PUC Minas

HASKELL

Caio Faria Diniz
Giuseppe Senna Cordeiro
Vinicius Miranda de Araújo

PUC Minas

SUMÁRIO

- Introdução;
- Cronologia;
- Paradigmas;
- Características Principais;
- Linguagem relacionadas;
- Desenvolvimento.

INTRODUÇÃO

PUC Minas

- O que é haskell?
  - Linguagem de programação funcional;
  - Tipagem forte e estática;
  - Baseada no cálculo lambda;
  - Foco em "o que fazer", não "como fazer";
  - Focada em produtividade, clareza e manutenibilidade;
  - Linguagem funcional mais pesquisada atualmente.

LINHA DO TEMPO

PUC Minas

- Influências:
  - **1930**: Alonzo Church desenvolveu o cálculo de lambda.
  - **1950**: John McCarthy, influenciado pela teoria do lambda, desenvolveu Lisp, a primeira linguagem funcional.
  - **1970**: ML, a primeira linguagem funcional moderna introduzindo a inferência de tipos e tipos polimórficos na linguagem.

LINHA DO TEMPO

PUC Minas

1987

Comunidade de programação funcional decidiu criar uma linguagem em um Conferência

1990

Primeira versão de Haskell foi definida

1999

Haskell 98, especificando uma versão mínima, estável e portável da linguagem

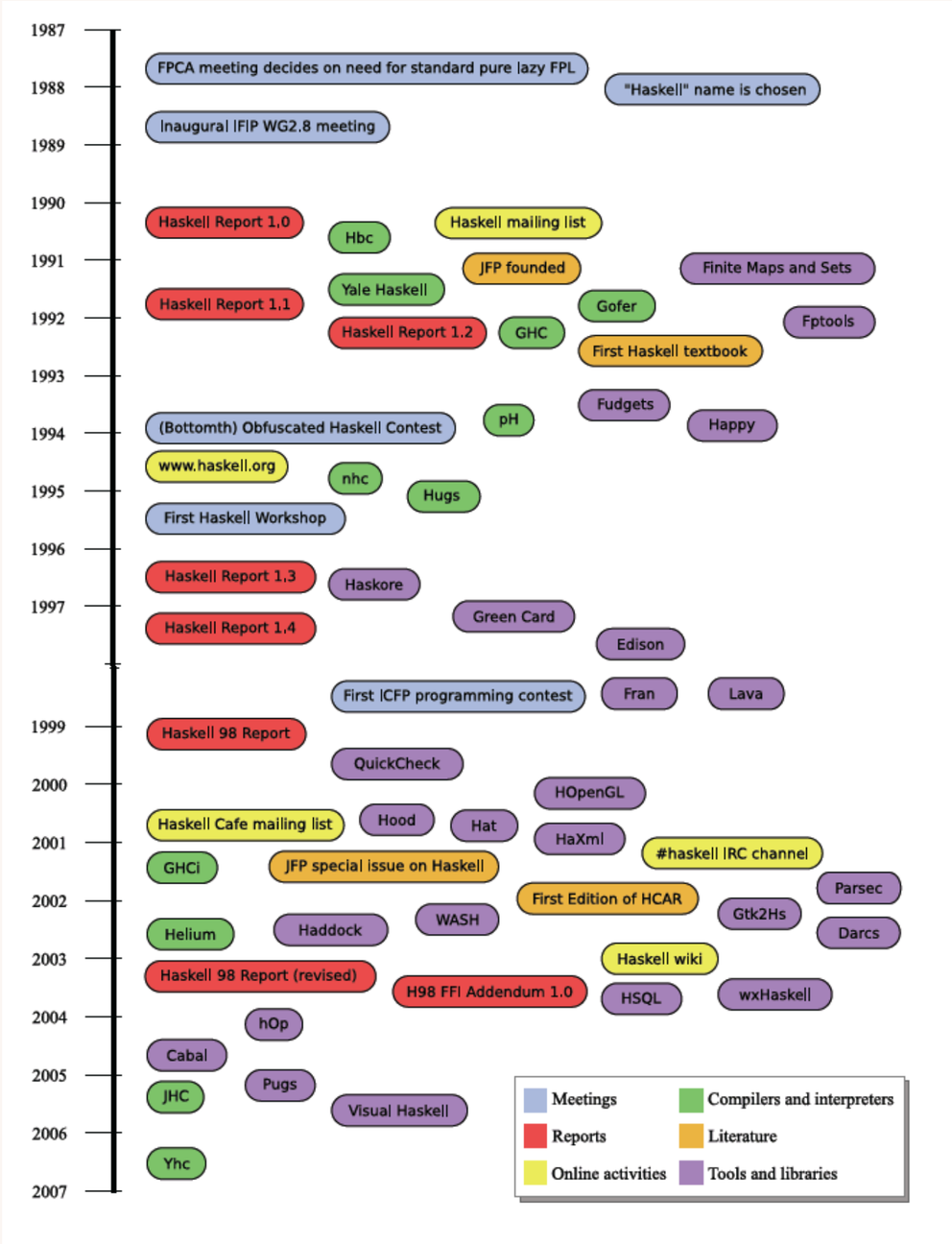
2003

O padrão Haskell 98 foi revisado

2006

Começou o processo de definição de um sucessor do padrão 98, conhecido informalmente por Haskell′ ("Haskell Prime")

LINHA DO TEMPO: INFOGRÁFICO



PARADIGMA

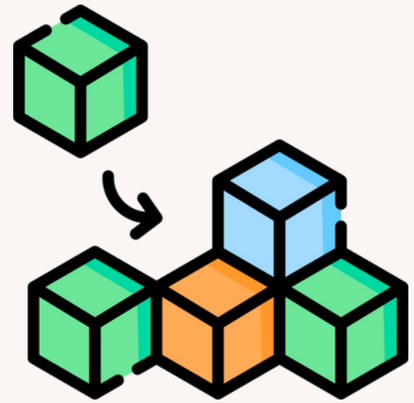
PUC Minas

- Funcional:
  - Sem atribuições;
    - ≠ Linguagens imperativas;
  - Funções como elemento central;
    - Código mais seguro;
  - Lazy Evaluation (Avaliação preguiçosa);
  - High-order Functions (Funções de ordem superior).

PARADIGMA

PUC Minas

- Modular:
  - Modularização natural;
  - Sem regras de escopo rígida;
  - Focado na criação de funções;
  - Permite controle de exportação e privatização;
  - Facilita a reutilização, organização e manutenção do código;



CARACTERÍSTICAS PRINCIPAIS

PUC Minas

- Avaliação Preguiçosa
- Polimorfismo Universal
- Função de Ordem Superior
- Estruturas de Dados de Tamanho Infinito

CARACTERÍSTICAS PRINCIPAIS

PUC Minas

Função de Ordem Superior

caracteristicas.hs

```haskell
-- Função que aplica uma função a cada elemento da lista
dobros :: [Int] -> [Int]
dobros xs = map (*2) xs

main = print (dobros [1, 2, 3]) -- Retorna [2, 4, 6]

```

Polimorfismo Universal

caracteristicas.hs

```haskell
-- Função polimórfica que retorna o valor que recebe
id :: a -> a
id x = x

-- Exemplos de uso
main = do
    print (id 5)        -- Retorna 5
    print (id "Olá")    -- Retorna "Olá"
    print (id [1, 2, 3]) -- Retorna [1, 2, 3]

```

Estruturas de Dados de Tamanho Infinito

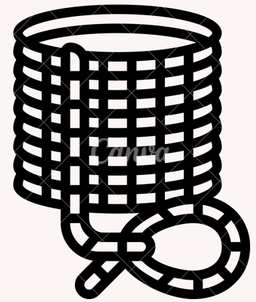
caracteristicas.hs

```haskell
-- Definindo uma lista infinita de números naturais
numerosNaturais :: [Int]
numerosNaturais = [0..]

main = print (take 5 numerosNaturais) -- Retorna [0, 1, 2, 3, 4]

```

AMARRAÇÕES

PUC Minas

- Tipos estático:
  - Todos os tipos são conhecidos em tempo de compilação;



- Inferência de tipos:
  - Não é necessário explicitar o tipo do identificador;

- Um mesmo identificador pode ser declarado em diferentes partes do programa.

VALORES: TIPOS PRIMITIVOS

- Numérico:
  - Inteiros:
    - **Integer**, **Int**;
  - Reais:
    - **Double**, **Float**;
- Unitário: **Unit**, implementação do conjunto 1;
- Lógico: **Bool**;
- Caractere: **Char**;

tipos.hs

```haskell
-- Inteiros
a :: Integer
a = 10 -- Tamanho decidido pela máquina

b :: Integer
b = 1234567890987654321 -- Tamanho arbitrário

-- Reais
c :: Double
c = 3.1415926

d :: Float
d = 3.14

-- Booleanos
e :: Bool
e = True
e' = False

-- Caracteres
f :: Char
f = 'a'

-- Strings
g :: String
g = "Hello, World!"
```

VALORES: TIPOS COMPOSTOS

PUC Minas

- Tuplas:
  - É a implementação do **produto cartesiano**;
- Listas:
  - Conjunto finito de dados;
  - Podem ser definidas por **compreensão.**
- Strings: Lista de caracteres.

tipos.hs

```haskell
-- Tupla
cliente :: ( String, Int, Char ) -- ( Nome, Idade, Sexo )
cliente = ( "João", 25, 'M' )

produto :: ( String, Float ) -- ( Nome, Preço )
produto = ( "Notebook", 2500.00 )

teste :: ( Int, Float, Char, String, [ Int ] )
teste = ( 1, 2.5, 'A', "Teste", [ 1, 2, 3 ] )

-- Lista
inteiros :: [ Int ]
inteiros = [ 1, 2, 3, 4, 5 ]

clientes :: [ ( String, Int, Char ) ]
clientes = [ ( "João", 25, 'M' ), ( "Maria", 30, 'F' ) ]

lambda :: [ n * n Z | n <- [ 1 .. 10 ] ]

```

FUNÇÕES

- Função Pura;
- Função de Ordem Superior;
- Função Parcial;
- Função Curried;
- Função Lambda;
- Função Recursiva;

PUC Minas

funcoes.hs

```haskell
-- Função Pura
soma :: Int -> Int -> Int
soma x y = x + y

-- Função de Ordem Superior
aplicaDuasVezes :: (a -> a) -> a -> a
aplicaDuasVezes f x = f (f x)

-- Função Parcial
somaParcial :: Int -> Int
somaParcial = soma 5  -- Função que soma 5 a um número

-- Função Curried
somaCurried :: Int -> Int -> Int
somaCurried x y = x + y
-- Equivalente a: somaCurried x = (\y -> x + y)

-- Função Lambda
(\x -> x * 2) 5  -- Retorna 10

-- Função Recursiva
fatorial :: Int -> Int
fatorial 0 = 1
fatorial n = n * fatorial (n - 1)

```

FUNÇÕES

- Função Composta;
- Função Parcialmente Definida;
- Função Monádica;
- Função com Parâmetro Polimórfico;

funcoes.hs

```haskell
-- Função Composta
funcaoComposta :: (b -> c) -> (a -> b) -> (a -> c)
funcaoComposta f g = f . g

-- Função Parcialmente Definida
head :: [a] -> a
head (x:_) = x
head [] = error "Lista vazia"

-- Função Monádica
leituraNome :: IO String
leituraNome = do
    putStrLn "Digite seu nome:"
    getLine

-- Função com Parêmetro Polimórfico
id :: a -> a
id x = x

```

LINGUAGENS RELCIONADAS

PUC Minas

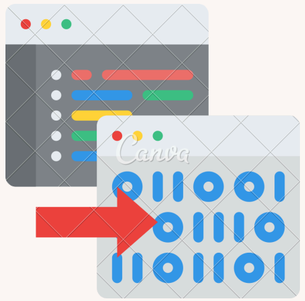
- Linguagens que Influenciaram:
  - **Miranda**;
  - **Standard ML (SML)**;

- Linguagens que foram Influenciadas:
  - **C#**;
  - **Python**;
  - **Scala**;

COMPILADOR

PUC Minas

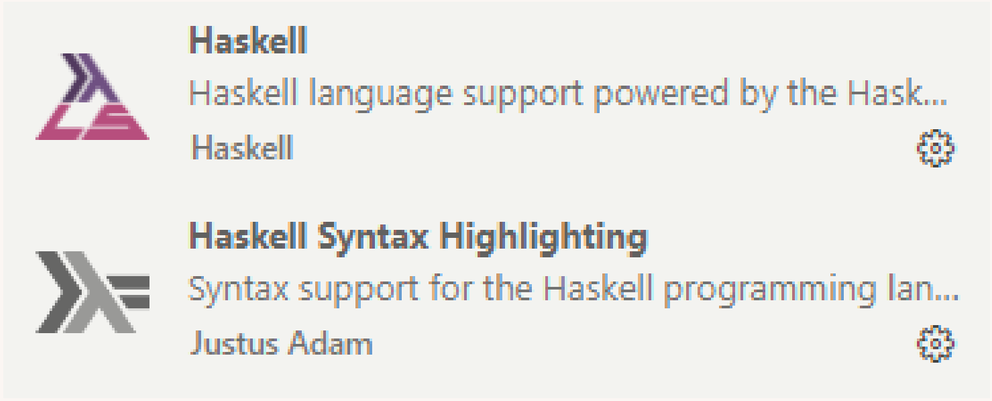
- **Hugs**: interpretador escrito em C, funciona em todas as máquinas (link);
- **GHC**: O *Glasgow Haskell Compiler* escrito em Haskell, mais popular, possui várias bibliotecas (link);
- **GHCup:** Ferramenta que facilita a instalação e o gerenciamento de múltiplas versões do GHC e outras ferramentas, como cabal e stack(link).



IDE'S

PUC Minas

- Não possui um ambiente próprio de desenvolvimento;

- Qualquer editor de texto pode ser utilizado:
  - Vim;
  - Visual Studio Code:
    - 

PUC Minas

CÓDIGO

code.hs

```haskell
-- Definição de tipos de dados básicos em Haskell
numeroInt :: Int
numeroInt = 10

numeroFloat :: Float
numeroFloat = 3.14

texto :: String
texto = "Olá, Haskell!"

-- Função recursiva: cálculo do fatorial
fatorial :: Int -> Int
fatorial 0 = 1
fatorial n = n * fatorial (n - 1)

-- Função polimórfica: função 'identidade' que retorna o mesmo valor que recebe
identidade :: a -> a
identidade x = x

-- Função de ordem superior: aplica uma função a cada elemento de uma lista
aplicaDuasVezes :: (a -> a) -> a -> a
aplicaDuasVezes f x = f (f x)

-- Função que usa a função de ordem superior 'map' para dobrar os números de uma lista
dobrarLista :: [Int] -> [Int]
dobrarLista xs = map (*2) xs

```

PUC Minas

CÓDIGO

code.hs

```haskell
-- Programa principal
main :: IO ()
main = do
    -- Mostra tipos de dados
    putStrLn ("Numero Int: " ++ show numeroInt)
    putStrLn ("Numero Float: " ++ show numeroFloat)
    putStrLn ("String: " ++ texto)

    -- Usa a função recursiva
    putStrLn ("Fatorial de 5: " ++ show (fatorial 5))

    -- Usa a função polimórfica 'identidade'
    putStrLn ("Identidade de 42: " ++ show (identidade 42))
    putStrLn ("Identidade de 'Haskell': " ++ show (identidade "Haskell"))

    -- Usa a função de ordem superior
    putStrLn ("Aplica duas vezes (*3) a 2: " ++ show (aplicaDuasVezes (*3) 2))

    -- Usa a função de ordem superior 'map'
    putStrLn ("Dobrando os valores da lista [1, 2, 3, 4]: " ++ show (dobrarLista [1, 2, 3, 4]))

```

PUC Minas

CÓDIGO

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS COMMENTS

```
vinma@LAPTOP-713PRD5R MINGW64 ~/Desktop/Estudos/LP (main)
$ ghc code.hs
[1 of 2] Compiling Main             ( code.hs, code.o )
[2 of 2] Linking code.exe [Objects changed]

vinma@LAPTOP-713PRD5R MINGW64 ~/Desktop/Estudos/LP (main)
$ ./code.exe
Numero Int: 10
Numero Float: 3.14
String: Olá, Haskell!
Fatorial de 5: 120
Identidade de 42: 42
Identidade de 'Haskell': "Haskell"
Aplica duas vezes (*3) a 2: 18
Dobrando os valores da lista [1, 2, 3, 4]: [2,4,6,8]

vinma@LAPTOP-713PRD5R MINGW64 ~/Desktop/Estudos/LP (main)
$
```

REFERÊNCIAS:

PUC Minas

- HASKELL. **Haskell: an advanced, purely functional programming language**. Disponível em: https://www.haskell.org. Acesso em: 15 set. 2024.
- UPENN. **Lecture 01: Introduction to Haskell. University of Pennsylvania, 2013**. Disponível em: https://www.seas.upenn.edu/~cis1940/spring13/lectures/01-intro.html. Acesso em: 15 set. 2024.
- FONTELA, Tailor. Capítulos de Haskell. 2022. Disponível em: https://haskell.tailorfontela.com.br/chapters. Acesso em: 15 set. 2024.
- ANDRIANA, Maria. **Paradigmas de programação funcional. Universidade Federal de Uberlândia**. Disponível em: https://www.facom.ufu.br/~madriana/PF_BCC.html. Acesso em: 15 set. 2024.
- IEEE. Análise da linguagem Haskell. IEEE Xplore, 2018. Disponível em: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8394394. Acesso em: 15 set. 2024.
- WIKIPEDIA. **Haskell (linguagem de programação). Wikipédia, a enciclopédia livre**. Disponível em: https://pt.wikipedia.org/wiki/Haskell_(linguagem_de_programa%C3%A7%C3%A3o). Acesso em: 15 set. 2024.
- SOUZA, Vitor. **Seminário de Haskell - LP. Universidade Federal do Espírito Santo, 2020**. Disponível em: http://www.inf.ufes.br/~vitorsouza/archive/2020/wp-content/uploads/teaching-lp-20142-seminario-haskell.pdf. Acesso em: 15 set. 2024.
- UFABC. **Introdução à linguagem Haskell. Universidade Federal do ABC**. Disponível em: https://haskell.pesquisa.ufabc.edu.br/haskell/03.haskell.basico.1/#:~:text=Haskell%20tem%20como%20características%3A,menores%20que%20em%20outras%20linguagens. Acesso em: 15 set. 2024.

REFERÊNCIAS:

PUC Minas

- HASKELL. **Haskell: an advanced, purely functional programming language**. Disponível em: https://www.haskell.org. Acesso em: 15 set. 2024.
- UPENN. **Lecture 01: Introduction to Haskell. University of Pennsylvania, 2013**. Disponível em: https://www.seas.upenn.edu/~cis1940/spring13/lectures/01-intro.html. Acesso em: 15 set. 2024.
- FONTELA, Tailor. Capítulos de Haskell. 2022. Disponível em: https://haskell.tailorfontela.com.br/chapters. Acesso em: 15 set. 2024.
- ANDRIANA, Maria. **Paradigmas de programação funcional. Universidade Federal de Uberlândia**. Disponível em: https://www.facom.ufu.br/~madriana/PF_BCC.html. Acesso em: 15 set. 2024.
- IEEE. Análise da linguagem Haskell. IEEE Xplore, 2018. Disponível em: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8394394. Acesso em: 15 set. 2024.
- WIKIPEDIA. **Haskell (linguagem de programação). Wikipédia, a enciclopédia livre**. Disponível em: https://pt.wikipedia.org/wiki/Haskell_(linguagem_de_programa%C3%A7%C3%A3o). Acesso em: 15 set. 2024.
- SOUZA, Vitor. **Seminário de Haskell - LP. Universidade Federal do Espírito Santo, 2020**. Disponível em: http://www.inf.ufes.br/~vitorsouza/archive/2020/wp-content/uploads/teaching-lp-20142-seminario-haskell.pdf. Acesso em: 15 set. 2024.
- UFABC. **Introdução à linguagem Haskell. Universidade Federal do ABC**. Disponível em: https://haskell.pesquisa.ufabc.edu.br/haskell/03.haskell.basico.1/#:~:text=Haskell%20tem%20como%20características%3A,menores%20que%20em%20outras%20linguagens. Acesso em: 15 set. 2024.

REFERÊNCIAS:

PUC Minas

- WIKIPEDIA. **Haskell Programming Language**. Disponível em: http://en.wikipedia.org/wiki/Haskell_programming_language. Acesso em: 15 set. 2024.
- HUDAK, Paul; PETERSON, John; FASEL, Joseph. **A Gentle Introduction to Haskell 98**. Disponível em: http://www.haskell.org/tutorial/. Acesso em: 15 set. 2024.
- HUGHES, John. **Why Functional Programming Matters**. Disponível em: http://www.cse.chalmers.se/~rjmh/Papers/whyfp.html. Acesso em: 15 set. 2024.
- SANTOS, Igor. **Twiki, Haskell**. Disponível em: http://twiki.im.ufba.br/bin/view/MAT052/HaskellIgor. Acesso em: 15 set. 2024.
- HASKELL-PRIME. **Haskell-Prime**. Disponível em: http://hackage.haskell.org/trac/haskell-prime. Acesso em: 15 set. 2024.
- DUTTON, C. **Introduction to Haskell**. Disponível em: http://www.iceteks.com/articles.php/haskell. Acesso em: 15 set. 2024.

PUC Minas

MUITO OBRIGADO!

Caio Faria Diniz
Giuseppe Senna Cordeiro
Vinicius Miranda de Araújo