

Questão 1 - [1 ponto] - Explique a diferença entre funções puras e funções de ordem superior em programação funcional. Dê exemplos de cada uma.

Funções puras não utilizam outras funções como entrada, há apenas dados. Já as funções de ordem superior utilizam outras funções (elas puras ou não) como parte da sua resolução.

Questão 2 - [1 ponto] - Discorra sobre as vantagens da linguagem de programação elixir e em que tipo de projeto ou cenário devemos optar pelo uso dessa tecnologia.

É uma programação funcional, que trabalha com resolução de grandes problemas segmentando em várias funções/etapas, homologando cada uma delas com dados imutáveis, tornando o processo mais fácil de ser testado. Após o teste de pequenas funções, as mesmas podem ser testadas em diversas partes do programa, garantindo sustentabilidade no projeto já que os testes serão eficazes como um todo.

Questão 3 - [0,5 pontos] - Escreva uma função que verifique se um número é par.

```
defmodule Questao do
```

```
  def verifica(num) do
    res = rem(num,2)
```

```
    if (res == 0) do
```

```
      IO.puts("O número inserido ( #{num} ) é PAR")
```

```
    else
```

```
      IO.puts("O número inserido ( #{num} ) é IMPAR")
```

```
    end
```

```
  end
```

```
end
```

```
Questao.verifica(10)
```

Questão 4 - [0,5 pontos] - Implemente uma função que calcule o dobro de cada elemento em uma lista.

```
defmodule Questao do
```

```
  def dobro(lista) do
```

```
    res = Enum.map(lista, fn x -> x * 2 end)
```

```
    res = Enum.join(res, ", ")
```

```
    IO.puts("O dobro dos números inseridos é [ #{res} ]")
```

```
  end
```

```
end
```

```
Questao.dobro([10,20,50])
```

Questão 5 - [0,5 ponto] - Crie uma função que retorne o último elemento de uma lista.

```
defmodule Questao do
```

```
  def ultimo(lista) do
```

```
    contagem = Enum.count(lista)
```

```
    contagem = contagem-1
```

```
    resultado = Enum.fetch!(lista,contagem)
```

```
    IO.puts("O ultimo elemento da lista é #{resultado} ")
```

```
  end
```

```
end
```

```
Questao.ultimo([3,2,5])
```

Questão 10 - [2,5 ponto] - Crie uma função que calcule a média de uma turma.

O exemplo abaixo demonstra como são passadas as notas dos alunos pertencentes a uma turma.

Você deve utilizar o método `reduce` para calcular a média.

```
notas_da_turma = [  
  {"Alice", [9.5, 8.0, 7.5]},  
  {"João", [8.0, 7.0, 6.5]},  
  {"Pedro", [9, 9.5, 9.0]},  
  {"Lucas", []},  
]
```

Dicas: Utilize `map` ou `flatMap` para extrair todas as notas dos alunos em uma única lista

```
def module Questao do
```

```
  def calcula_media(alunos_notas) do
```

```
    alunos = Enum.map(alunos_notas, fn {_nome, nota} -> count(nota) > 0 end)
```

```
    notas = Enum.flat_map(alunos, fn {_nome, nota} -> nota end)
```

```
    IO.puts(notas)
```

```
  end
```

```
end
```

```
notas_da_turma = [  
  {"Alice", [9.5, 8.0, 7.5]},  
  {"João", [8.0, 7.0, 6.5]},  
  {"Pedro", [9, 9.5, 9.0]},  
  {"Lucas", []},  
]
```

```
Questao.calcula_media(notas_da_turma)
```

