**RELATÓRIO PROJETO N2**

Sistemas de Informação – 01J12

Eduardo Hiroyuki Tamaributi – 32331762

Julia Kovacs Takamura – 32371489

**Decisões de implementação**

Para a implementação do projeto, foram criadas 5 funções que executam as tarefas de cadastrar, consultar, atualizar, excluir e ver um relatório completo do armazém.

Os produtos foram criados na forma de dicionário, contendo um nome, um código e sua quantidade. Depois do item ser cadastrado, ele é adicionado à lista armazém.

A função de consultar um produto itera sobre os itens da lista armazém comparando o código do produto com o código fornecido pelo usuário, caso o produto seja encontrado, a função exibe as informações do produto, caso contrário, uma mensagem de erro será exibida.

A função de atualizar as informações de um produto utiliza o mesmo método para identificar se o código solicitado pelo usuário está presente na lista de produtos, caso estiver, é perguntado ao usuário se ele deseja atualizar o nome e/ou a quantidade daquele determinado produto.

A função de excluir um produto também utiliza o mesmo método para identificar se o código solicitado está presente na lista de produtos, caso estiver, é perguntado se o usuário deseja remover o item selecionado.

A função de exibir um relatório do armazém primeiramente exibe um cabeçalho com os tópicos: nome, código e quantidade. Além disso, é feito um cálculo de espaçamento para deixar a tabela formatada, ou seja, todas as colunas possuirão a largura da maior palavra. Depois de exibir o cabeçalho, as informações de cada produto são exibidas em cada linha da tabela.

Diagrama

Descrição gerada automaticamente

Planejamento do código - Figma

**MENU**

Texto

Descrição gerada automaticamente

**CADASTRAR PRODUTO**

Texto

Descrição gerada automaticamente

**CONSULTAR PRODUTO**

Interface gráfica do usuário, Texto

Descrição gerada automaticamente

**ATUALIZAR PRODUTO**

Texto

Descrição gerada automaticamente

**REMOVER PRODUTO**

Texto

Descrição gerada automaticamente

**RELATÓRIO DE PRODUTOS**

Diagrama

Descrição gerada automaticamente com confiança média

**Código Final Desenvolvido**

**def insira\_codigo () :**

**# Função para o usuário inserir o código do produto | Como o código foi utilizado diversas vezes, ele foi transformado em uma função**

**while True :**

**# Recebe um input do código do produto e caso ele possuir 4 dígitos, retorna o valor**

**codigo = int(input("Código do produto: "))**

**if codigo > 999 and codigo < 10000 :**

**return codigo**

**else :**

**print('Erro! Código deve conter 4 dígitos\n')**

**def cadastra\_produto () :**

**# Cria um dicionário {produto} que possui as categorias: none, código e quantidade**

**produto = {}**

**produto['nome'] = input("Nome do produto: ")**

**produto['code'] = insira\_codigo()**

**while True :**

**produto['quant'] = int(input("Quant: "))**

**if produto['quant'] > 0 :**

**break**

**else :**

**print("Erro! Quant deve ser maior que 0\n")**

**print(f'{produto["nome"]} cadastrado com sucesso\n')**

**return produto**

**def consulta\_produto (lista, codigo) :**

**# Faz a consulta de um produto específico cadastrado**

**for i in range (len(lista)) : # Itera entre os elementos da lista [estoque]**

**if lista[i]['code'] == codigo :**

**# Caso o código do elemento for o mesmo que o parâmetro da função, printa esse elemento**

**if len(lista[i]['nome']) > 6 :**

**espacamento = len(lista[i]['nome']) - 6**

**print('Produto', end='')**

**print(' ' \* espacamento, end='|')**

**print(' Codigo | Quant')**

**else :**

**print('Produto| Codigo | Quant')**

**print(f'{lista[i]["nome"]} | {lista[i]["code"]} | {lista[i]["quant"]}\n')**

**break**

**else :**

**# Caso não haja correspondência para o código inserido como parâmetro, devolve uma mensagem de erro**

**print("Erro! Produto não encontrado\n")**

**def atualiza\_produto (lista, codigo) :**

**# Atualiza o nome e quantidade de um produto da lista [estoque]**

**for i in range (len(lista)) : # Itera entre os elementos da lista [estoque]**

**if lista[i]['code']== codigo :**

**# Caso o código do elemento for o mesmo que o parâmetro da função, atualiza o seu nome e/ou quantidade**

**print(f'Produto a ser atualizado: {lista[i]["nome"]}')**

**while True :**

**att\_nome = input("Deseja atualizar o nome (s/n): ")**

**if att\_nome.lower() == 's' :**

**lista[i]['nome'] = input("Nome atualizado: ")**

**break**

**elif att\_nome.lower() == 'n' :**

**break**

**else :**

**print("Erro! Comando inválido\n")**

**while True :**

**att\_quant = input("Deseja atualizar a quant (s/n): ")**

**if att\_quant.lower() == 's' :**

**lista[i]['quant'] = int(input("Quant atualizada: "))**

**break**

**elif att\_quant.lower() == 'n' :**

**break**

**else :**

**print("Erro! Comando inválido\n")**

**print(f'{lista[i]["nome"]} atualizado com sucesso\n')**

**consulta\_produto(lista, codigo)**

**break**

**else :**

**# Caso não haja correspondência para o código inserido como parâmetro, devolve uma mensagem de erro**

**print("Erro! Produto não encontrado\n")**

**def excluir\_produto (lista, codigo) :**

**# Exclui um determinado elemento da lista [estoque]**

**for i in range (len(lista)) : # Itera entre os elementos da lista [estoque]**

**if lista[i]['code'] == codigo :**

**# Caso o código do elemento for o mesmo que o parâmetro da função, pergunta se o usuário quer fazer a remoção do item**

**while True :**

**confirmar = input(f"Deseja remover o item {lista[i]['nome']} (s/n): ")**

**if confirmar == 's' :**

**lista.pop(i)**

**print("Produto removido com sucesso\n")**

**break**

**elif confirmar == 'n' :**

**print("Operação cancelada\n")**

**break**

**else :**

**print('Erro! Comando inválido')**

**break**

**else :**

**print("Erro! Produto não encontrado\n")**

**def exibe\_relatorio (lista) :**

**# Exibe um relatório do estoque apresentando o nome, código e quantidade de cada item**

**if len(lista) > 0 :**

**# Faz o cabeçalho do relatório**

**espacamento = len(lista[0]['nome'])**

**for i in range (len(lista)) :**

**if len(lista[i]['nome']) > espacamento :**

**espacamento = len(lista[i]['nome'])**

**espacamento -= 7**

**if espacamento < 0 :**

**espacamento = 0**

**print('Produto', end='')**

**print(' ' \* espacamento, end='|')**

**print(' Codigo | Quant')**

**espacamento += 7**

**for i in range (len(lista)) :**

**# Printa os itens**

**print(lista[i]['nome'], end='')**

**print(' ' \* (espacamento - len(lista[i]['nome'])), end='|')**

**print(f' {lista[i]["code"]} | {lista[i]["quant"]}')**

**print()**

**else :**

**# Caso a lista [estoque] esteja vazia, retorna uma mensagem de erro**

**print("Erro! Armazém vazio\n")**

**print('+++++++ MENU – CONTROLE DE ESTOQUE +++++++')**

**estoque = []**

**while True :**

**# Printa as opções que o usuário possui**

**print('1. Cadastrar Produto\n2. Consultar Produto\n3. Atualizar Produto\n4. Excluir Produto\n5. Relatório de Produtos\n6. Encerrar')**

**escolha = int(input("Opção escolhida: "))**

**print()**

**if escolha == 6 :**

**break**

**elif escolha == 1 :**

**estoque.append(cadastra\_produto())**

**elif escolha == 2 :**

**consulta\_produto(estoque, insira\_codigo())**

**elif escolha == 3 :**

**atualiza\_produto(estoque, insira\_codigo())**

**elif escolha == 4 :**

**excluir\_produto(estoque, insira\_codigo())**

**elif escolha == 5 :**

**exibe\_relatorio(estoque)**

**else :**

**print("Erro! Comando inválido\n")**