



Nayana Farias
Pedro Endo

Scala

Scalable language

- Visão Geral
- Classes
- Case Classes
- Objetos
- Traits
- Abstract Classes
- Scala vs Java
- Aceitação pela comunidade

Visão Geral

- Linguagem de programação multi-paradigma feita para apresentar os padrões de programação mais comuns de forma concisa.
- Criada pelo suíço Martin Odersky.
 - *TypeSafe (Lightbend)*.
- Estaticamente tipada, tipos inferidos.

```
val x = 99  
val nome = "José"
```

Visão Geral

- Orientada a Objetos e Funcional

```
val isPar = (x : int) => x % 2 == 0
val numbers = List(1, 2, 3, 4, 5, 6, 7, 8, 9)

println(numbers.filter(isPar))
>2, 4, 6, 8

println(numbers.filter(!isPar))
>1, 3, 5, 7, 9
```

Classes

- Herança baseada em classes (organização de código)
- Métodos de classes não possuem retornos
- Não é necessário declarar return
- Possui construtor automático

```
class Logger(val context: String = "generic")
```

Case Classes

- Imutáveis por padrão
- Não precisam ser inicializadas com new
- Possuem um método cópia embutido (.copy())
- Geralmente usados para criação de tipos algébricos (ex: árvores) ou pattern matchings

Objects

- Tipo especial de classe
- Automaticamente singletons
- Todos os membros de um objeto são estáticos
- Podem herdar classes

```
object Folder {  
  def apply(name: String,  
            folder: Option[Folder] = None) = new File(name, folder)  
}
```

Traits

- Similar à interfaces java
- Podem implementar métodos
- Teoricamente devem ter apenas 1 responsabilidade
- NÃO possuem parâmetros de construtor

COMPOSITION OVER INHERITANCE

- Polimorfismo e reuso de código
- classes usam composição ao conter implementações de métodos de outros componentes ao invés de herança

```
trait Boldable {  
  def bold(text: String) = s"***$text**"  
  def unbold(text: String) = ???  
}  
  
trait Italicizable {  
  def italicize(text: String) = s"*$text*"  
  def unitalicize(text: String) = ???  
}  
  
object MarkdownWrapper extends Boldable with Italicizable {  
  def boldAndItalic(text: String) = bold(italicize(text))  
}  
  
MarkdownWrapper.wrapWithBoldAndItalic("Scala rocks!")  
// returns `***Scala rocks!***`
```


Abstract Classes

- Assim como Java, só pode ser herdada, nunca instanciada
- Podem possuir construtores e parâmetros
- Utilizada quando implementada uma classe Java dentro do Scala

```
abstract class BaseModel(db: Database, table: String) {  
  val id: Int // abstract property  
  val t = TableHelper(db, table)  
  
  def toJson: String // abstract method  
  
  def get = t.get(id)  
  
  def save: Option[Int] = t.save(toJson)  
  
  def update: Boolean = t.update(toJson)  
  
  def delete: Boolean = t.delete(id)  
}  
  
case class PostTemplate(title: String, body: String)  
  
class Post(id: Int,  
          template: PostTemplate) extends BaseModel(new Database, "posts") {  
  def toJson =  
    s"""{"id": $id, "title": "${template.title}", "body":  
    "${template.body}"}"""  
}  
  
val p = new Post(1, PostTemplate("Scala OOP Galore", "Scala is a hybrid..."))  
  
p.toJson  
  
// returns `{"id": 1, "title": "Scala OOP Galore", "body": "Scala is a  
hybrid..."}`
```

Scala

- Linguagem “complexa”
- 50% menos LOC que Java
- Roda em JVM
- Compatível e interoperável com Java
 - IDE's, ferramentas, bibliotecas
- Funções aninhadas e anônimas
- `var x val`

```
def productsByCategory(category: String) =  
  orders.flatMap(o => o.products)  
  .filter(p => p.category == category)
```

Java

- Linguagem simples
- Verboso
- Requer + LOC
- Encapsulamento

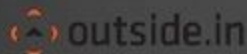
```
public List<Product> getProductsByCategory(String category) {  
    List<Product> products = new ArrayList<Product>();  
    for (Order order : orders) {  
        for (Product product : order.getProducts()) {  
            if (category.equals(product.getCategory())) {  
                products.add(product);  
            }  
        }  
    }  
    return products;  
}
```

Companies Using Scala



Novell

SIEMENS



amazon



Bank of America

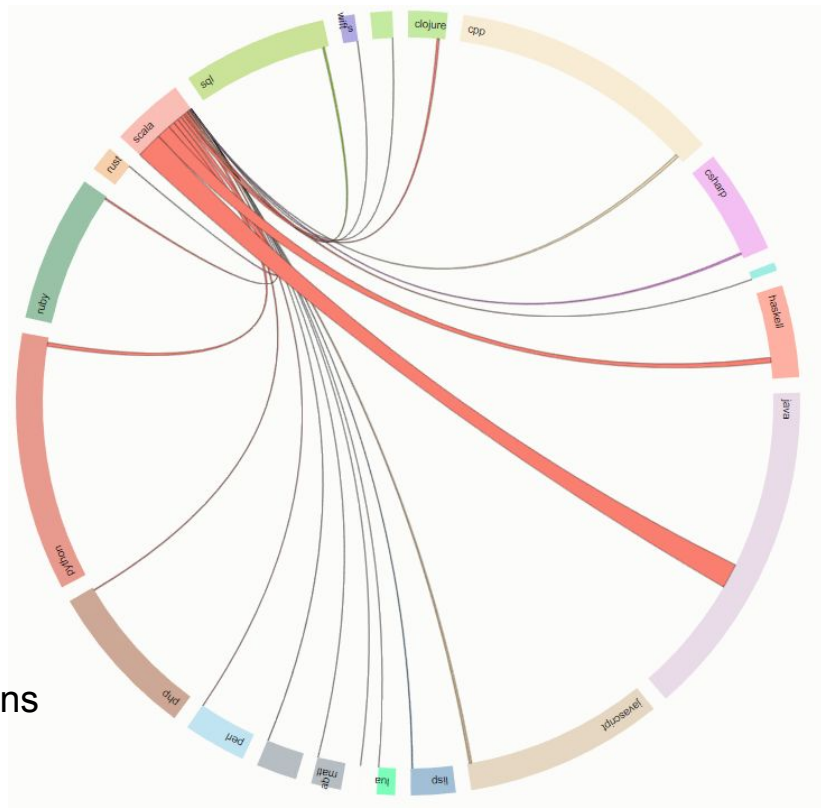
Yammer

Thatcham

Autodesk

CREDIT SUISSE

Scala e outras linguagens

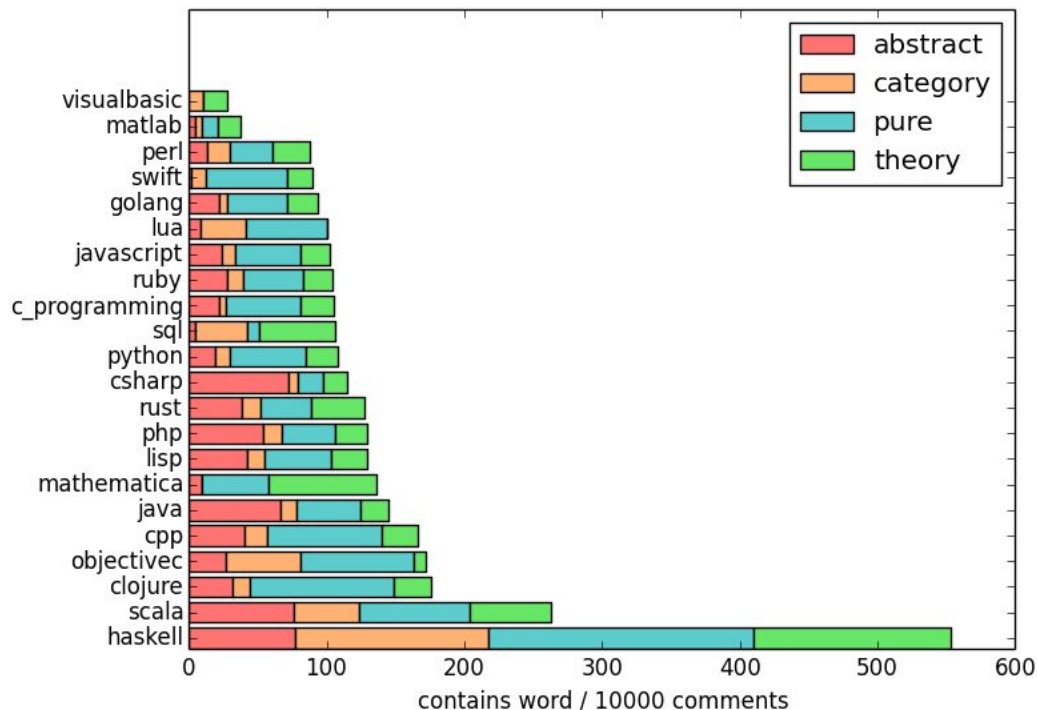


Associação entre linguagens

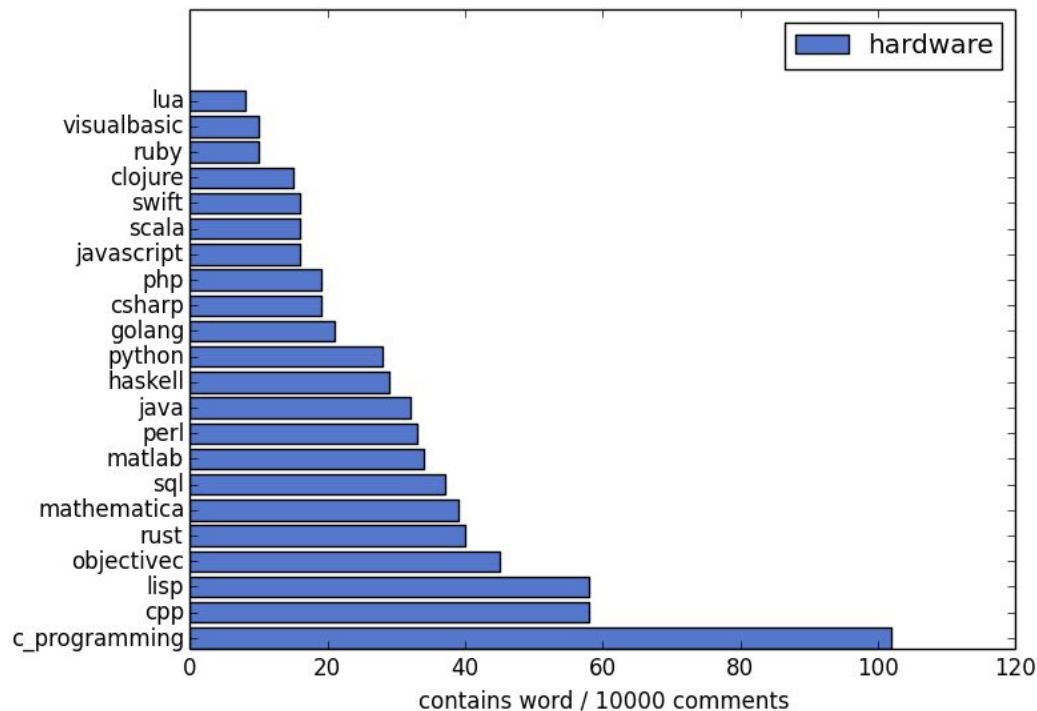
[Fonte](#) da imagem

by Tobias Hermann

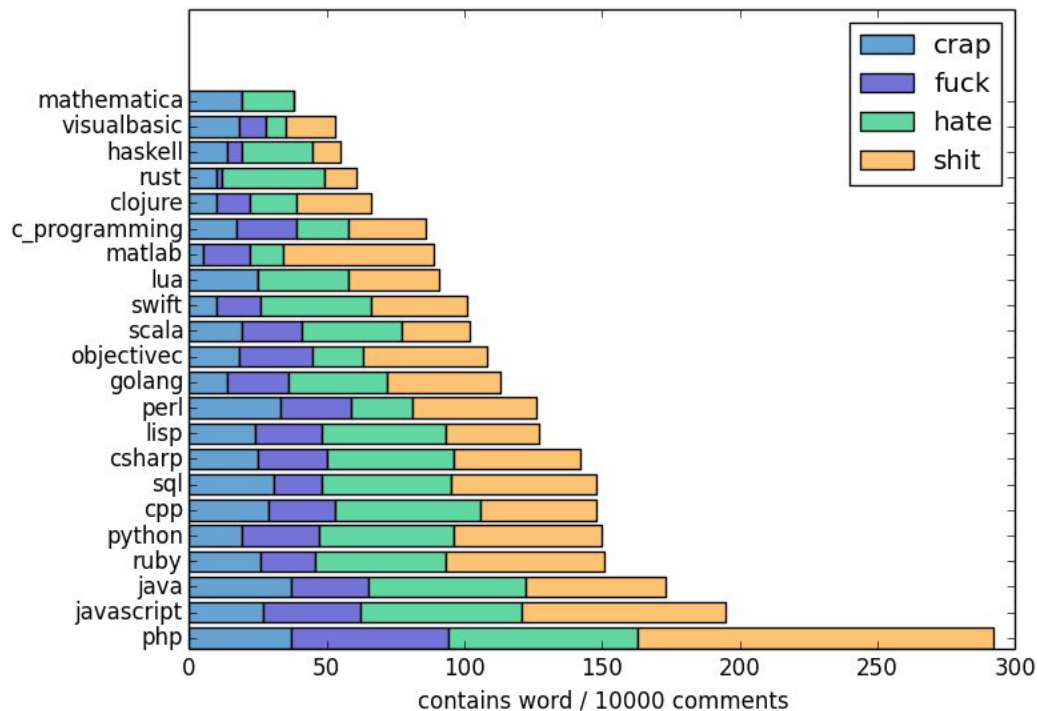
Scala e outras linguagens



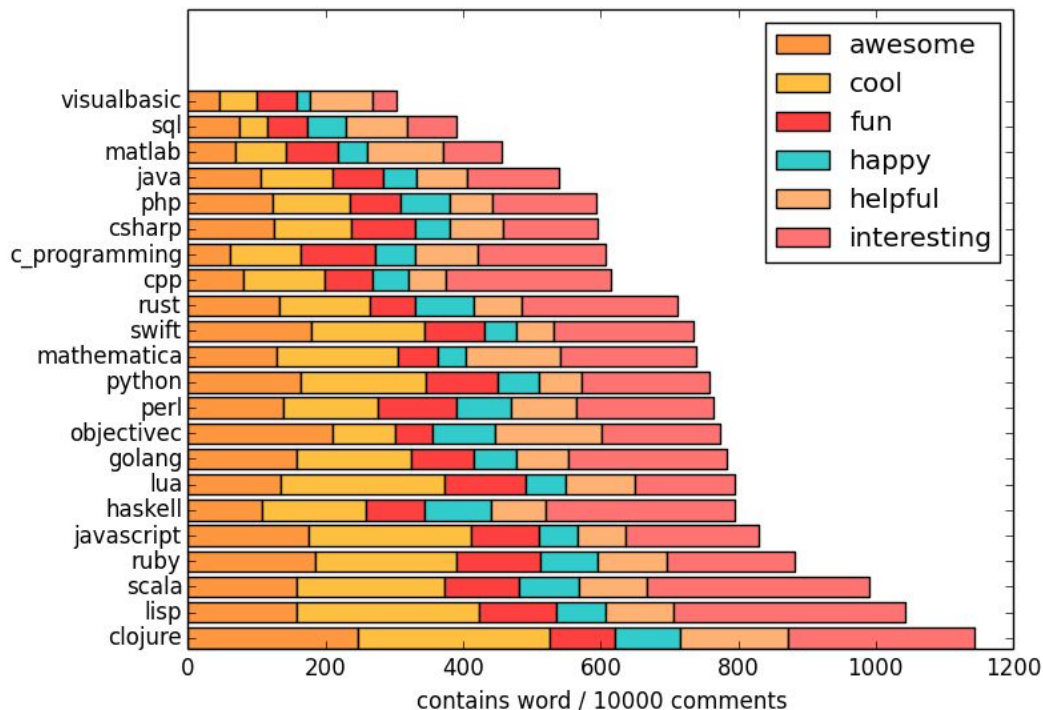
Scala e outras linguagens



Scala e outras linguagens



Scala e outras linguagens



Bibliografia

- [How to build a Scala REST CRUD application with Spray](#)
- [Tutorials Point - Scala](#)
- [What is Scala?](#)
- [Documentação oficial do Scala](#)
- [Artigo "Scala OOP Galore"](#)
- ["Linguagem de Programação Scala" na wikipedia](#)
- [Paradigma de programação funcional Scala](#)
- [Pattern matching em Scala](#)



Obrigado.