**Pontifícia Universidade Católica do Paraná**

**Escola Politécnica**

**Bacharelado em Ciência da Computação**

**Programação Funcional**

**Prof. Frank Coelho de Alcantara**

**Matheus Leindorf Muller**

**Trabalho 1 - Currying**

**Curitiba**

**2022**

O processo de Currying nos possibilita pegar uma função com dois ou mais parâmetros e transformar em duas ou mais funções lambdas que contêm apenas um parâmetro, onde cada uma dessas funções retorna outra função. Dessa forma, no final do processo iremos ter o resultado final.(Servadei, 2017, p.18) Por exemplo, temos uma função que recebe três parâmetros e mandamos um parâmetro por vez e depois desse processos obtemos o resultado.

O método Currying recebe esse nome em homenagem ao matemático Haskell Curry que foi um dos pioneiros da criação do lambda cálculo. (Thompson, p.23)

Uma das vantagens da função Curry é que ela é considerada mais flexível do que as funções que utilizam tuplas, pois muitas vezes podemos aplicar a técnica Curry em funções normais (Lämmel, 2011, p.58). Esse método tem como principal função simplificar a leitura das funções matemáticas, assim sendo mais fácil de identificar as diferentes partes da função.

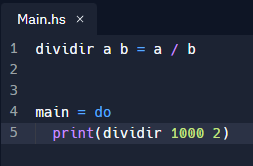
Utilizando o cálculo lambda junto com o método Currying, podemos colocar diversos parâmetros. Para uma função de calcular uma divisão, ficaria assim:

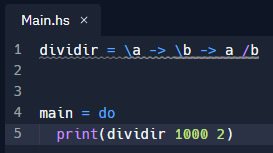
**λa . λb. a / b**

**λa . a / 1000 2**

**1000 / 2**

**500**

Uma função básica de divisão em Haskell seria representada da seguinte forma:

Agora aplicando o método Curry no código acima ficaria da seguinte forma:

**Referências**

SERVADEI, Giacomo. Toward a more expressive pattern matching in Haskell. 2017. Disponível em <<https://www.politesi.polimi.it/bitstream/10589/140161/1/Tesi.pdf>>. Acesso em 07 de Agosto de 2022.

THOMPSON, Simon. Haskell The Craft of Functional Proa ramming. Disponível em <<http://www0.cs.ucl.ac.uk/teaching/3C11/book/Ch4.pdf>>. Acesso em 07 de agosto de 2022.

LAMMEL, Ralf.Introduction to Haskell. 2011. Disponível em <<https://userpages.uni-koblenz.de/~laemmel/paradigms1011/resources/pdf/haskell.pdf>>. Acesso em 07 de agosto de 2022.