Trilha Java

THE DEVELOPER'S CONFERENCE

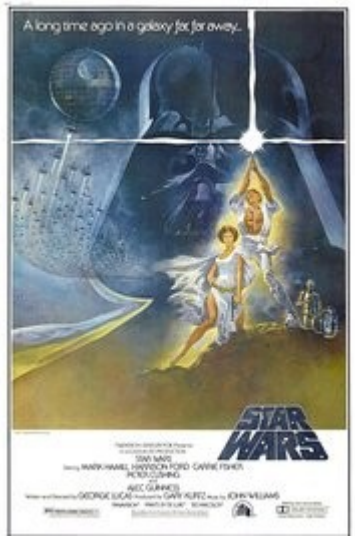
# Aprendendo Java 8 Lambdas com Star Wars

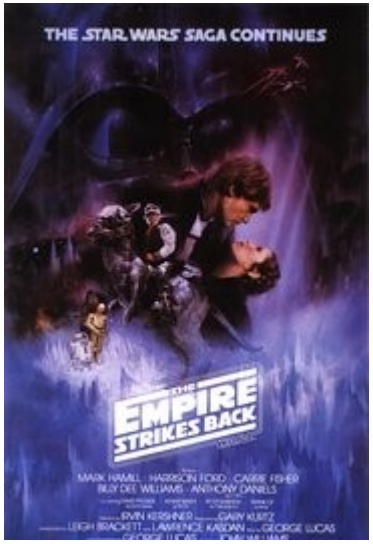
Globalcode

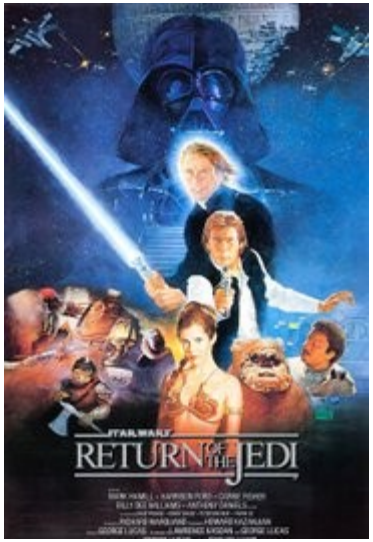
# Ortolan!!!

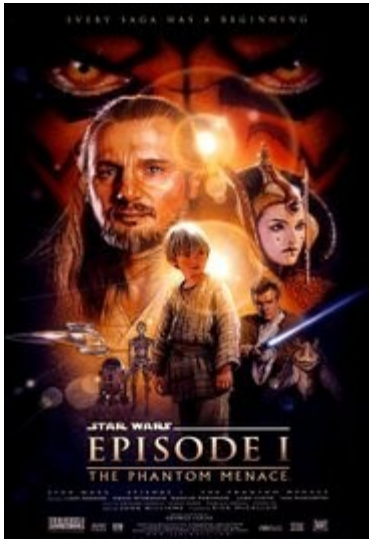
# Star Wars

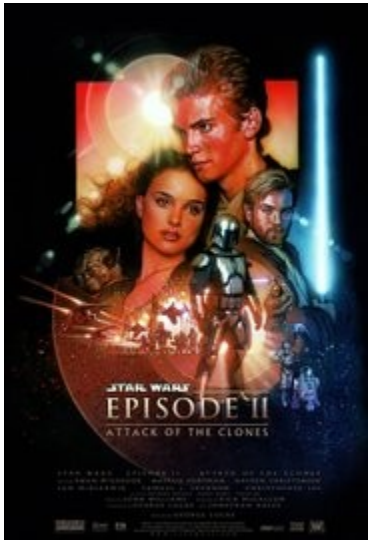
THE DEVELOPER'S CONFERENCE

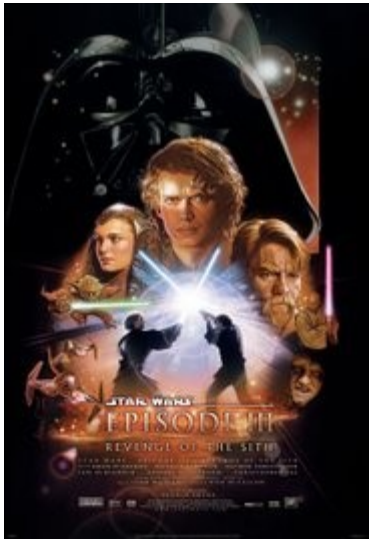


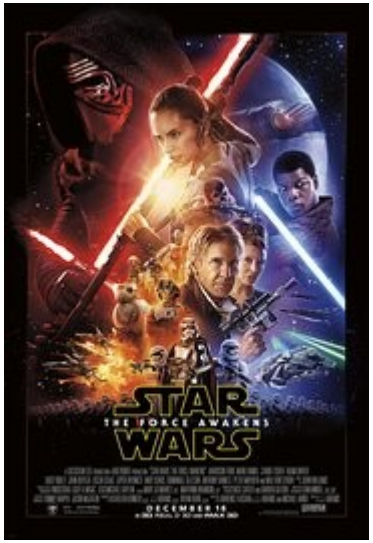



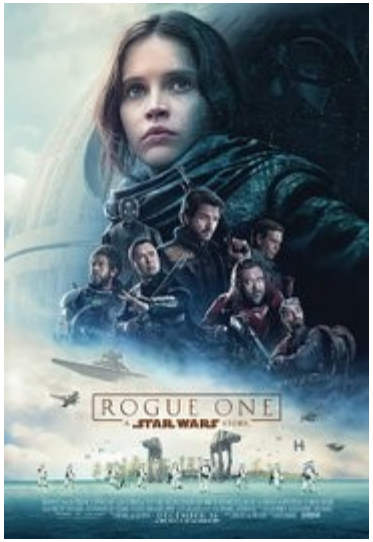




Globalcode

# JSR 335

THE DEVELOPER'S CONFERENCE

- JSR 335
  - Java 8 Lambdas
  - Métodos default nas interfaces
  - Métodos estáticos nas interfaces
  - Expressões Lambdas
  - Streams
  - Filter
  - Map
  - Sorted
  - Collectors

Globalcode

# Lambdas

THE DEVELOPER'S CONFERENCE

- Lambdas
  - Linguagem funcional
    - Funções Anônimas
  - Java
    - @FunctionalInterface
    - Um método
    - 0..n parâmetros
    - Runnable
    - ActionListener
    - Comparator
    - Sua própria interface

Globalcode

# Filter

THE DEVELOPER'S CONFERENCE

- .filter
  - Selecionar Dados
  - Predicate<T>
  - .test(T): boolean
  - .and
  - .or
  - .negate
  - .isEqual

Selecionar

- Dirigidos Por Irvin Kershner
- Na década de 80
- Com Luke Skywalker

Globalcode

# Predicate

THE DEVELOPER'S CONFERENCE

- Dirigidos por Irvin Kershner
  - Usar uma expressão Lambda
    - **m → m.getDirector().equals("Irvin Kershner")**
    - m = Instância de StarWarsMovie
    - m.getDirector().equals("Irvin Kershner") = implementação do meu Predicate!

Globalcode

# Predicate

THE DEVELOPER'S CONFERENCE

- Na década de 80
  - Implementação de um Predicate
  - m = Instancia de StarWarsMovie
  - getReleaseDate(): LocalDate

```
(m.getReleaseDate().isEqual(startingDate) ||
m.getReleaseDate().isAfter(startingDate))
 && (m.getReleaseDate().isEqual(finishinDate) ||
     m.getReleaseDate().isBefore(finishinDate));
```

Globalcode

# Predicate

THE DEVELOPER'S CONFERENCE

- Tem Luke Skywalker
  - Retornar uma expressão lambda
  - m → Instância de StarWarsMovie
  - getMainCharacters(): List<StarWarsCharacters>
  - contains: boolean

```
m -> m.getMainCharacters().contains(character);
```

Globalcode

# Predicate

THE DEVELOPER'S CONFERENCE

```
movies
        .stream()
        .filter(m -> m.getDirector().equals("Irvin Kershner"))
        .filter(lukeSkywalker())
        .filter(inThe80s)
        .collect(Collectors.toList())
        .forEach(m -> System.out.println(m.getTitle()));
```

```
Find movies directed by Irvin Kershner with Luke SkyWalker in the 80's
------------------------------------------------------------------------
The Empire Strikes Back
```

Globalcode

# Map

THE DEVELOPER'S CONFERENCE

- .map
  - Transformar Dados
  - Function<T, R>
  - .apply(T):R
  - .compose
  - .andThen
  - .identity

Quero uma listagem de:

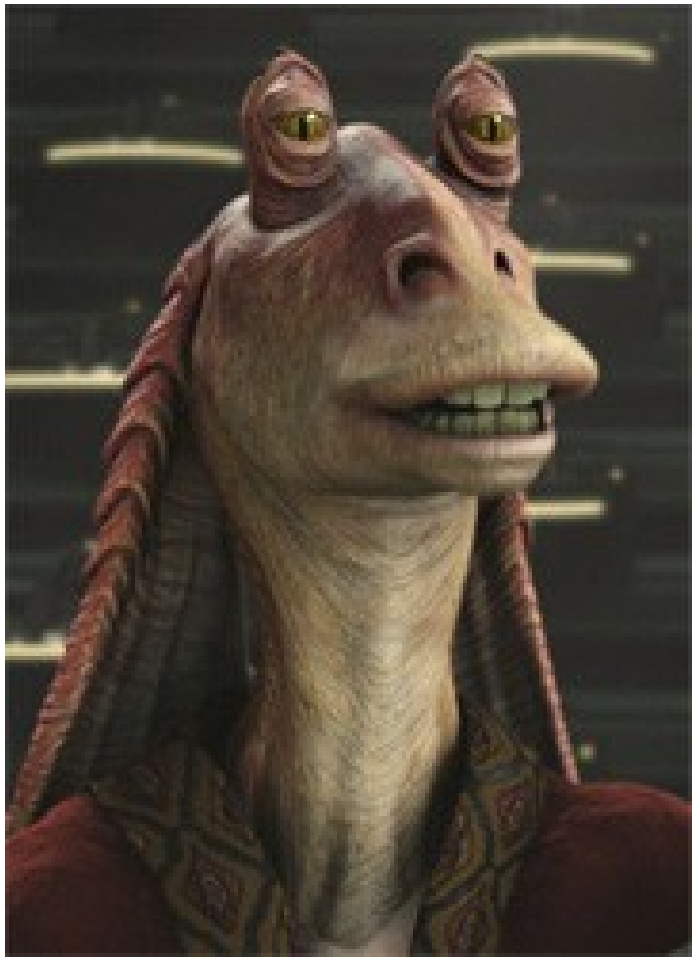
Todos os títulos

Todos os diretores

Todos os personagens

Globalcode

# Aviso



**CONSELHO JEDI ADVERTE:**

**Listar todos os personagens pode conter Jar Jar Binks**

# Function

THE DEVELOPER'S CONFERENCE

- Todos os Títulos
  - Implementar uma Function

```
public String apply(StarWarsMovie starWarsMovie) {
    return starWarsMovie.getTitle();
}
```

Globalcode

# Function

THE DEVELOPER'S CONFERENCE

- Todos os diretores
  - Usar expressão Lambda
  - **m → m.getDirector()**
  - **distinct() → remove duplicados**

```
movies
        .stream()
        .map(m -> m.getDirector())
        .distinct()
        .collect(Collectors.toList());
```

Globalcode

# Function

THE DEVELOPER'S CONFERENCE

- Todos os personagens
  - **flatmap**
  - Extrair um Stream dentro de um Stream

```
return movies
        .stream()
        .map(l -> l.getMainCharacters())
        .flatMap(List::stream)
        .distinct()
        .collect(Collectors.toList());
```

Globalcode

# Sorted

THE DEVELOPER'S CONFERENCE

- .sorted
  - Ordenar Dados
  - Comparator<T>
  - .compare(T, T): int
  - .reversed
  - .thenComparing
  - .comparing
  - .comparingInt|Long
  - .comparingDouble

Ordernar filmes:

- Ordem de Lançamento
- Ordem Cronológica
- Ordem Pessoal

Globalcode

# Comparator

THE DEVELOPER'S CONFERENCE

- Ordem de Lançamento
  - Usar comparing
  - Deve implementar Comparable!

```
Comparator.comparing(
    StarWarsMovie::getReleaseDate
)
```

```
Release Order
-------------
A New Hope
The Empire Strikes Back
The Return of the Jedi
The Phantom Manace
The Attack of the Clones
The Revenge of the Sith
The Force Awakens
Rogue One
```

Globalcode

# Comparator

THE DEVELOPER'S CONFERENCE

- Ordem Cronológica
  - Usar expressão Lambda
  - **(m1, m2) → parâmetros do Método compare**
  - **m1 e m2 → Instâncias de StarWarsMovie**
  - **m1.getChronologicalOrder() - m2.getChronologicalOrder()**

```
(m1, m2) ->
    m1.getChronologicalOrder() -
    m2.getChronologicalOrder()
```

```
Chronological Order
-------------------
The Phantom Manace
The Attack of the Clones
The Revenge of the Sith
Rogue One
A New Hope
The Empire Strikes Back
The Return of the Jedi
The Force Awakens
```

Globalcode

# Comparator

THE DEVELOPER'S CONFERENCE

- Ordem pessoal
  - Usar comparingInt

```
Comparator.comparingInt(
    StarWarsMovie::getPersonalOrder
)
```

```
Personal Order
--------------
The Empire Strikes Back
Rogue One
The Return of the Jedi
The Force Awakens
A New Hope
The Revenge of the Sith
The Phantom Manace
The Attack of the Clones
```

Globalcode

# Conclusões

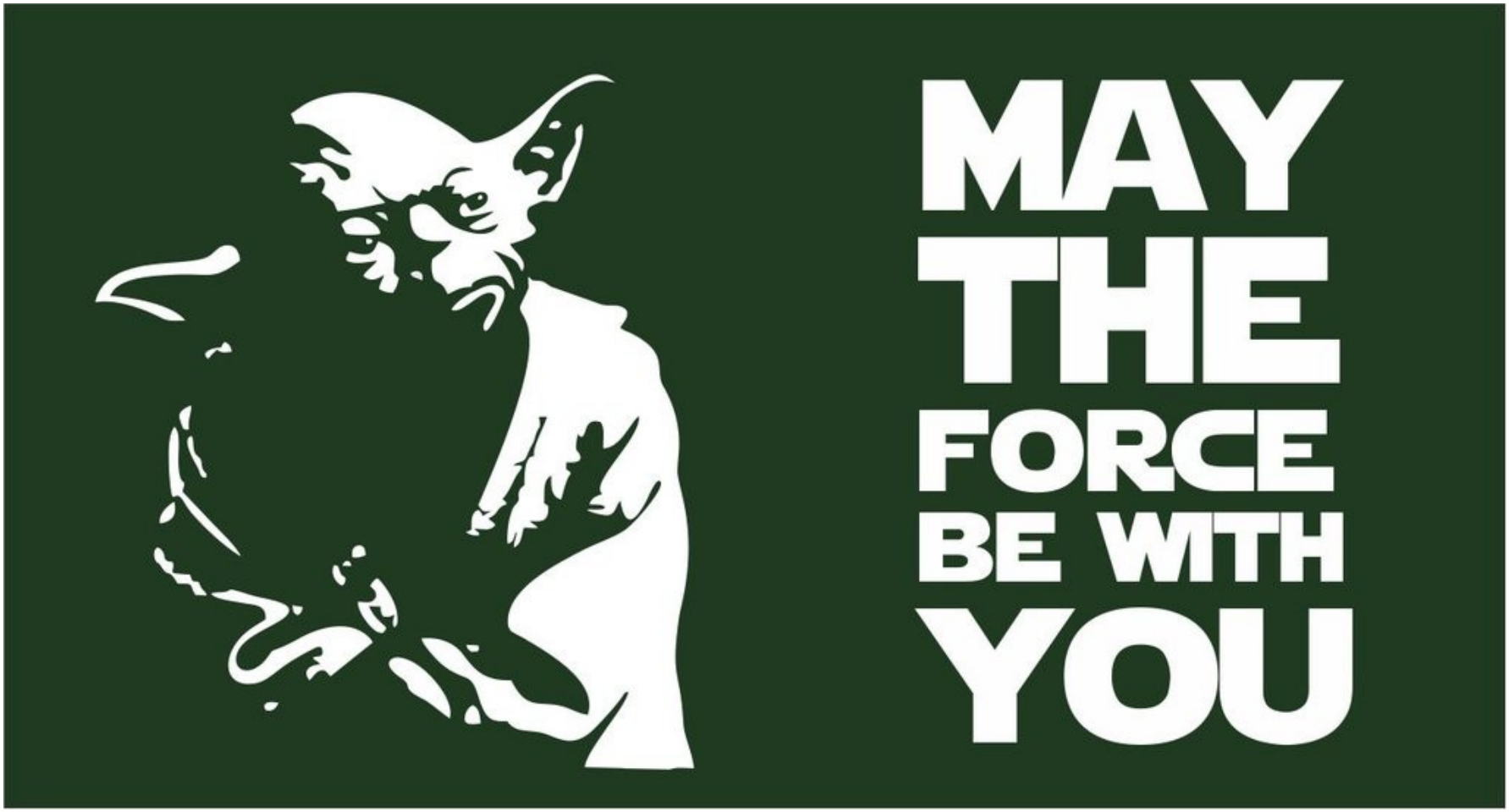
THE DEVELOPER'S CONFERENCE

- Enfim,
  - Java é uma linguagem verbosa
    - Java 8 Lambdas tira isso
    - Depende do programador também
  - Aprendizado
    - Não se assute, parece difícil
    - "I am one with the Lambda, and the Lambda is with me"
  - Complexidade
    - Retira complexidade ciclomática do código
    - Pensar em alternativas para condicionais e loops

Globalcode

# May the Force be With You

THE DEVELOPER'S CONFERENCE

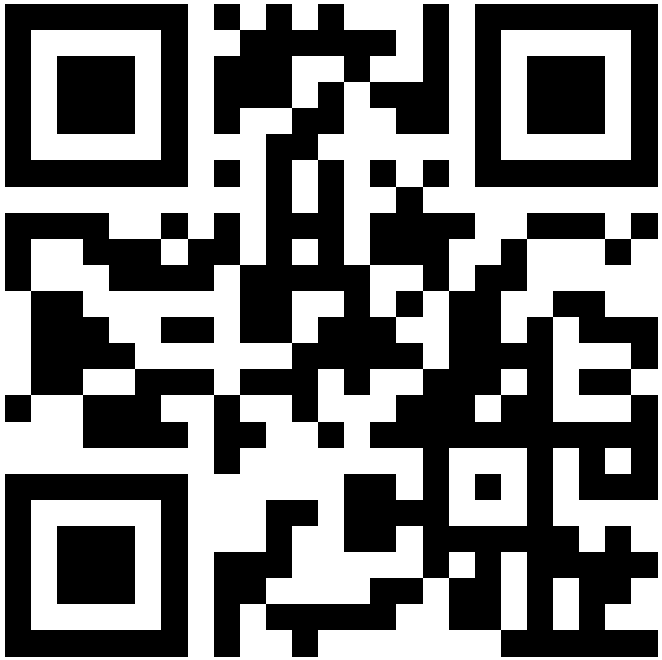



Globalcode

# Paulo Henrique Ortolan

E-mail: paulo.ortolan@gmail.com

Twitter: @pauloortolan



THE DEVELOPER'S CONFERENCE

Globalcode