

Entregue todos os métodos em um arquivo chamado `lab5.py`. Critérios de avaliação: Principalmente (25%) o código funciona, os métodos fazem o que foi pedido, parâmetros de entrada e valores de retorno são corretos! **Adicionalmente** (50%) o código usa o tratamento de exceções corretamente. Adicionalmente (25%) o código é legível, eficaz, o mais simples possível, não cria variáveis desnecessárias.

1. (3 pontos) Crie a função chamada `absoluto` com um parâmetro de entrada. O parâmetro de entrada esperado é um número ou uma string de um número, mas a função deve tratar os erros que ocorrem caso a entrada não for "válida" e deve imprimir uma mensagem apropriada. Caso a entrada for válida, a função retorna o valor absoluto do número passado. Veja os exemplos de chamada:

```
>>> absoluto("-0.98")
0.98
>>> absoluto(12)
12.0
>>> absoluto([1,2])
TypeError para <class 'list'>
>>> absoluto("a")
ValueError para 'a'
```

2. Crie a classe `Loja` com dois atributos: `nome`, `produtos` e os seguintes métodos:

- (a) (1 ponto) Construtor com dois parâmetros de entrada: uma string `nome` e um dicionário `produtos` com valor *default* dicionário vazio. As chaves do dicionário devem ser as categorias dos produtos (e.g., esmalte) e os valores são conjuntos das marcas daquela categoria (e.g., Colorama). Vê um exemplo abaixo:

```
>>> beleza1 = Loja("Beleza", {"esmalte": {"Colorama", "Max Factor"}, "desodorante": {"Nivea", "Dove"}})
>>> beleza1.produtos
{'esmalte': {'Max Factor', 'Colorama'}, 'desodorante': {'Nivea', 'Dove'}}
```

- (b) (2 pontos) Método chamado `adicionarProduto` com dois parâmetros de entrada: `categoria` e `marca` que adiciona o produto da `marca` na `categoria`. Trate o caso que a `categoria` não existir no dicionário `produtos` com o tratamento de exceções. É proibido usar comando `if` e outras expressões booleanas.

```
>>> beleza1.adicionarProduto("xampu", "TRESemmé")
>>> beleza1.produtos
{'esmalte': {'Max Factor', 'Colorama'}, 'desodorante': {'Nivea', 'Dove'}, 'xampu': {'TRESemmé'}}
```

- (c) (2 pontos) Método chamado `verCategoria` com o parâmetro de entrada `categoria` que retorna o conjunto das marcas da `categoria`. Se a `categoria` não existir no dicionário `produtos`, ocorre um erro – trate esse erro com o tratamento de exceções e imprime uma mensagem como no exemplo abaixo. É proibido usar expressões booleanas.

```
>>> beleza1.verCategoria("esmalte")
{'Max Factor', 'Colorama'}
>>> beleza1.verCategoria("sabão")
Categoria sabão não catalogada.
```

- (c) (2 pontos) Método chamado `removerMarca` com o parâmetro de entrada `marca` que remove essa `marca` de todos os conjuntos de todas as categorias.

```
>>> beleza1.removerMarca("TRESemmé")
>>> beleza1.produtos
{'esmalte': {'Max Factor', 'Colorama'}, 'desodorante': {'Nivea', 'Dove'}, 'xampu': set()}
```