



PONTIFÍCIA UNIVERSIDADE CATÓLICA DO PARANÁ  
Escola Politécnica

Curso: Ciência da Computação  
Disciplina: Programação Funcional

Trabalho 01 – 05/agosto/2022 – Currying

Nome: Gustavo Rodrigues Guimaraes

1. Descreva sobre a funcionalidade de “Currying”?
  - a. o Currying é uma forma de, digamos que parcelar os parâmetros que uma função recebe, colocando como um exemplo:

funct                      nome\_funcao(a,b,c):                      Return                      a+b\*c

Ao invés de fazer isso e chamar ela como :nome\_funcao(2,3,4) afim de receber 14 como resultado, podemos parcelar isso chamando outra função, assim o que retorna não é um valor, mas sim uma nova função:

funct nome\_funcao(a): a+nome\_funcao2(b)

funct nome\_funcao(b): b\*nome\_funcao3(c)

funct nome\_funcao(c): c

assim                      parcelamos                      sua                      chamada                      como:  
a = nome\_funcao(2)

b = a(3)

c = b(4)

Ou seja, a função estará a se desenvolver, contudo a sua função anterior já estará com um valor definido.

2. Qual sua Finalidade?

- a. Reduzir a quantidade de parâmetros que uma função recebe e conseguir de certa forma diminuir a repetibilidade de cálculos, como por exemplo, ainda considerando o modelo da questão 1:
  - i. quero descobrir uma divisão a qual tem uma multiplicação dos dividendos, então recebo 3 valores, na qual o dividendo A é 2, o dividendo B é 5, o divisor é X, e o resultado é 0,0014

```
funct nome_funcao(X): nome_funcao2(b)/ X
```

```
funct nome_funcao(b): b+nome_funcao3(c)
```

```
funct nome_funcao(c): c
```

assim                    parcelamos                    sua                    chamada                    como:

```
X = loop(nome_funcao(x),x++, while X!=0,0014)
```

```
a = X(2)
```

```
b = a(5)
```

3. Por que ela tem esse nome?

- a. Inicialmente ela possui esse nome devido ao seu criador, haskell curry que devido as suas contribuições na área computacional, dentre uma delas ele contribuiu com a criação de um conceito, tal esse que teria o seu sobrenome como forma de homenagem

4. Forneça exemplos em um cálculo lambda(haskell)

- a. Por haskel já ser uma linguagem puramente funcional, então o calculo lambda usando carrying é extremamente essencial e ajuda muito, entao um exemplo simples é:

(Código)

```
media :: Float->Float -> Float
```

```
media a b = (a + b )/ 2
```

(terminal)

```
s = media 2
```

```
s 3
```