

# Paradigmas de Linguagens Computacionais

Samuel Paz Mendes ( spm )

Marlon Reghert Alves dos Santos ( mras )

Tomer de Lima Simis ( tls )

# Perguntas a serem respondidas

1. O que é um paradigma de linguagem de programação?
2. Quais são as características fundamentais do paradigma de programação funcional? Como ele se diferencia da programação imperativa?
3. Quais são as vantagens do paradigma funcional, em contrapartida com o paradigma imperativo ?
4. Quais são as origens da programação funcional?
5. Que grandes empresas utilizam esse paradigma? Que tipo de sistema é desenvolvido usando linguagens funcionais?

# 1) Paradigmas de linguagem de programação

- Um paradigma de linguagem de programação é um conceito que determina a visão que o programador tem sobre a estruturação e execução de um programa.
- Diferem nos conceitos e abstrações utilizados para representar os elementos de um programa (variáveis, fluxo de dados, restrições, funções/procedimentos )

# 1) Paradigmas de linguagem de programação

- Linguagens de programação possuem paradigmas diferentes, diferenciando-as quanto a forma como esta é utilizada e os problemas nos quais ela mais se adapta
- Teoricamente as linguagens devem possuir o mesmo poder computacional, porém os paradigmas relacionados à estas acabam às tornando mais conveniente para resolução de determinados problemas
- Ex: Imperativo, Funcional, O. Objeto, Descritivo, O. Aspectos.

## 2) Paradigma de programação funcional

- Foge da ideia do programa como um estado
- A computação é vista como a avaliação de um conjunto de funções matemáticas, evitando estados e dados mutáveis.
- Funções vistas como valores de 1ª ordem (valores de entrada retorno, parâmetro, etc.)
- É um paradigma declarativo.

## 2) Programação funcional

- Mapeamento de valores de entrada em valores de retorno.
- Sem alocamento explícito de memória ou declaração explícita de variáveis
- Funções não alteram ou modificam o resultado de outras funções ( assegura que uma função computa o mesmo resultado para um mesmo conjunto de parâmetros )
- Ex: Haskell, Perl 6, LISP, etc.

### 3) Paradigma declarativo X imperativo

- Estado
  - Funcional: Não mantem estado
  - Imperativo: Mantém estado
- Definição da função
  - Funcional: retorno imutável para uma entrada y
  - Imperativo: retorno pode variar de retorno para uma entrada y e funções sem retorno

### 3) Paradigma declarativo X imperativo

- Computação
  - Funcional: Avaliação de funções
  - Imperativo: Sequência de instruções que modificam valores de variáveis
- Alocação de memória explícita
  - Funcional: Não há alocação explícita
  - Imperativo: Há alocação explícita



### 3) Paradigma declarativo X imperativo

- Declaração de variáveis
  - Funcional: Não declara
  - Imperativo: Declaração explícita
- Modelo de computação no qual é baseado:
  - Funcional: Lambda Calculus
  - Imperativo: Máquina de Turing

### 3) Paradigma funcional X imperativo

- Vantagens do funcional
  - Facilidade para escrever código concorrente usado em processadores de multicore.
  - Alto nível de abstração.
  - Ausência de operações de atribuição tornam a programação funcional muito mais simples para provas e análises matemáticas.

### 3)Paradigma funcional X imperativo

- Desvantagens do funcional

Problemas que envolvem muitas variáveis ou muitas atividades sequenciais (Ex: contas de banco) o podem ser mais complexos de serem resolvidos em linguagens funcionais

- Eficiência

### 3)Paradigma funcional X imperativo

- Vantagens Imperativo
  - Eficientes, geralmente possibilitam acesso direto à memória fazendo com que os ambientes que executam linguagens imperativas possuam boa performance.
  - Modela “naturalmente” a solução de um problema.

### 3)Paradigma funcional X imperativo

- Desvantagens Imperativo
  - Código de difícil reuso
  - Manutenção sujeita à criação de novos bugs
  - Propenso à ter difícil legibilidade

## 4) Origens

- Alonzo Church faz uso dos sistemas de Calculo Lambda para criar uma definição de Algoritmo, as linguagens funcionais se baseiam neste modelo de computação.
- A lógica combinatória é, frequentemente, notada como mais abstrata que o cálculo lambda e como precedente na invenção.

## 5) Empresas

- A programação funcional é mais popular na academia, porém recentemente diversas empresas vem se utilizando de linguagens funcionais para o desenvolvimento de suas aplicações, exemplos:
- Linguagem *Erlang* (Criada pela Ericsson): Utilizada para implementar os tratamentos para tolerância a falhas em sistemas de Telecomunicações, até hoje utilizada e aderida por empresas como: Facebook, Whatsapp, T-Mobile, Nortel, etc.

## 5) Empresas

- *OCalm* introduzida em 1990: Muito utilizada em aplicações comerciais de análise financeira.
- Tendência: Scala, linguagem funcional
  - Em 2009 o Twitter informa que vem migrando boa parte de seu backend para Scala.
  - Foursquare e LinkedIn utilizam Scala.
- Linspire (S.O baseado em GNU/Linux ) utilizou Haskell para o desenvolvimento das ferramentas do sistema.



## 5) Empresas

- Web frameworks em Haskell:
  - Yesod
  - Happstack
  - Snap

# Dúvidas

