

Trabalho

T300 – Programação Funcional

Prof. Prof. Cenez de Rezende

Tema do Trabalho – Aplicação de Técnicas Funcionais

Data da Entrega: AVA

Número de alunos: máximo de 3 alunos por grupo

Este trabalho deve ser apresentado no google meet, para aferição das atividades do acadêmico. Sem esta apresentação, não haverá pontuação.

Etapa 1:

1) [1 Pontos] Programe os seguintes cenários:

- a) Em programação funcional, qual a **finalidade** prática da notação **lambda** e da **compreensão** de listas? Dê algum **exemplo** em **Elixir** que explore **compreensão** de listas e **lambda** em conjunto;
- b) **Apresente e explique** um código que usa a técnica de **currying** em **elixir**.

2) [1 Pontos] Usando **Elixir** ou **Haskell**, defina uma função **f(ps,xs)** que recebe duas listas **ps** e **xs** como parâmetros de **f**. A primeira lista (**ps**) possui pares internos do tipo **{X, Y}**, e a segunda lista (**xs**) possui elementos pares do tipo **{X,X}**. Portanto, **X** e **Y** são qualquer tipo de dados. Uma chamada à função **f**, como **f(ps, xs)**, retorna uma lista com os pares de **ps** cujo primeiro elemento esteja na lista **xs**.

Por exemplo:

chamada **f(ps, xs)**, com **ps** e **xs** valorados assim: **f([{x₁,y₁}, {x₂,y₂}, {x₃,y₃}], [{x₁,x₃}])**,

nós temos o retorno **[{x₁,y₁}, {x₃,y₃}]**, que são respectivamente o par 1 **{x₁,y₁}** e o par 3 **{x₃,y₃}** da lista **ps**.

Ou seja, **x₁** e **x₃** estão na lista **xs**, enquanto **x₂** não está na lista **xs**.

3) [1 Pontos] Escreva uma função que receba duas listas e retorne outra lista com os elementos das listas originais intercalados: **Exemplo: intercala([1, 2, 3], [4, 5, 6]) = [1, 4, 2, 5, 3, 6]**

4) [3 Pontos] Dado um valor **k** e uma lista de números, imprimir **k** listas da seguinte forma:

Entrada: f(k, [v₁, v₂, ..., v_n])

Saída com k listas: [lista₁, lista₂, lista₃, ..., lista_k] tal que as listas incluem valores **v_i** não repetidos, e sequenciados como os 3 exemplos seguintes:

Entradas (f(k, lista))

f(3, [1,2,3,4,5,6])

f(2, [1,2,3,4,5,6])

f(3, [1,2,3,4,5,6,7,8,9,10])

Saídas (lista de listas)

→ [[1,4],[2,5],[3,6]]

→ [[1,3,5],[2,4,6]]

→ [[1,4,7,10],[2,5,8],[3,6,9]]

- 5) [1 Pontos] Fazer uma função **anagrama(s1, s2):Bool** que recebe dois parâmetros do tipo string (s1, s2), retornando **verdadeiro** ou **falso** de acordo com as seguintes regras:
- a) Se s1 e s2 possuem o mesmo tamanho e as mesmas letras, independente da ordem, retornar verdadeiro. Veja exemplos:
- i) anagrama(“amar”, “mara”) retorna verdadeiro;
 - ii) anagrama(“amar”, “nara”) retorna falso;
 - iii) anagrama(“pato”, “topa”) retorna verdadeiro;
 - iv) anagrama(“unifor”, “roufin”) retorna verdadeiro;

Etapa 2 (Map e Filter):

As funções **filter (filtrar)** e **map (mapear)**, em linguagens funcionais são comuns e facilitam bastante nossa programação. Essas funções, na verdade, são pequenos blocos de codificações que seguem técnicas de programação funcional, possíveis de serem implementadas. Nos próximos exercícios, sua atividade é implementar em **Elixir** as funções **mapear** e **filtrar**, para que aceitem funções e listas como argumentos. Deixo aqui um exemplo de uso das funções nativas em Elixir:

```
Enum.filter([1,2,3,4,5,6], fn(x) -> rem(x, 2)==0 end)
```

```
# [2, 4, 6]
```

```
Enum.map([1,2,3,4,5,6], fn(x) -> x*2 end)
```

```
# [2, 4, 6, 8, 10, 12]
```

Ou seja, **você construirá** um módulo, por exemplo **Unifor**, que dentro dele conterà as funções **mapear** e **filtrar**. **Não é permitido usar o módulo Enum do Elixir, que já contém tais funções nativas**. Sua **implementação** deve ser **pura**, usando recursividade ou compreensão de listas. Portanto, não se deve usar recursos do Elixir que já fazem tal funcionalidade.

[1,5 Pontos] Implementar a função **mapear**.

```
defmodule Unifor do
```

```
.....
```

```
end
```

[1,5 Pontos] Implementar a função **filtrar**.

```
defmodule Unifor do
```

```
.....
```

```
end
```

Critérios de Avaliação

Ausência de efeito colateral;

Corretude nos resultados;

Funções objetiva e curtas. Não devem exceder 5 linhas;

Entrega no prazo;