

# Trabalho Prático I – Programação Funcional

Este trabalho deve ser postado no Moodle como um arquivo compactado (.zip) contendo o arquivo fonte do programa. O nome do arquivo enviado deve ser formado pelo nome do aluno sem espaço e acentuação. Por exemplo, João Souza fez o trabalho, então o nome do arquivo é JoaoSouza.zip.

O trabalho consiste em criar um programa para realizar vendas. Cada questão refere-se a uma parte do programa. Note que algumas funções auxiliam na criação das demais. Lembre-se que uma função pode exigir a criação de outras auxiliares.

**Questão 01** [1,0]. Crie os tipos: *Nome*, equivalente a *String*; *Valor*, equivalente a *Float*; *Quantidade*, equivalente a *Int*; *Produto*, com nome e valor; *Item*, com produto e quantidade. Crie uma função chamada *produtos* que retorna uma lista com 10 produtos a sua escolha.

**Questão 02** [1,0]. Crie as seguintes funções auxiliares: *repete*, que recebe um elemento de qualquer tipo e um inteiro e retorna uma lista com o elemento repetido o número de vezes informado; *index*, que recebe um elemento de um subtipo de *Eq* e uma lista do mesmo tipo e retorna o índice do elemento na lista se ele existir ou *Nothing*, caso o elemento não exista na lista; e elemento que recebe uma lista de qualquer tipo e um índice (*Int*) e retorna o elemento no índice informado ou *Nothing*, para um índice inválido. Para essa questão pesquise o tipo *Maybe* do Haskell.

**Questão 03** [1,0]. Crie a função *addProduto* que recebe uma lista de *Produto* e um *Produto* e retorna a lista com o produto adicionado no final. Crie a função *remProduto* que recebe uma lista de *Produto* e um *Nome* e retorna a lista sem o produto com nome passado. Crie a função *buscaProduto* que recebe uma lista de *Produto* e um *Nome* e retorna um *Maybe Produto*, que será o produto da lista possui aquele nome ou *Nothing* (ver tipo *Maybe*).

**Questão 04** [0,5]. Crie uma função chamada *alinhaEsq* que recebe uma *String*, um caractere e um inteiro como parâmetros e retorna uma *String* composta pela *String* informada e uma sequência do caractere de modo que o tamanho total da *String* resultante ocupe quantidade informada. Crie uma função chamada *alinhaDir* que recebe uma *String*, um caractere e um inteiro como parâmetros e retorna uma *String* composta por uma sequência do caractere de modo seguida da *String* informada e que o tamanho total da *String* resultante ocupe quantidade informada. Ex.:

```
> alinhaEsq "AGUA MINERAL" \.' 20
"AGUA MINERAL....."
> alinhaDir "AGUA MINERAL" \ ' 20
"      AGUA MINERAL"
> alinhaDir "CAFÉ" \.' 20
".....CAFE"
```

**Questão 05** [1,0]. Crie um operador *\$\$* não associativo e com prioridade 5. Esse operador recebe um *Valor* e um inteiro e retorna uma *String* com o valor no formato  $[0-9]^+.[0-9]_1[0-9]_2 \dots [0-9]_n$ . Crie também uma função chamada *dinheiro* que recebe um *Valor* e usa o operador *\$\$* para transformar esse valor em uma *String* com um cifrão (\$) na frente e sempre dois algarismos depois do ponto. Ex.:

```
> 12.5 $$ 2;
"12.50"
> 155.982401 $$ 3;
"155.982"
> dinheiro 3647.74299
"$3647.74"
```

**Questão 06** [1,0]. Crie uma função chamada *formataItem* que recebe um *Item* e retorna um item da nota fiscal em uma String de 80 caracteres. Essa função deve formatar a String retornada utilizando espaços em banco da seguinte forma: 45 caracteres para o nome do produto; 25 para o valor e a quantidade; e 10 para o subtotal. Ex.:

```
> formataItem ("AMENDOIM", 3.99, 5);
AMENDOIM..... $3.99 x 5 =    $19.95
```

**Questão 07** [1,0]. Crie uma função chamada *total* que recebe como entrada uma lista de itens, calcula o valor total dos itens da lista e retorna esse valor formatado como dinheiro. O valor total é igual à soma de dos valores dos produtos dos itens de venda multiplicados por sua quantidade vendida.

**Questão 08** [1,0]. Crie uma função chamada *notafiscal* que recebe como entrada uma lista de itens e retorna uma String que representa a nota fiscal da venda. A largura da linha da nota é 80 caracteres.

```
> notafiscal [(("AMENDOIM", 3.99),5), (("HEINEKEN 5L", 64.99),10)]

*****
                        NOTA FISCAL
*****

AMENDOIM..... $3.99 x 5 =    $19.95
HEINEKEN 5L..... $64.99 x 10 = $649.90

*****
                        TOTAL: $669.85
*****
```

**Questão 09** [1,0]. Crie a função *proditem* que não possui entradas e retorna uma lista de *Item* contendo todos os elementos da lista *produtos* e a quantidade sendo o índice de cada produto na lista. Crie também a função *proditemx*, que é semelhante a função *proditem*, porém recebe como entrada as uma lista de quantidades. Se esta lista de quantidades for menor que a lista de produtos, então somente os n primeiros produtos são considerados, onde n é o tamanho da lista de quantidades.

**Questão 10** [1,0]. Crie a função chamada *itensn* que recebe uma lista da tupla (*Nome*, *Quantidade*) e retorna uma lista do tipo *Item*, com os produtos identificados a partir do nome na lista *produtos* (Questão 1). Crie também a função chamada *itensi* que recebe uma lista da tupla (*Int*, *Quantidade*) e retorna uma lista do tipo *Item*, sendo que o *Int* da tupla corresponde ao índice na lista *produtos*.

**Questão 11** [0,5]. Crie a função *venda* que recebe uma lista de *Item* e realiza a venda, imprimindo a nota fiscal. O tipo de retorno dessa função é *IO()*. Ex.:

```
> venda (itensi [(1,2), (2,5), (4,10)] produtos)

*****
                        NOTA FISCAL
*****

AGUA MINERAL..... $2.50 x 2 =    $5.00
CHOCOLATE..... $3.99 x 5 =    $19.95
LEITE..... $2.99 x 10 =    $29.90

*****
                        TOTAL: $54.85
*****
```