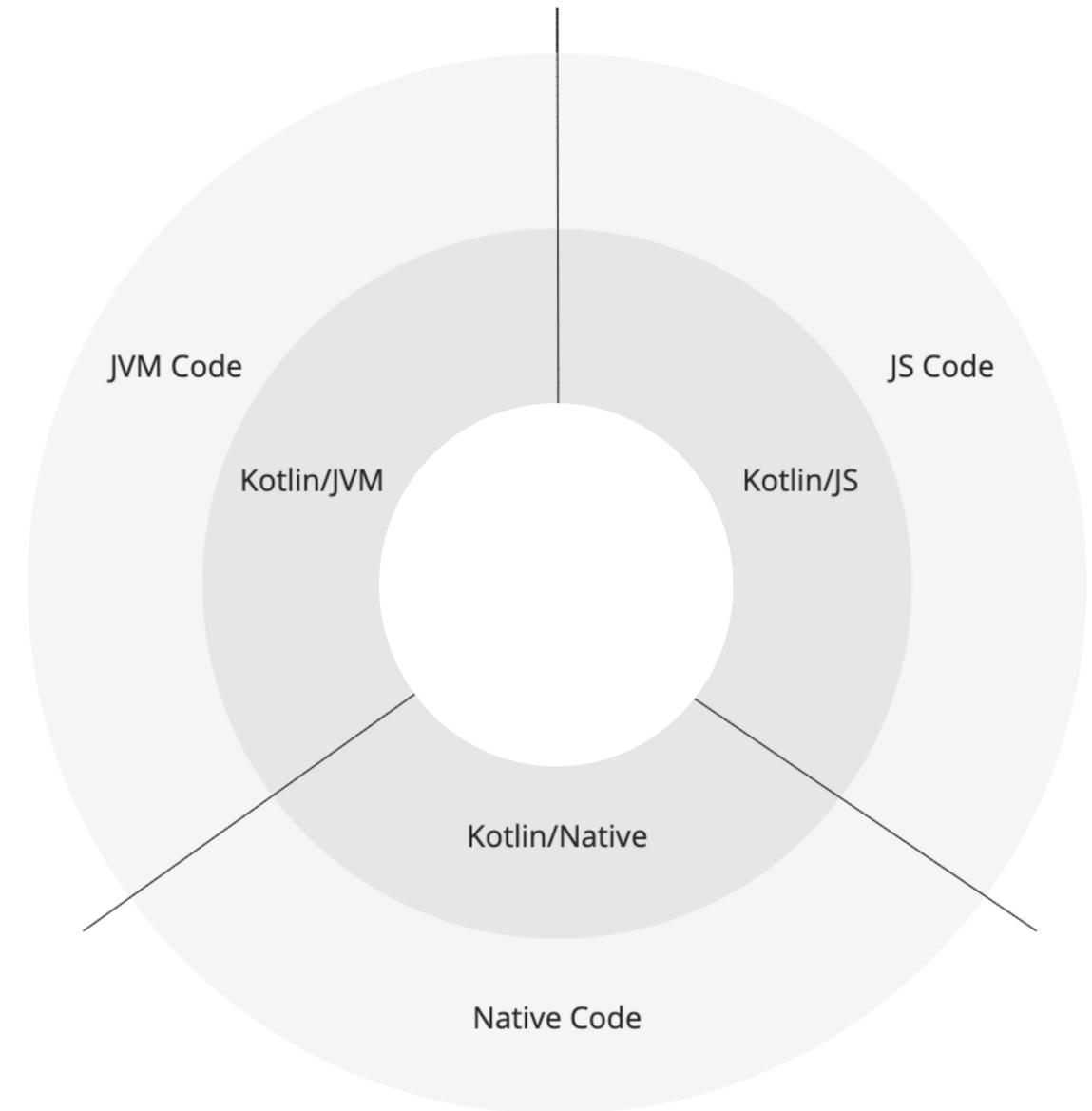
Características

- Multiplataforma
 - JVM
 - Kotlin Native



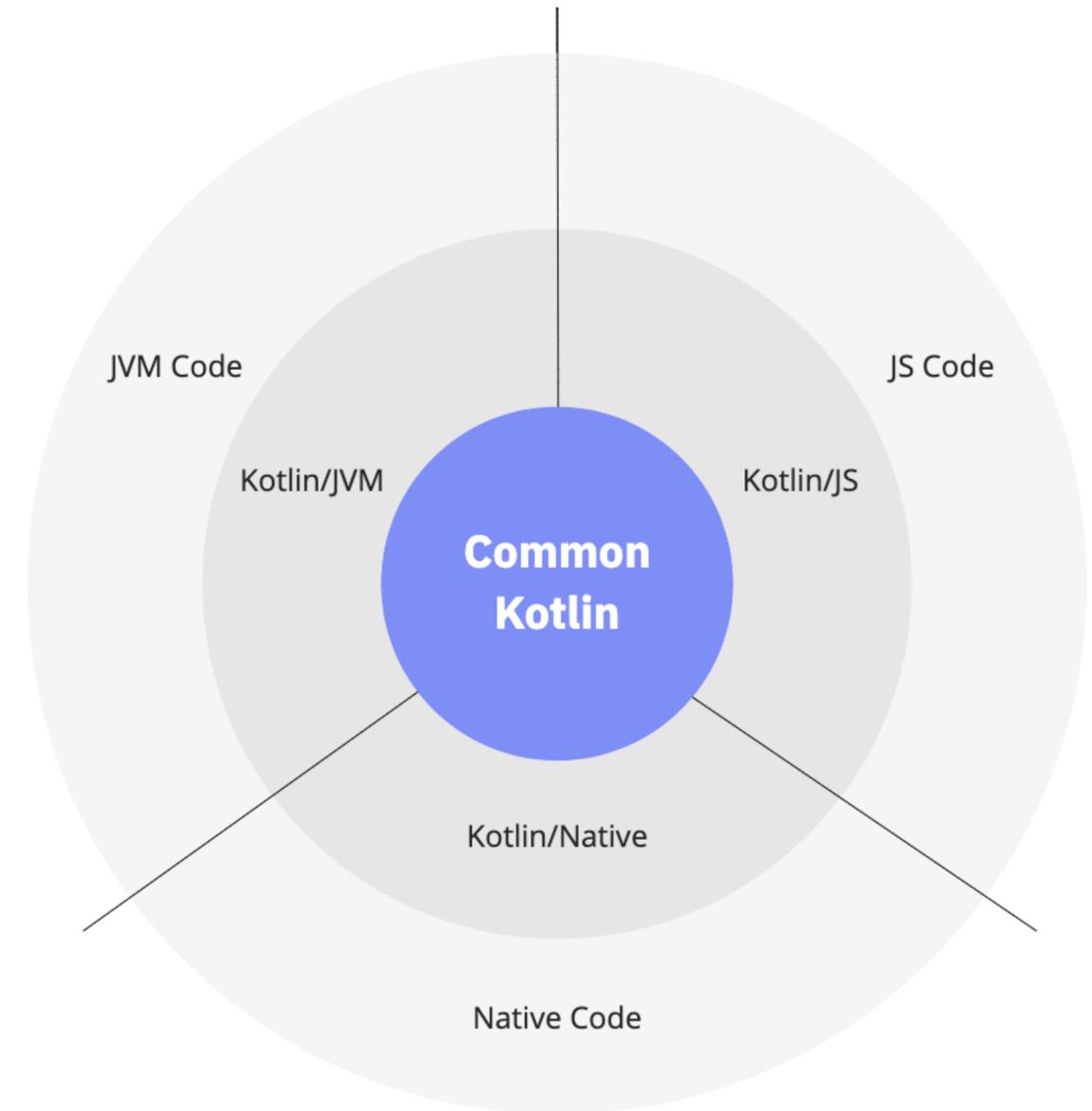
Características

- Multiplataforma
 - JVM
 - Kotlin Native
 - Kotlin for JavaScript



Características

- Multiplataforma
 - JVM
 - Kotlin Native
 - Kotlin for JavaScript



Características

Kotlin Multiplatform Mobile

Características

Kotlin Multiplatform Mobile

Kotlin



Características

Características

- Funcional e Orientada a Objetos

Características

- Funcional e Orientada a Objetos
 - Funções de primeira classe (*First-class functions*)

Características

- Funcional e Orientada a Objetos
 - Funções de primeira classe (*First-class functions*)
 - Funções podem ser: armazenadas em variáveis, passadas como parâmetros ou retornadas por outras funções.

Características

- Funcional e Orientada a Objetos
 - Funções de primeira classe (*First-class functions*)
 - Funções podem ser: armazenadas em variáveis, passadas como parâmetros ou retornadas por outras funções.
 - Imutabilidade

Características

- Funcional e Orientada a Objetos
 - Funções de primeira classe (*First-class functions*)
 - Funções podem ser: armazenadas em variáveis, passadas como parâmetros ou retornadas por outras funções.
 - Imutabilidade
 - Garante que o estado dos objetos não mudam após sua criação

Características

- Funcional e Orientada a Objetos
 - Funções de primeira classe (*First-class functions*)
 - Funções podem ser: armazenadas em variáveis, passadas como parâmetros ou retornadas por outras funções.
 - Imutabilidade
 - Garante que o estado dos objetos não mudam após sua criação
 - Sem efeitos colaterais (*No side effects*)

Características

- Funcional e Orientada a Objetos
 - Funções de primeira classe (*First-class functions*)
 - Funções podem ser: armazenadas em variáveis, passadas como parâmetros ou retornadas por outras funções.
 - Imutabilidade
 - Garante que o estado dos objetos não mudam após sua criação
 - Sem efeitos colaterais (*No side effects*)
 - Funções puras sempre retornam o mesmo resultado da a uma entrada

Características

Características

- Estaticamente Tipada

Características

- Estaticamente Tipada
- Moderna

Características

- Estaticamente Tipada
- Moderna
- Concisa

Características

- Estaticamente Tipada
- Moderna
- Concisa
- Segura

Características

- Estaticamente Tipada
- Moderna
- Concisa
- Segura
- Interoperável com Java

Características

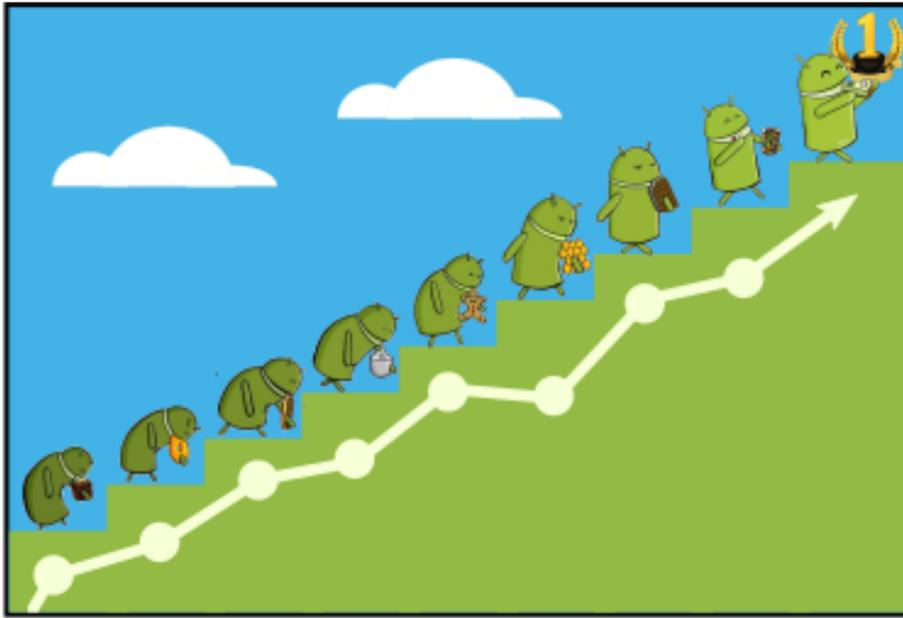
- Estaticamente Tipada
- Moderna
- Concisa
- Segura
- Interoperável com Java
- Free e Open Source

Kotlin e Android

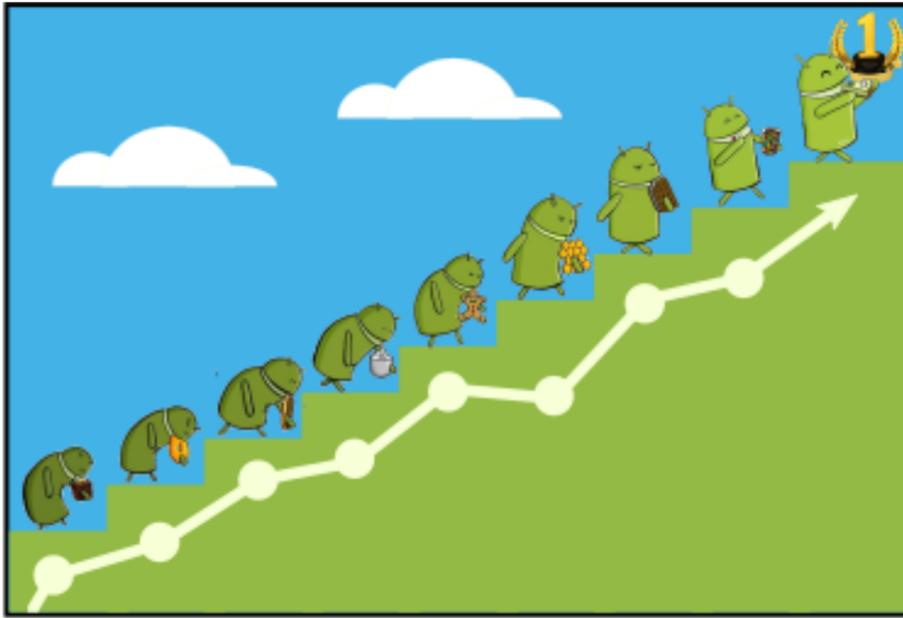
In 2008



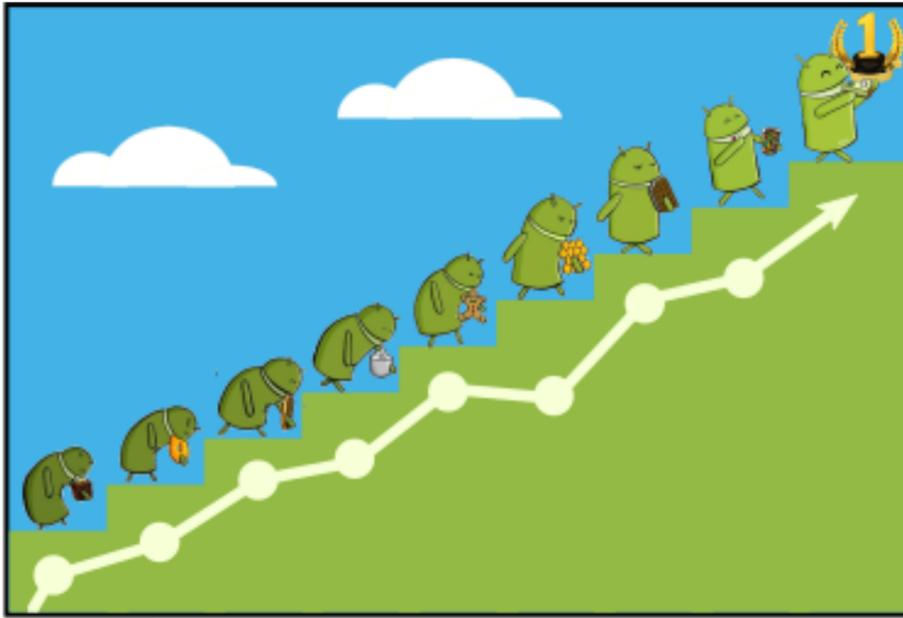
In 2008



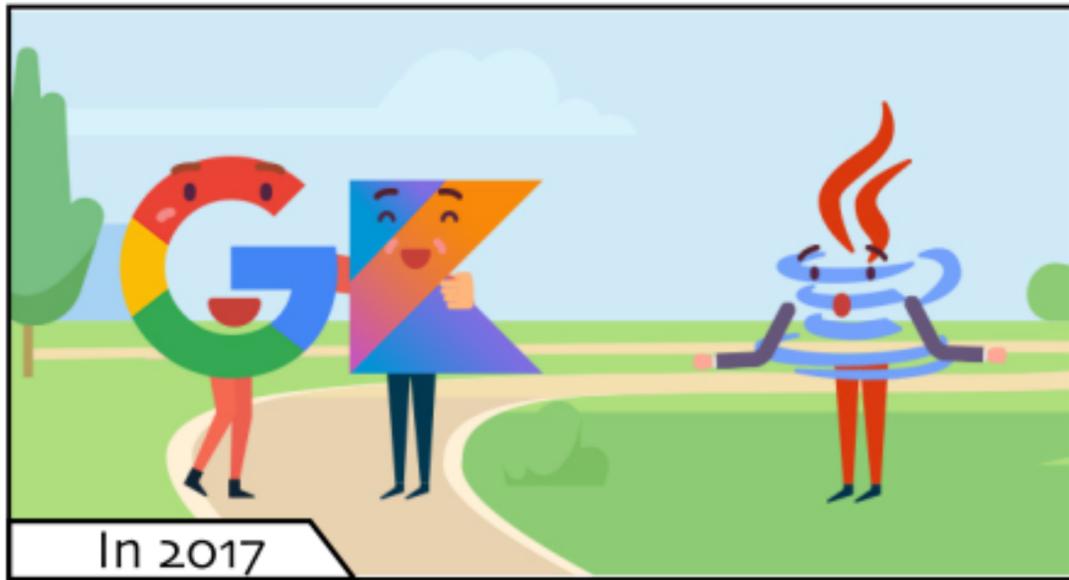
In 2008



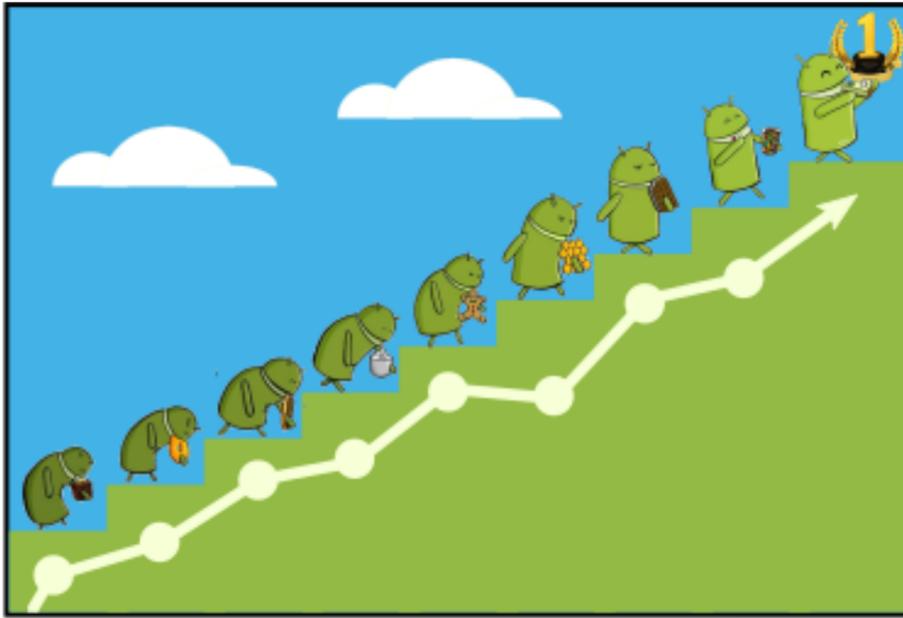
In 2008



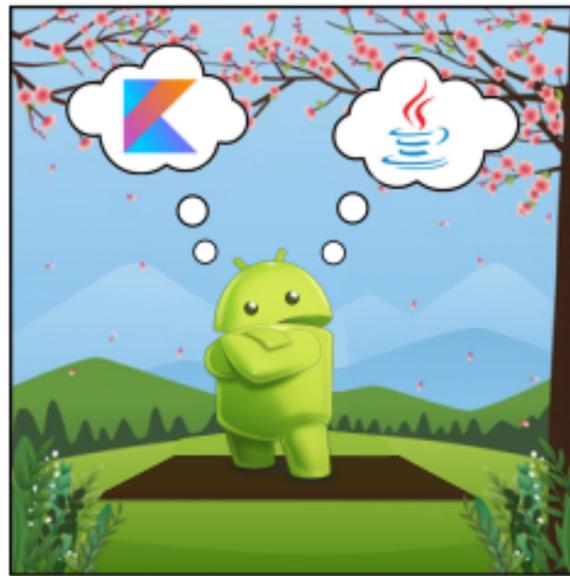
In 2017

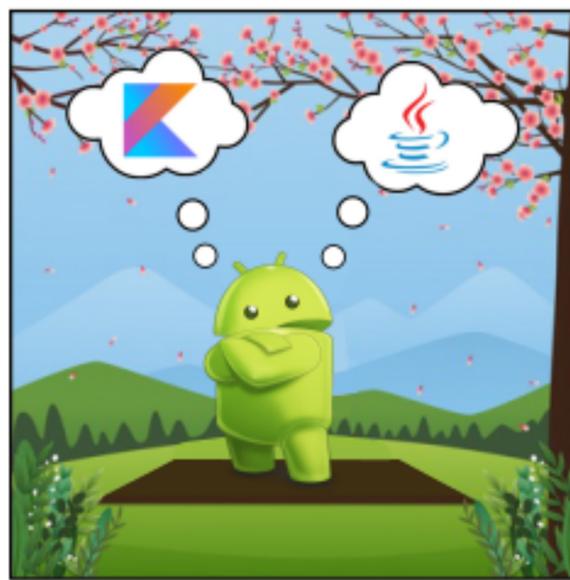
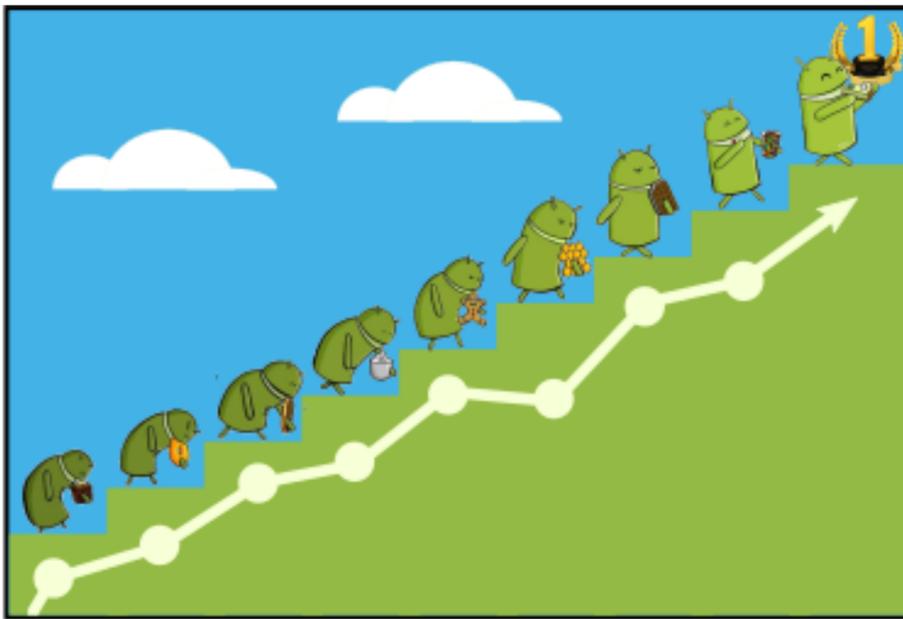
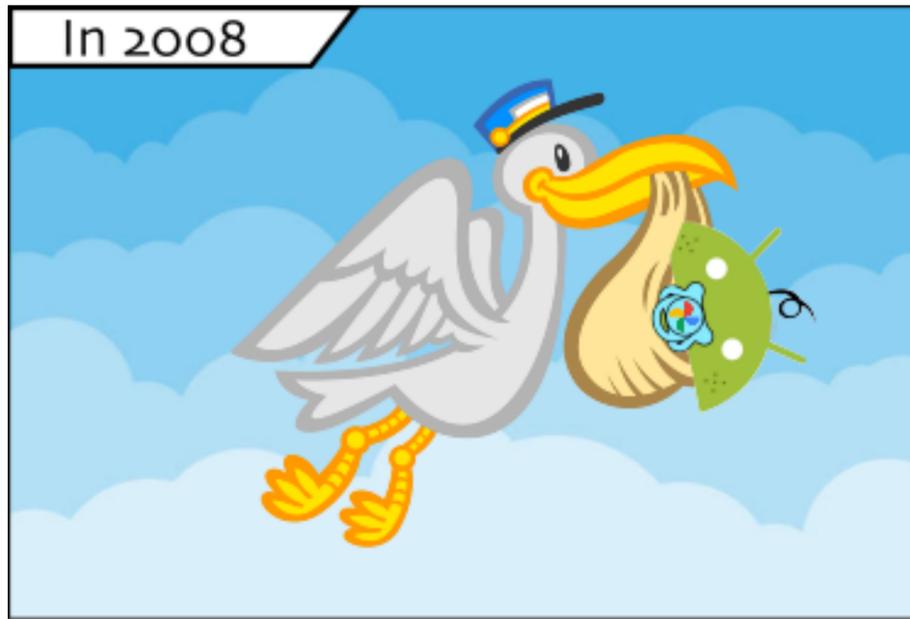


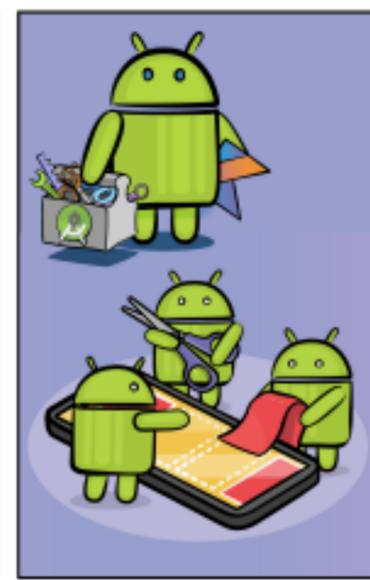
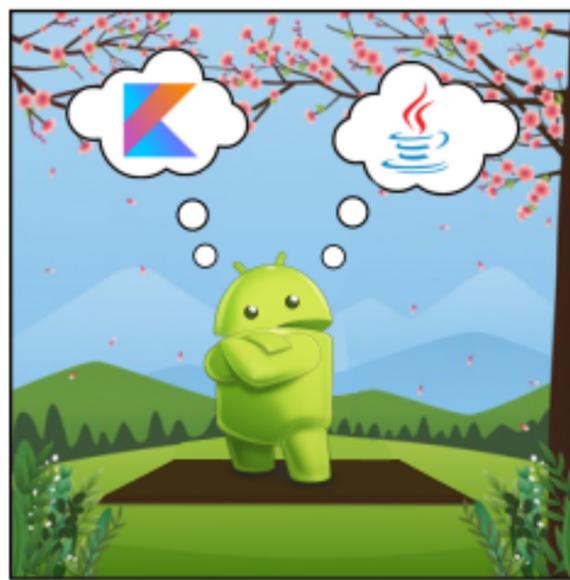
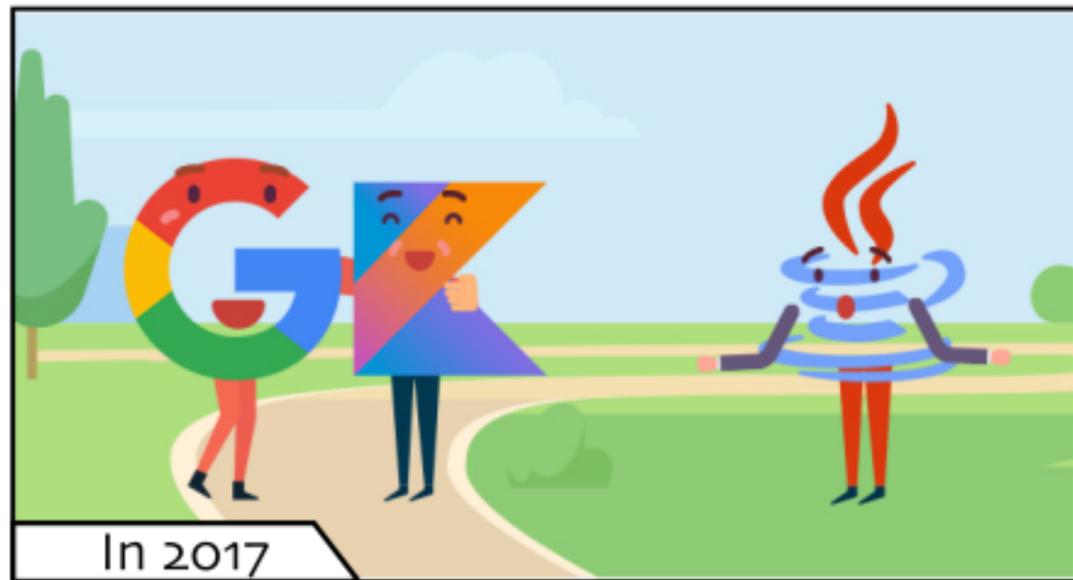
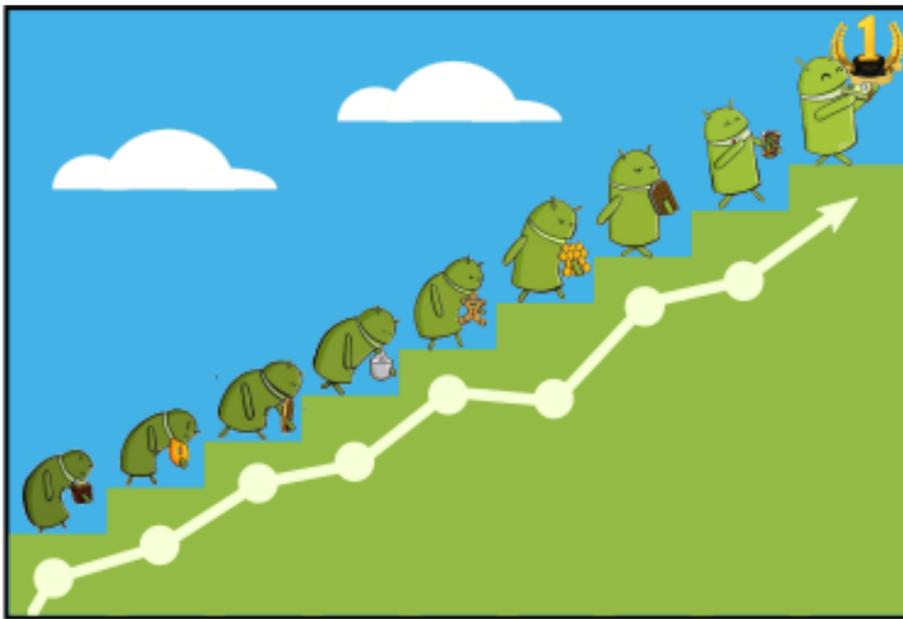
In 2008

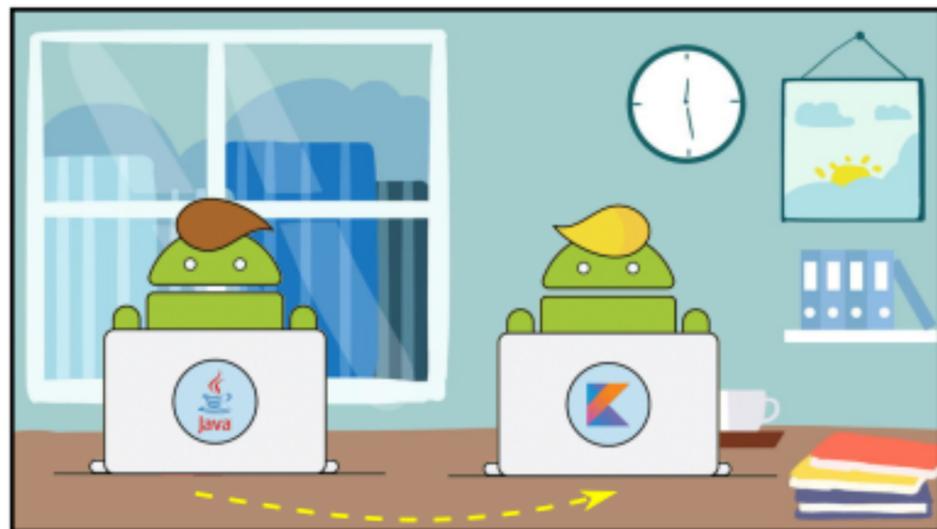
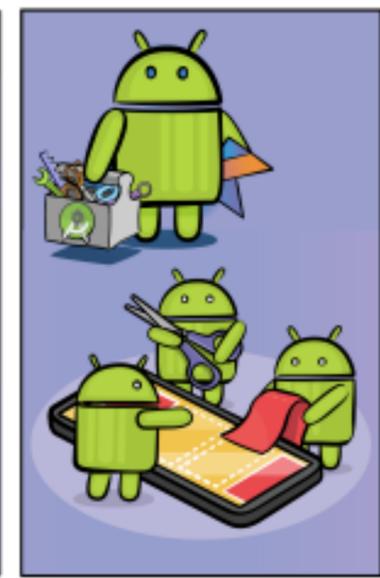
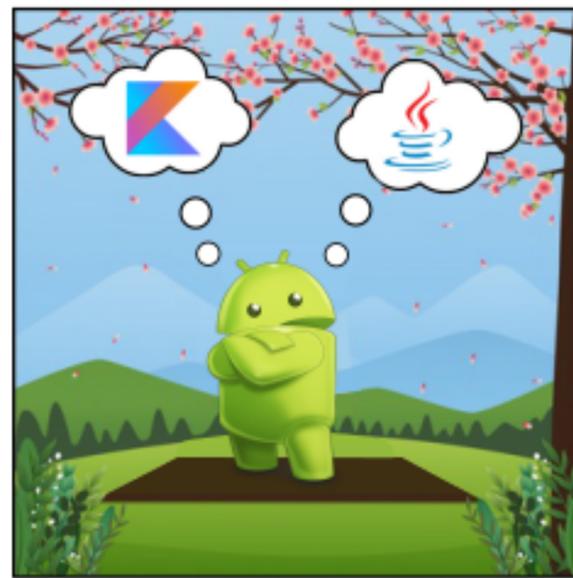
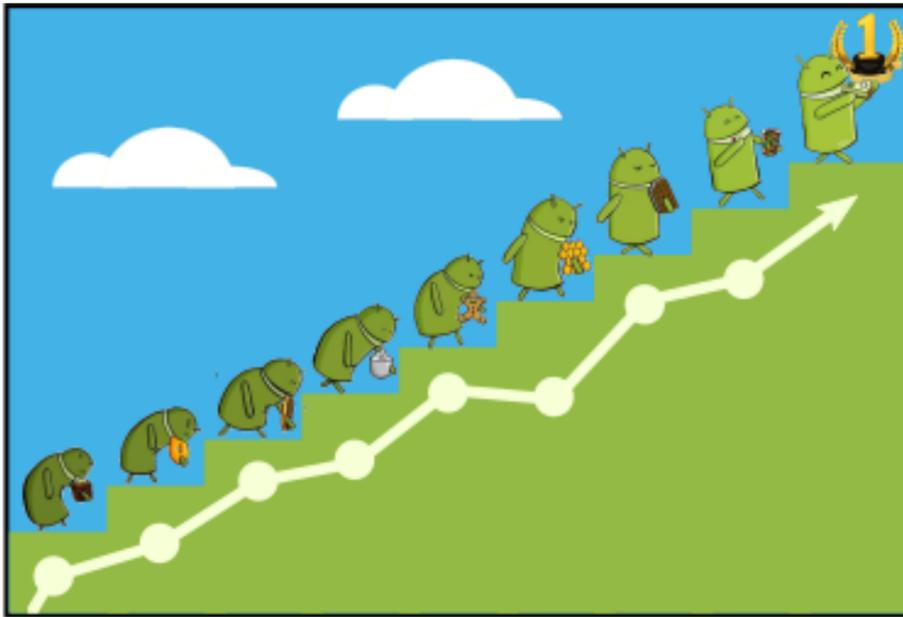


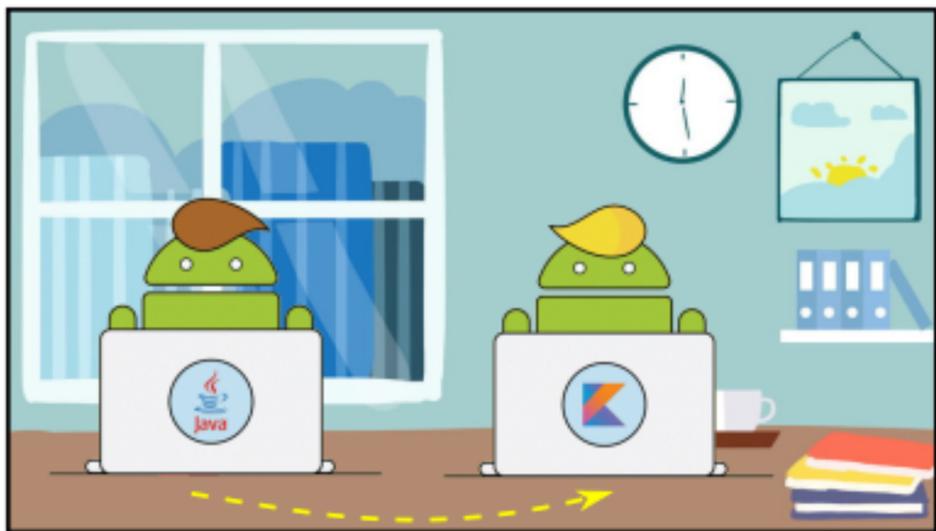
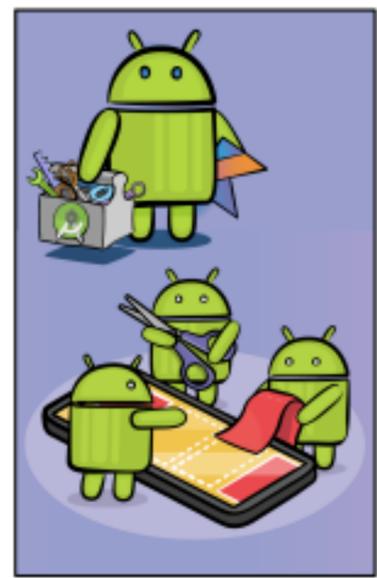
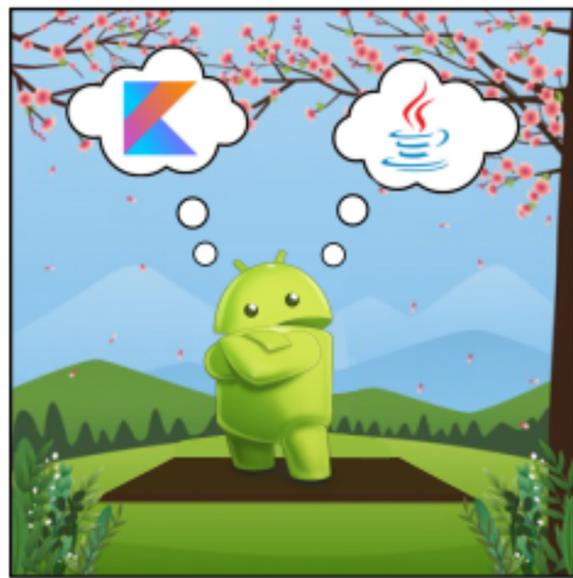
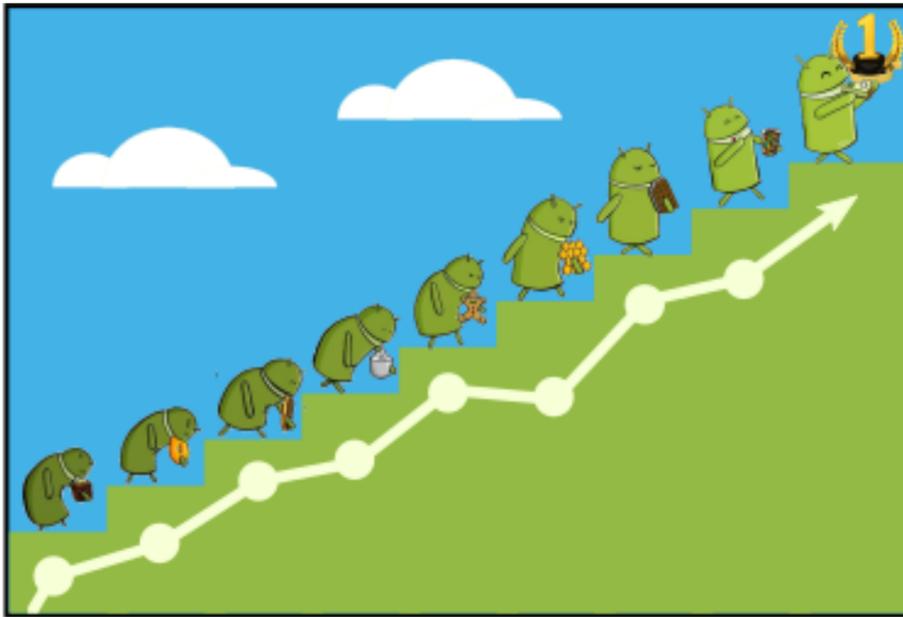
In 2017

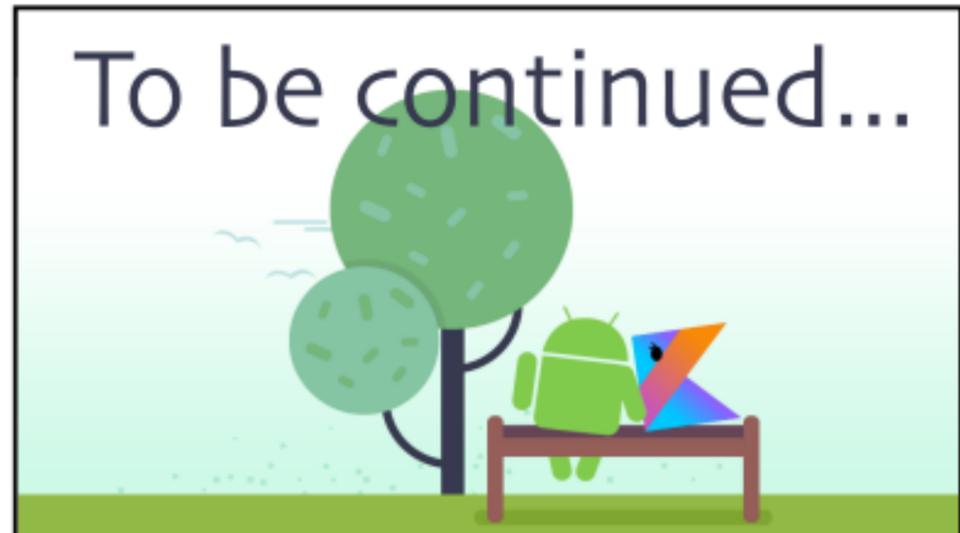
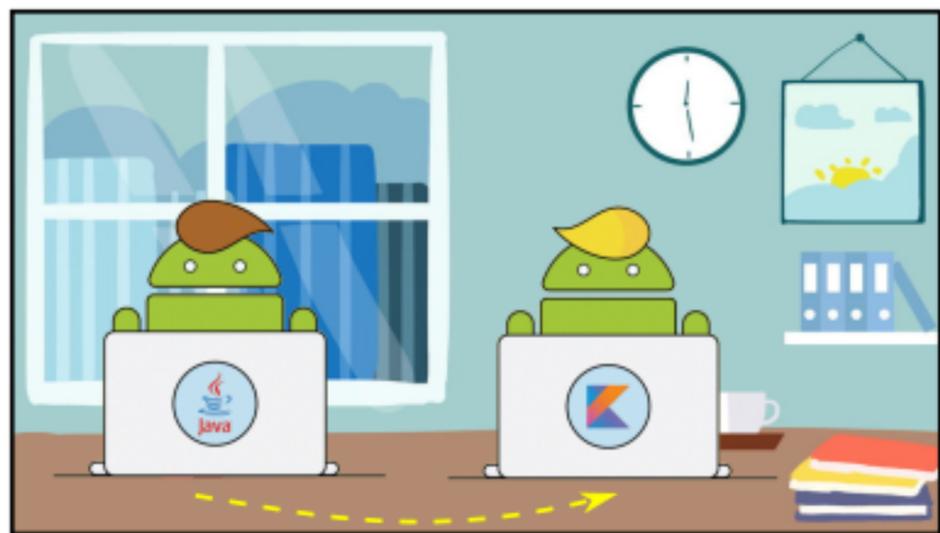
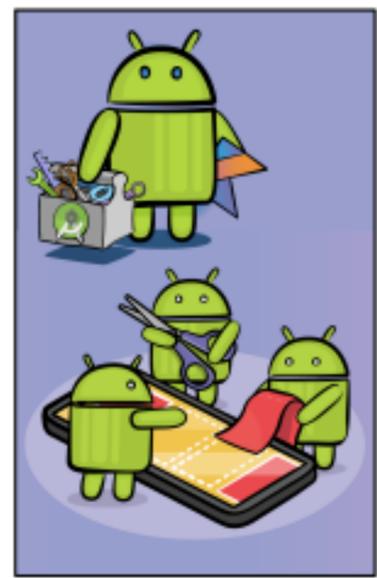
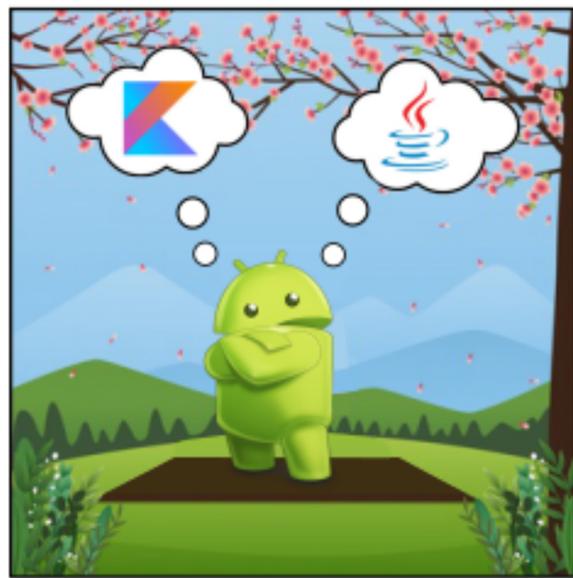
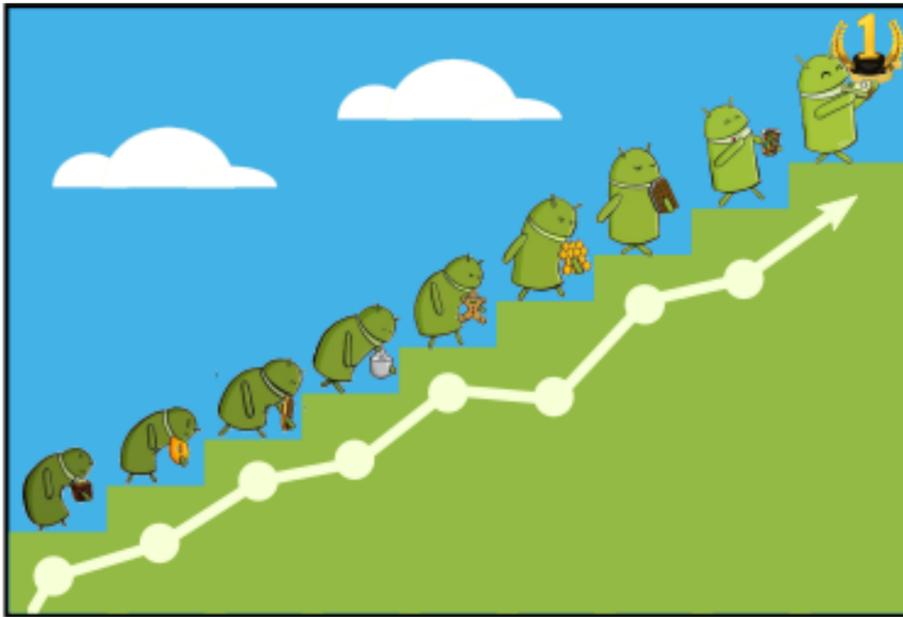








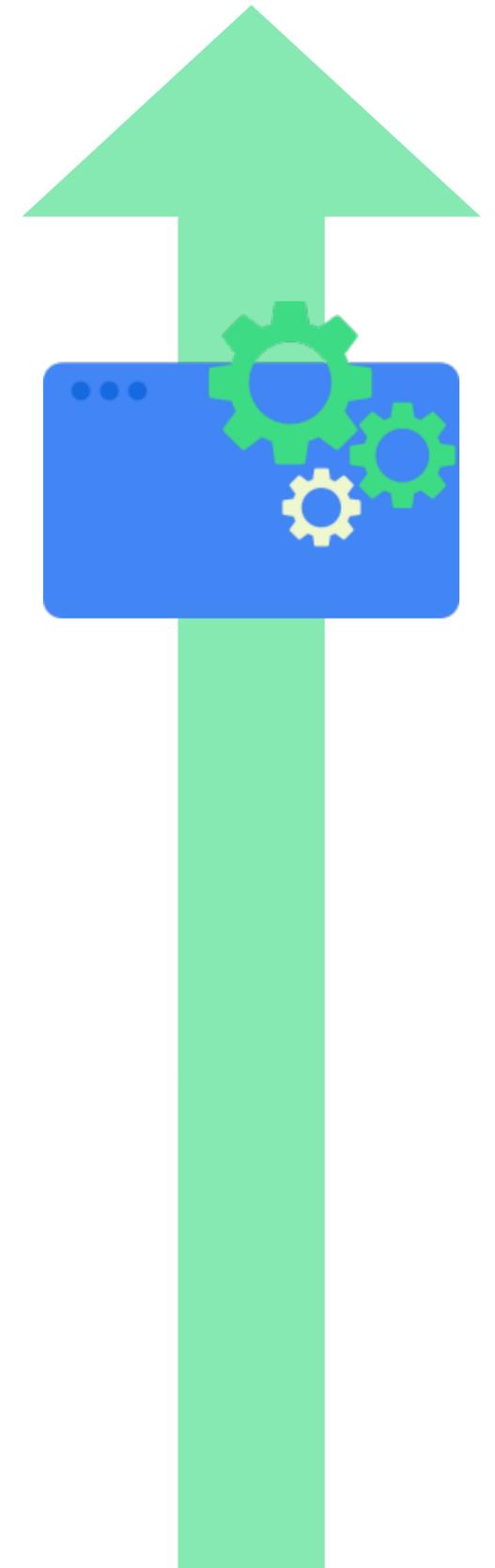




De acordo com o Google

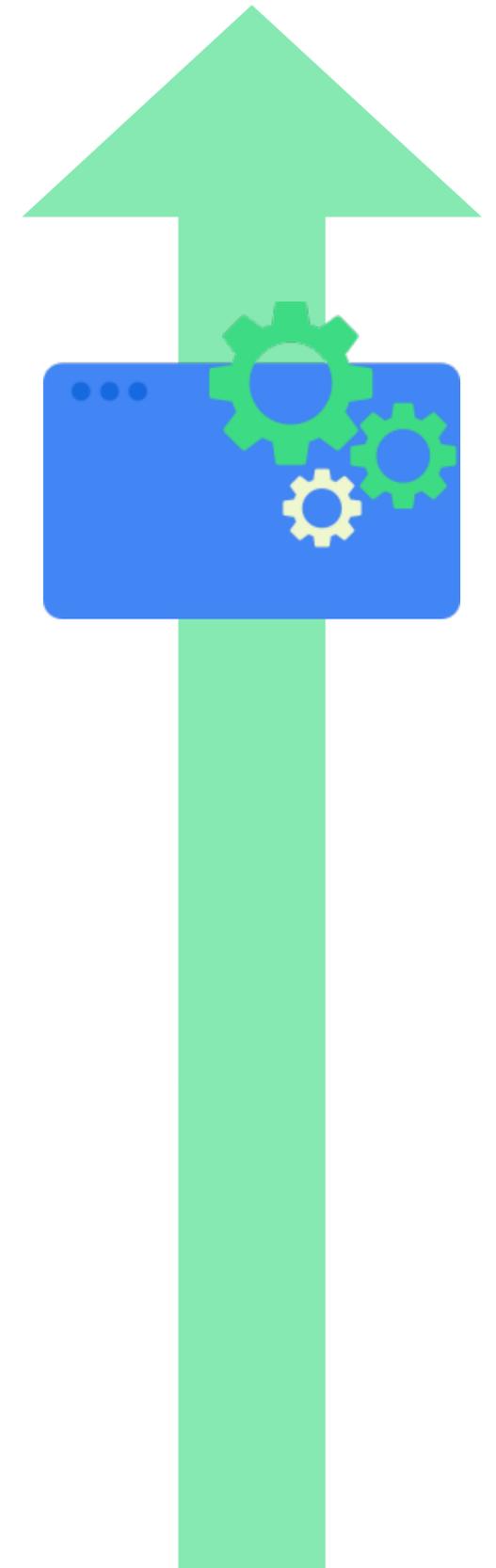
De acordo com o Google

- Aumenta a produtividade



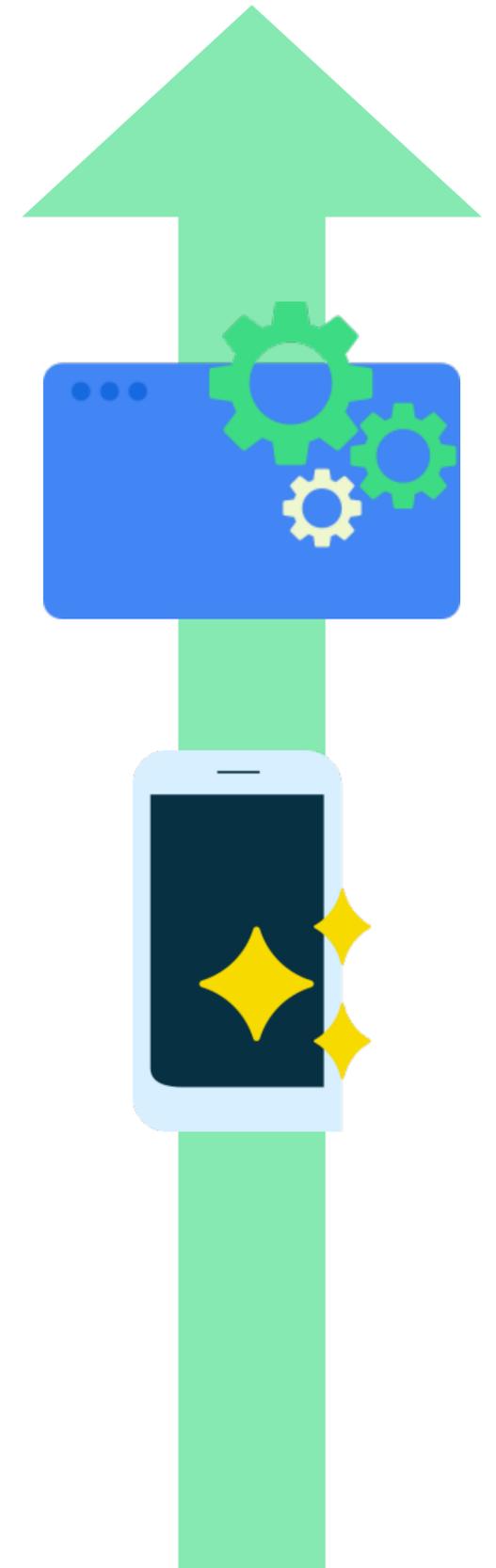
De acordo com o Google

- **Aumenta a produtividade**
 - Menos código boilerplate para manter, permite que desenvolvedores foque em expressar suas idéias



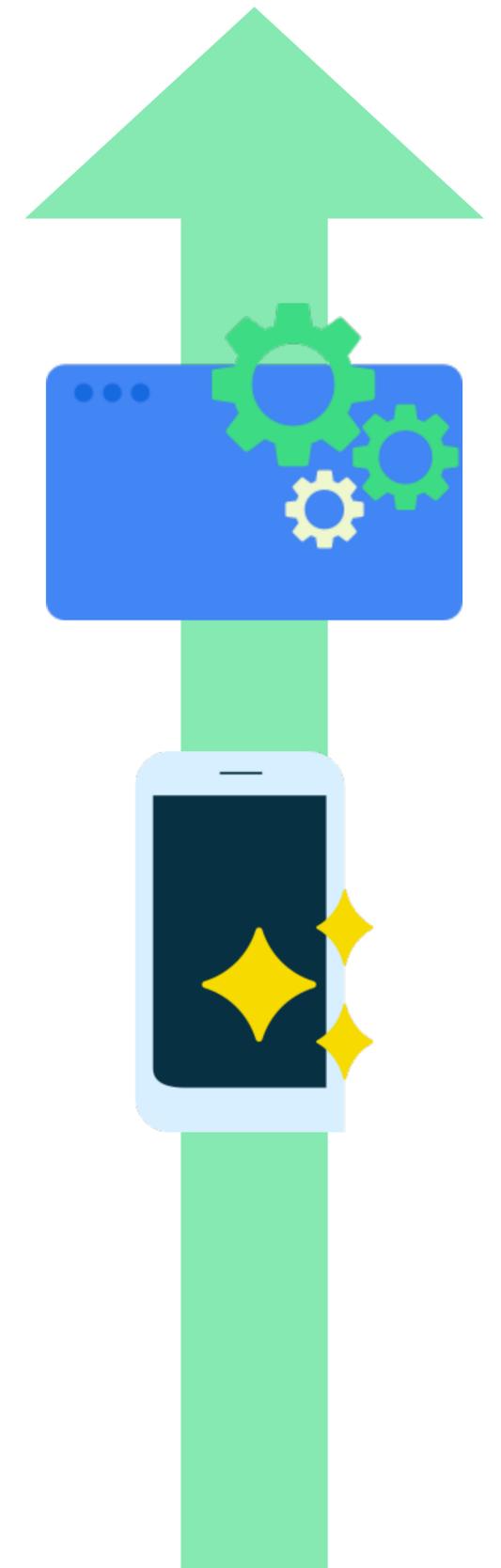
De acordo com o Google

- **Aumenta a produtividade**
 - Menos código boilerplate para manter, permite que desenvolvedores foque em expressar suas idéias
- **Aplicações de qualidade superior**



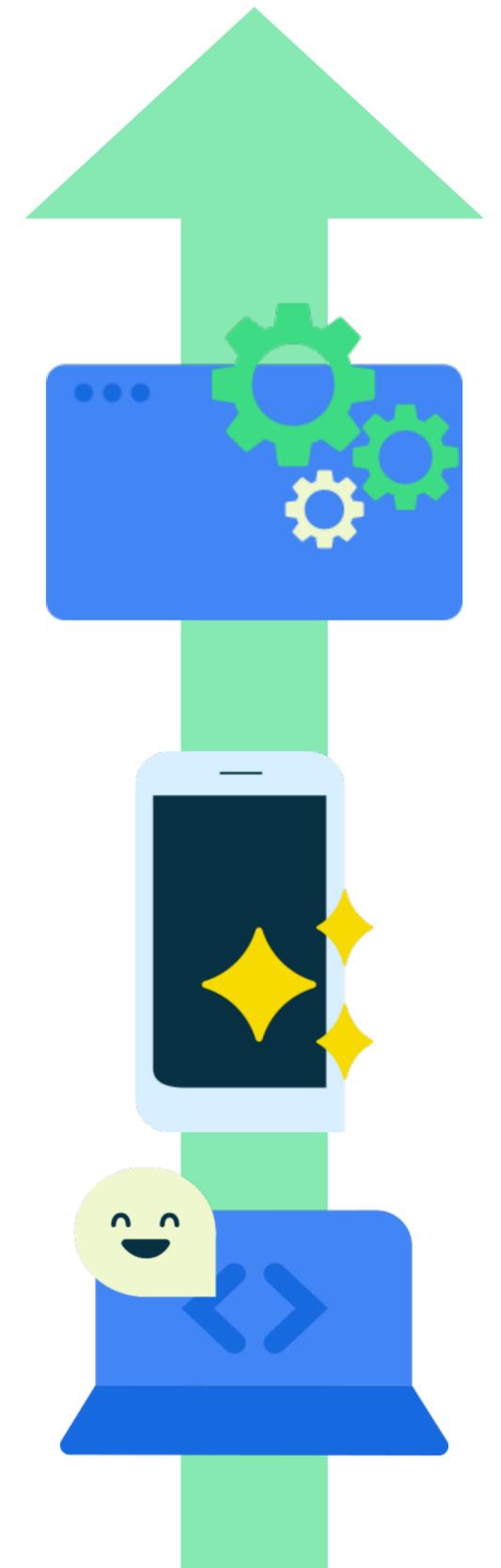
De acordo com o Google

- **Aumenta a produtividade**
 - Menos código boilerplate para manter, permite que desenvolvedores foque em expressar suas idéias
- **Aplicações de qualidade superior**
 - Dentre as 1000 aplicações no top do Google Play, aquelas que usam Kotlin quebram 20% menos



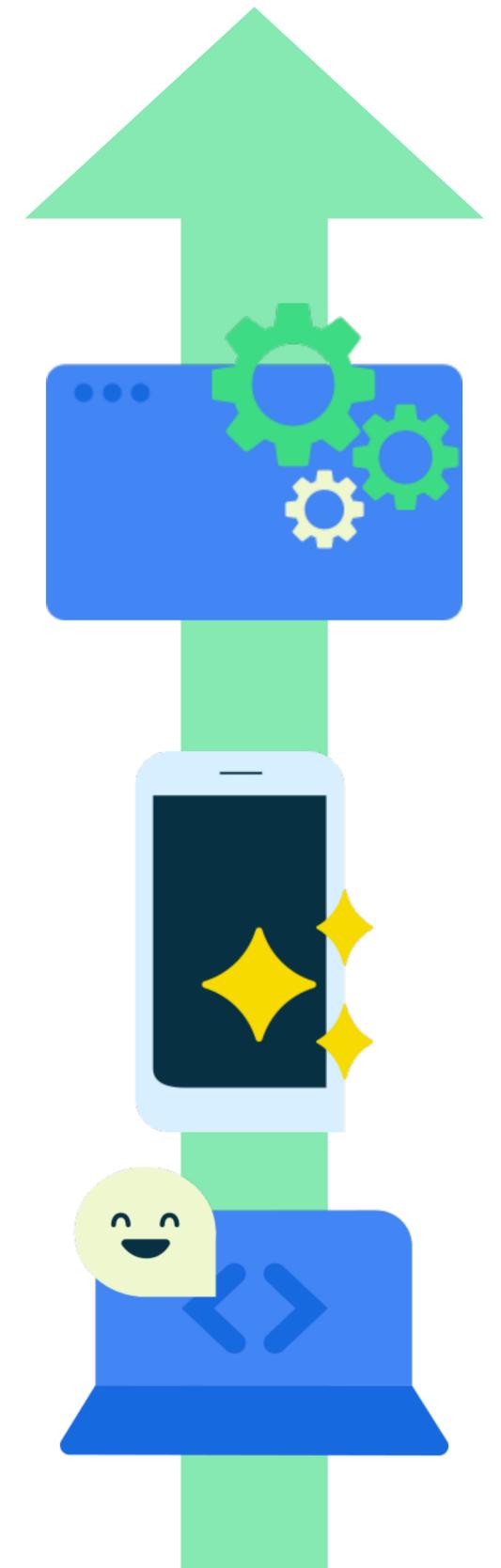
De acordo com o Google

- **Aumenta a produtividade**
 - Menos código boilerplate para manter, permite que desenvolvedores foque em expressar suas idéias
- **Aplicações de qualidade superior**
 - Dentre as 1000 aplicações no top do Google Play, aquelas que usam Kotlin quebram 20% menos
- **Desenvolvedores mais felizes**



De acordo com o Google

- **Aumenta a produtividade**
 - Menos código boilerplate para manter, permite que desenvolvedores foque em expressar suas idéias
- **Aplicações de qualidade superior**
 - Dentre as 1000 aplicações no top do Google Play, aquelas que usam Kotlin quebram 20% menos
- **Desenvolvedores mais felizes**
 - Desenvolvedores Android reportaram estar mais felizes por usarem Kotlin



De acordo com o Google

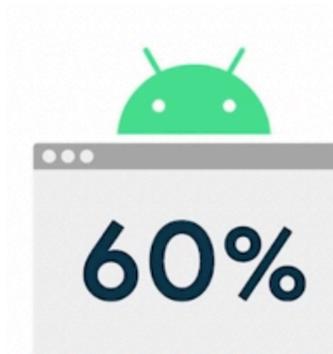
De acordo com o Google

Junte-se a mais de 5.8 milhões de programadores

De acordo com o Google

Junte-se a mais de 5.8 milhões de programadores

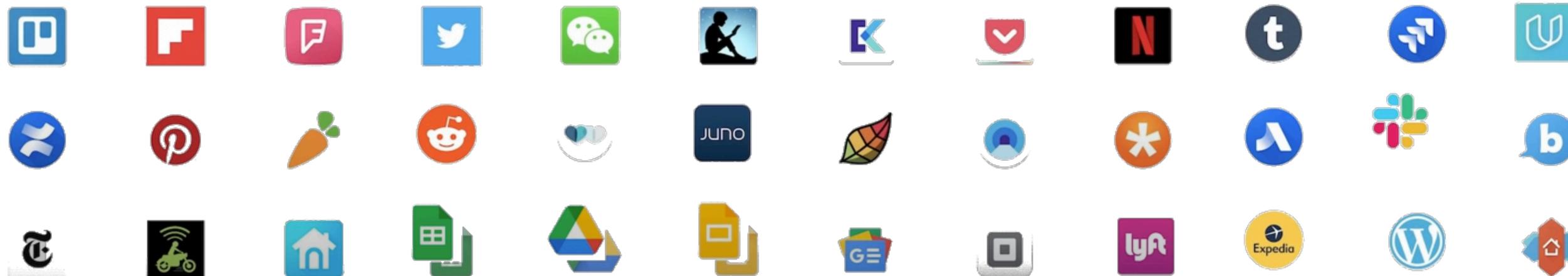
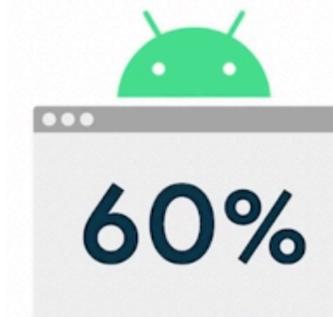
- Kotlin é utilizado por **mais de 60%** dos desenvolvedores Android profissionais.



De acordo com o Google

Junte-se a mais de 5.8 milhões de programadores

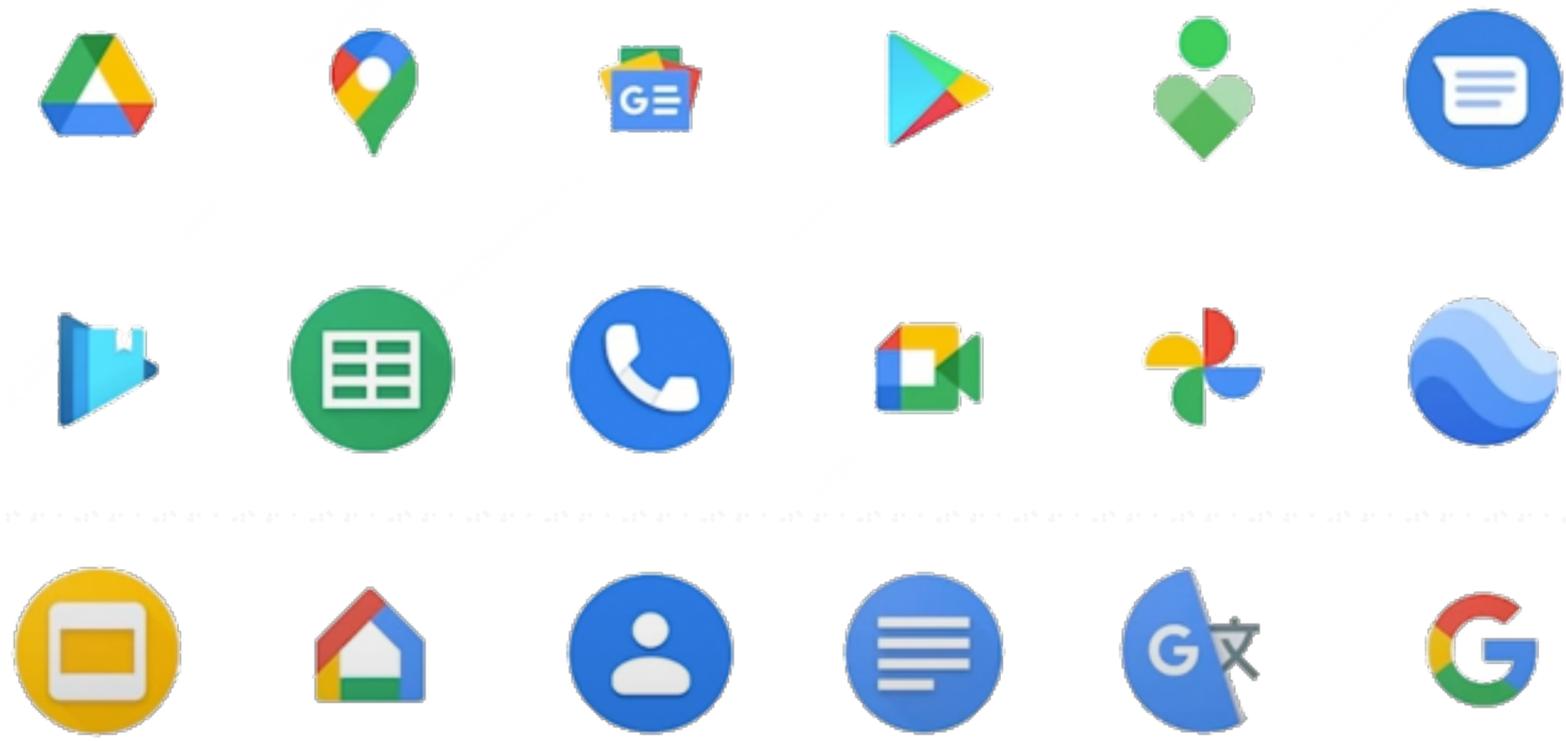
- Kotlin é utilizado por **mais de 60%** dos desenvolvedores Android profissionais.
- **80% das aplicações no top 1,000** contém código escrito em Kotlin



De acordo com o Google

De acordo com o Google

Mais de 70 aplicativos do Google

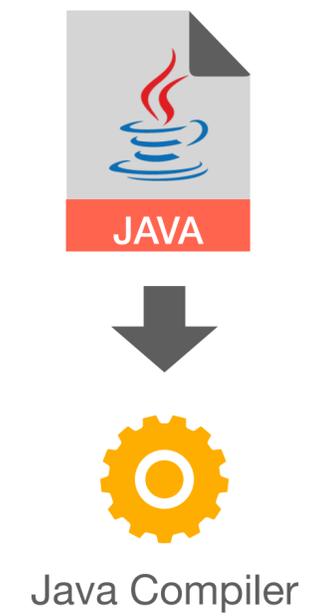
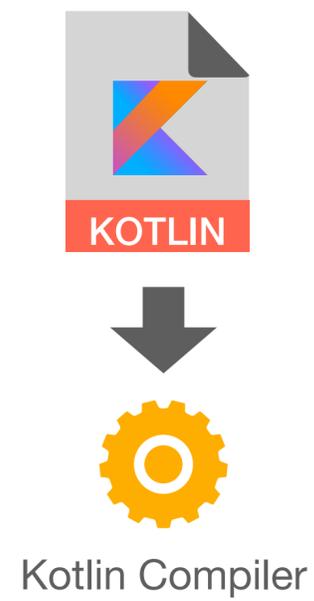


Interoperabilidade

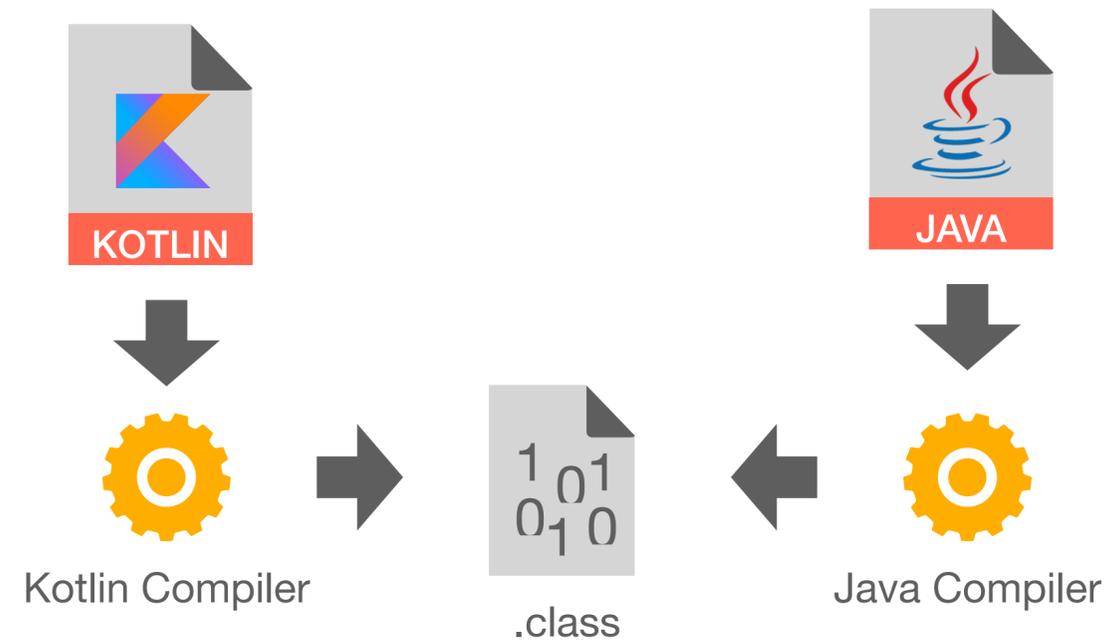
Interoperabilidade



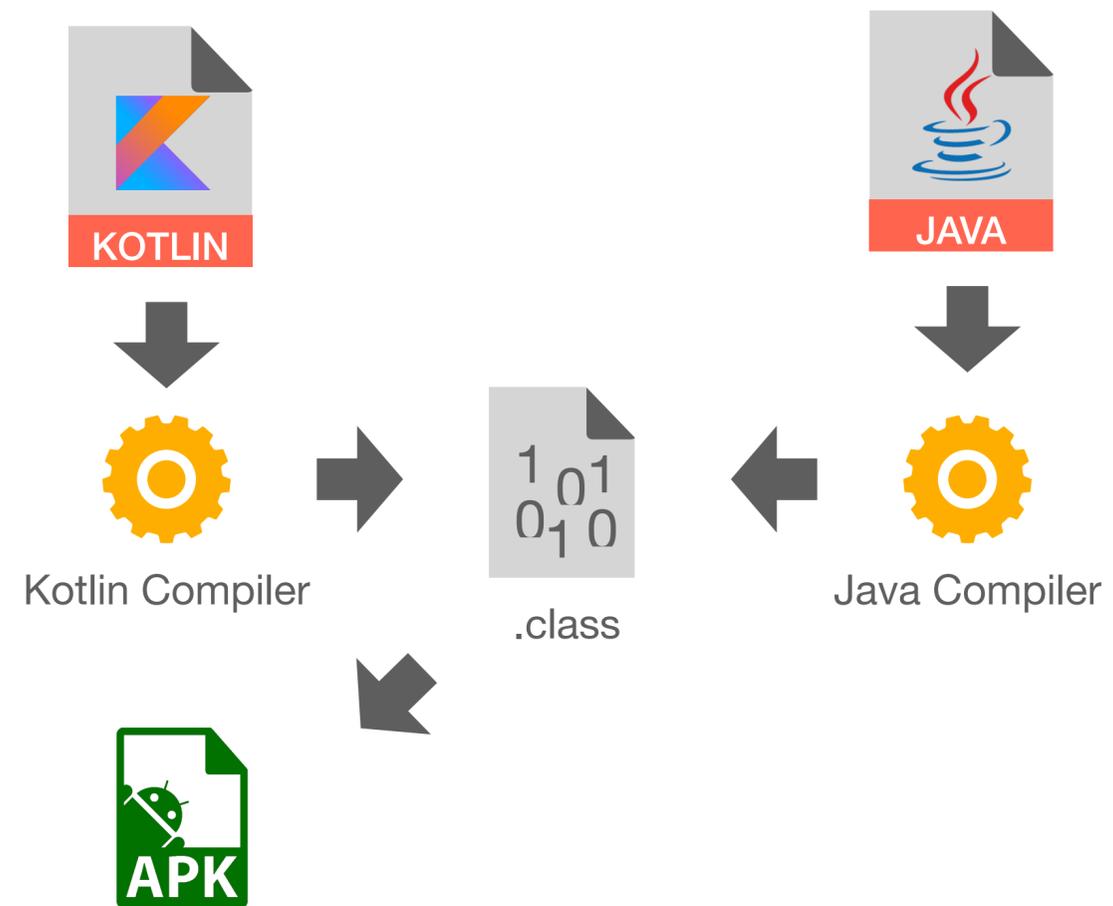
Interoperabilidade



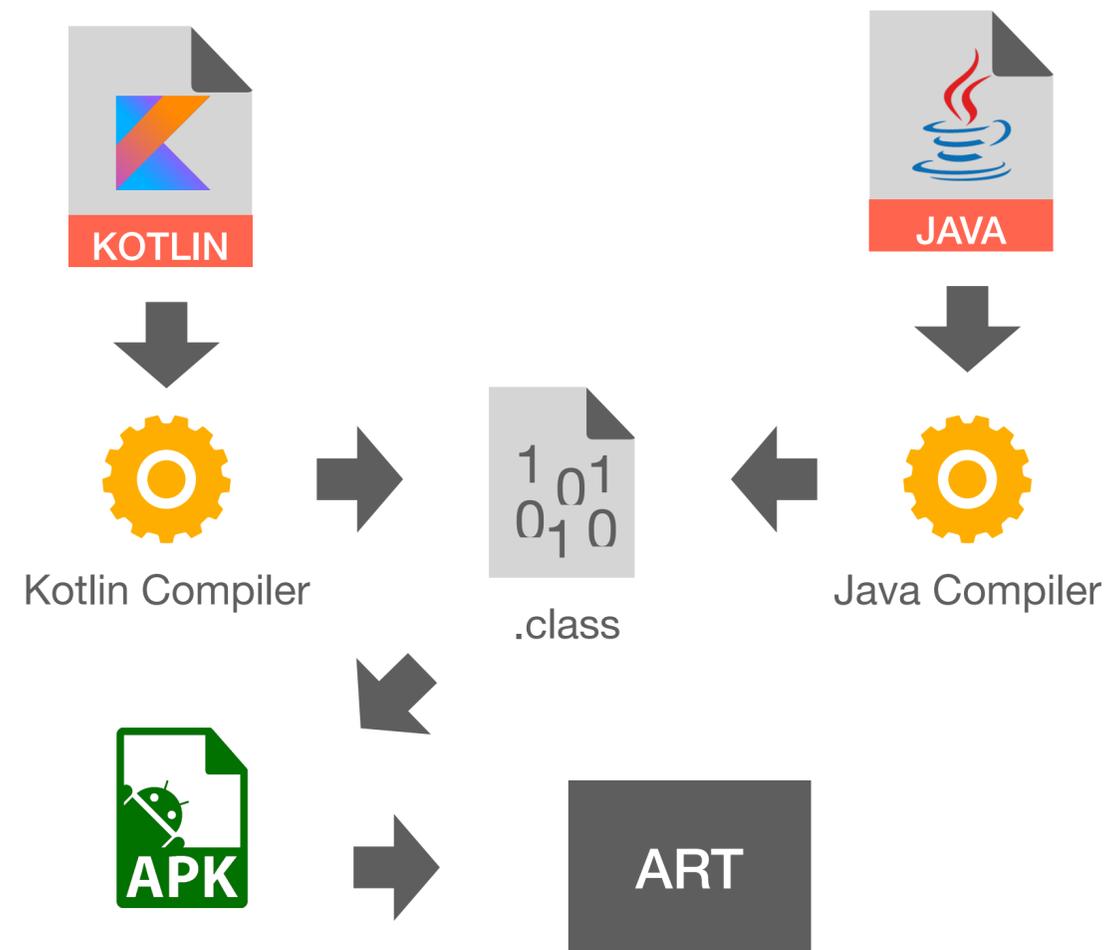
Interoperabilidade



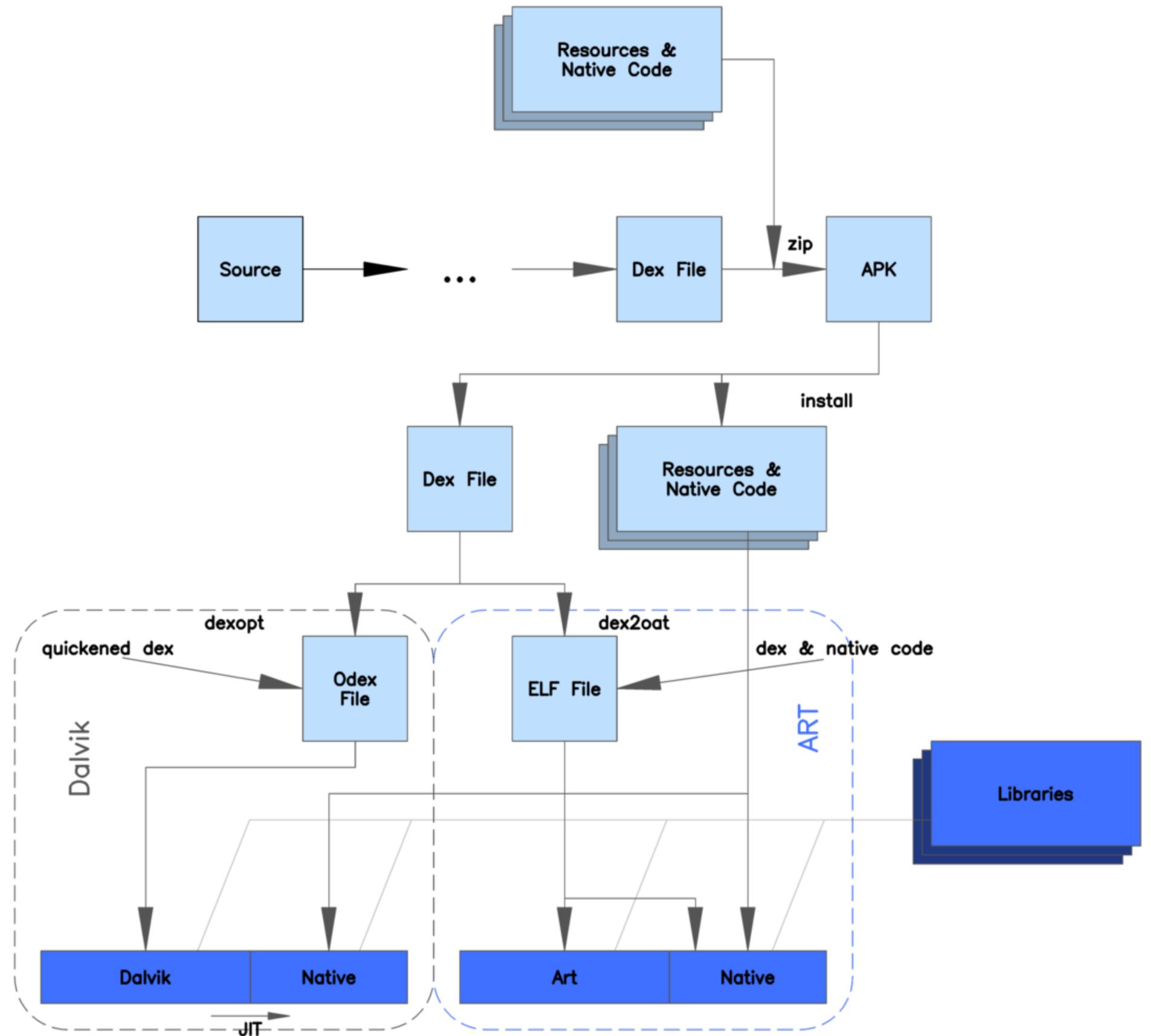
Interoperabilidade



Interoperabilidade

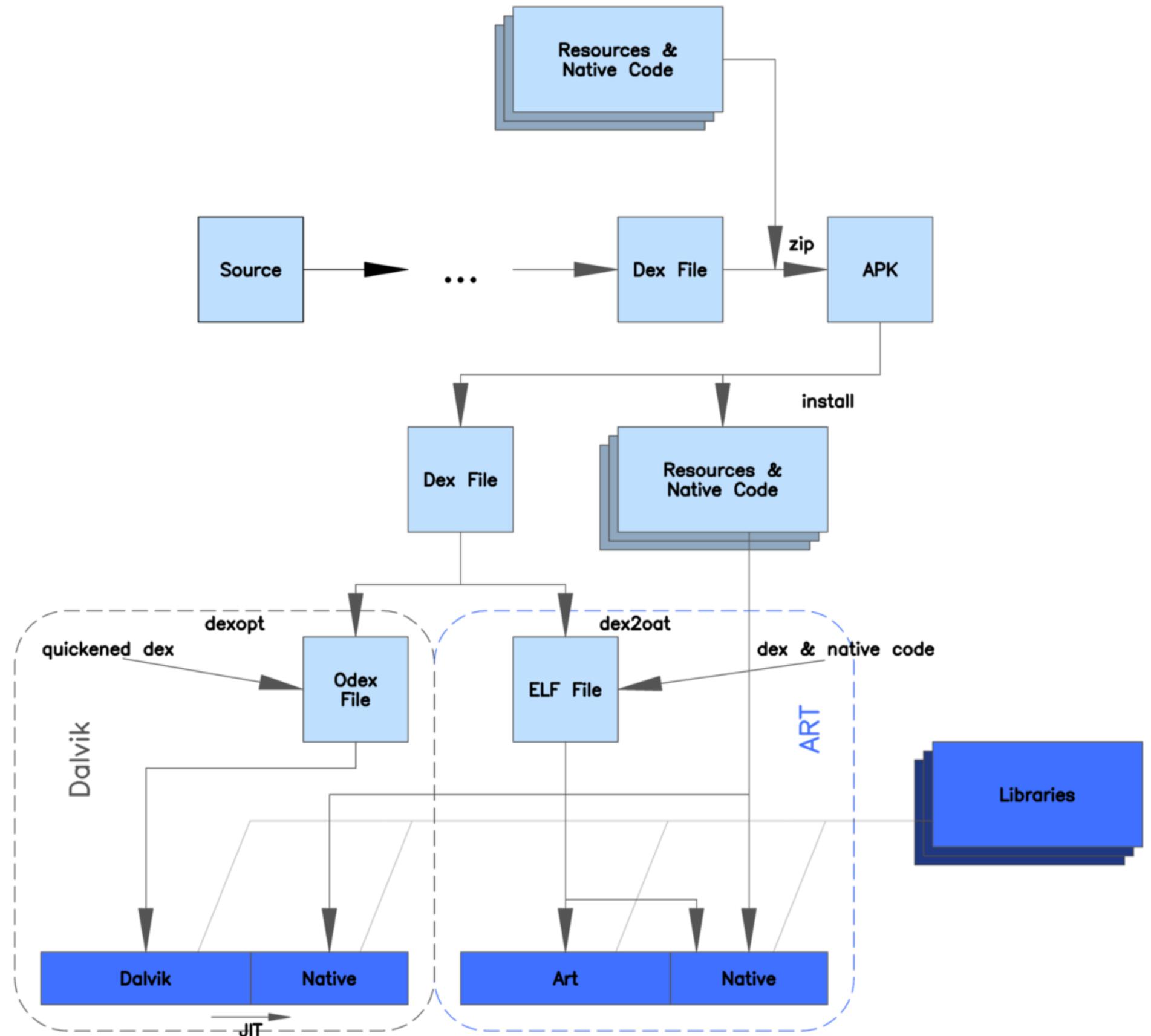


Mais detalhes



Mais detalhes

- <https://medium.com/programming-lite/android-core-jvm-dvm-art-jit-aot-855039a9a8fa>
- https://en.wikipedia.org/wiki/Android_Runtime
- <http://adbackstage.libsyn.com/episode-160-art-history>
- <https://www.youtube.com/watch?v=0J1bm585UCc>



Hands On

Tarefa de casa

Kotlin Koans

- Pulem os seguintes exercícios
 - Operator
 - For-loop
 - Invoke
 - Delegation works
 - Builders
 - Generics