

# Programação Funcional

## I/O em Haskell

**Prof. Edson Alves**

Faculdade UnB Gama

2020

# Sumário

1. Entrada e saída em console
2. Manipulação de arquivos

## Código puro e código impuro

- ▶ Haskell determina uma clara separação entre código puro e código impuro
- ▶ Esta estratégia permite que código puro fique isento de efeitos colaterais
- ▶ Além de facilitar a divisão semântica do código, ela permite aos compiladores otimizar e paralelizar trechos de código automaticamente
- ▶ Como as rotinas de entrada e saída interagem com o mundo externo, todas elas produzem ou estão suscetíveis a efeitos colaterais, sendo assim, códigos impuros

## Leitura e escrita de strings em console

- ▶ Haskell provê um conjunto de funções para escrita e leitura de dados a partir do console
- ▶ No que diz respeito à strings, duas funções básicas são `putStrLn` e `getLine`
- ▶ A função `putStrLn` escreve uma string no console, seguida de uma quebra de linha, e tem tipo

```
ghci> :type putStrLn
putStrLn :: String -> IO ()
```

- ▶ Já a função `getLine` lê uma string do console até encontrar uma quebra de linha e a retorna, sem a quebra
- ▶ O tipo da função `getLine` é

```
ghci> :type getLine
getLine :: IO String
```

## Exemplo de leitura e escrita de strings em console

```
1 -- Lê uma string do console e a imprime, sem modificações
2 main = do
3     putStrLn "Insira uma string: "
4     s <- getLine
5     putStrLn $ "A string inserida foi '" ++ s ++ "'"
```

# Referências

1. **SHALOM**, Elad. *A Review of Programming Paradigms Throughtout the History – With a Suggestion Toward a Future Approach*, Amazon, 2019.
2. **SULLIVAN**, Bryan O.; **GOERZEN**, John; **STEWART**, Don. *Real World Haskell*, O'Reilly.