



Programação Funcional



Haskell, ERLANG, LISP



Grupo

Erica Amoedo
Matheus Soares
Vinícius Pessoa

Haskell



Haskell

- Linguagem puramente funcional.
- Tipos definidos na hora da compilação.
- Open source. (Packages)
- Os tipos não precisam ser explicitados.

Erlang



Erlang

- Lançada em 1986 pela empresa Ericsson
- Atualmente encontra-se na versão 22 - maio de 2019
- Linguagem de uso geral
- Tolerante a falhas
- Hot Swapping



Lisp



Lisp

- Desenvolvida por John McCarthy em 1958, enquanto estava no MIT
- Nome derivado de "LISt Processor"
- Uso de S-expressions
- Existem vários dialetos, Ex: Common Lisp, Scheme, Clojure

Comparação das linguagens

Tipagem

- Inferência de tipos;
- Tipagem Estática;



- Inferência de tipo;
- Tipagem dinâmica;
- Tipagem forte.



- Dinâmica
- Fortemente tipada



Simplicidade

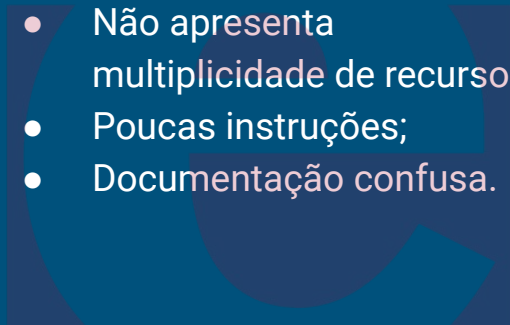
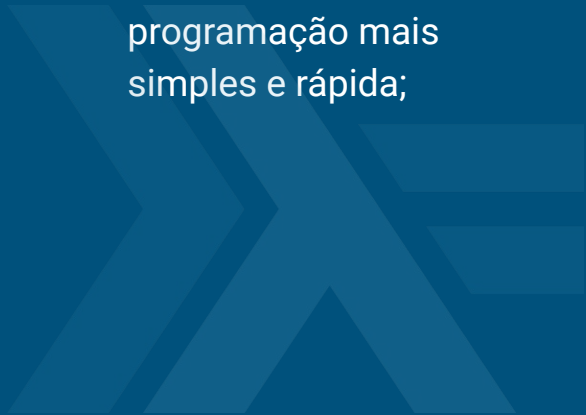
- Funções podem ser escritas em poucas linhas;
- Algumas funções inerentes à plataforma são abstraídas;
- Poucos tipos de dados primitivos;
- Regras bem definidas;
- Variáveis são capitalizadas;
- A última instrução de uma função é sempre retornada.
- Possui uma das sintaxes mais simples, com poucos tipos de dados primitivos

Legibilidade

- Poucos parênteses e chaves;
- Tudo está explícito, não há estado implícito ou variáveis globais pré-definidas;
- Considerada por alguns como a linguagem mais legível;
- Sintaxe fácil de ser entendida;
- Ausência de parênteses e chaves;
- Cada linha de uma função encerra com uma vírgula;
- A última linha de uma função encerra com um ponto.
- Muitos parênteses.
- Difícil de ler no início por usar uma estrutura diferente das linguagens mais comuns.

Facilidade de escrita

- Poucas vírgulas, chaves e parênteses, tornando a programação mais simples e rápida;
- Utiliza um conceito parecido com as 'arrow functions' do Javascript;
- Não apresenta multiplicidade de recurso;
- Poucas instruções;
- Documentação confusa.
- Simples, porém o uso de parênteses pode trazer confusão



Confiabilidade

- Safe Haskell (GHC 7.2)
 - Segurança de tipagem;
 - Transparência referencial;
 - Encapsulamento modular rigoroso;
 - raciocínio modular;
 - consistência semântica;

- Baixa verificação de erros durante a compilação;
- try... ..catch não trivial de ser utilizado;
- Variáveis imutáveis.

- Estável



Referências

Haskell. Disponível em: [<https://www.haskell.org/>]. Acesso em: 23/09/2019.

Safe Haskell. Disponível em: [https://wiki.haskell.org/Safe_Haskell]. Acesso em 23/09/2019.

CLisp Wiki. Disponível em: [<https://cliki.net/>]. Acesso em: 22/09/2019

Erlang. Disponível em: [<https://www.erlang.org/>]. Acesso em: 22/09/2019

Linguagem de programação: critérios. Disponível em:
[<https://universidadedatecnologia.com.br/linguagem-de-programacao-criterios/>].
Acesso em: 23/09/2019