

Protótipo de implementação sobre controle de estoque

Matheus Silveira Estima¹

¹ Centro de Ciências Computacionais – Universidade Federal do Rio Grande (FURG)
Campus Carreiros Caixa postal 474- Rio Grande – RS – Brazil

matheus.estima@furg.br

***Abstract.** O presente relatório tem o intuito de apresentar um protótipo de implementação sobre controle de estoque, feito em Python, baseado em um projeto em UML e implementação em Smalltalk.*

1. Informações Gerais

Para a elaboração do presente relatório foi estudado um UML e uma implementação em Smalltalk sobre um controle de estoque, tendo todas suas classes, métodos e atributos apontados. Com isso foi desenvolvido um protótipo do mesmo em Python, seguindo as estruturas apresentadas.

2. UML

Segundo a Wikipédia[1], “a UML (Linguagem de Modelagem Unificada), é uma linguagem-padrão utilizada para a elaboração da estrutura de projetos de software, podendo ser empregada para a visualização, especificação, construção e documentação de ferramentas que utilizem o sistema.”.

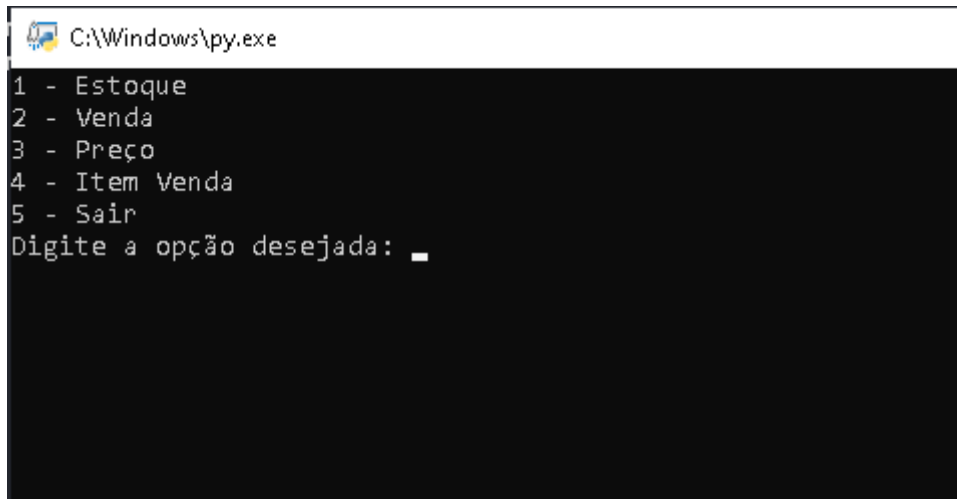
3. Objetivo

O presente relatório tem o intuito de analisar um projeto de controle de estoque em UML e implementado em Smalltalk. A partir disso, foi desenvolvido um protótipo funcional em Python, seguindo o mesmo padrão de classes, métodos e atributos para seu desenvolvimento e utilização, podendo ser executado e testado.

4. Aplicação

A aplicação[2], sendo feita em Python, foi feita utilizando classes, na qual cada uma compõe uma funcionalidade sendo chamada no momento em que o usuário escolhe qual

função ele deseja utilizar. O usuário pode fazer o controle de seu estoque e de outras ferramentas(Figura 1), estando todas vinculadas ao mesmo, sendo assim, quando ele faz uma venda, alteração de preço ou deseja fazer uma baixa, modifica em seu estoque.



```
C:\Windows\py.exe
1 - Estoque
2 - Venda
3 - Preço
4 - Item Venda
5 - Sair
Digite a opção desejada: _
```

Figura 1 -Ferramentas apresentadas pela aplicação

Entre as suas finalidades, são utilizadas as classes de Estoque, Produto, Preço, Venda e Itens Venda. Com isso, dentro de cada estrutura(Figura 2), é feita a criação dos objetos para poder ser trabalhado com a requisição do usuário.

```
2 class Estoque:
3     def __init__(self, nome, quantidade):
4         self.nome = nome
5         self.quantidade = quantidade
6     def baixa(self, quantidade):
7         if self.quantidade < quantidade:
8             print('Quantidade insuficiente em estoque')
9             return
10        self.quantidade -= quantidade
11    def repor(self, quantidade):
12        self.quantidade += quantidade
13    def mostra(self):
14        print('Produto: ', self.nome)
15        print('Quantidade: ', self.quantidade)
```

Figura 2 - Estrutura do Estoque

5. Referências

Linguagem de Modelagem Unificada - <https://pt.wikipedia.org/wiki/UML> [1]

Aplicação de controle de estoque em Python -
<https://github.com/Estima01/Implementa-o-vendas-O.O> [2]