

**UNIVERSIDADE DE FORTALEZA**

**CIÊNCIA DA COMPUTAÇÃO**

MATHEUS MOURA DE LIMA

**RELATÓRIO DE REQUISITOS DO SISTEMA STOCKEASE**

FORTALEZA

2024

**Requisitos**

Requisitos Funcionais:

* RF01: Cadastro de Usuário:
  + O sistema deve permitir o cadastro de novos usuários com um nome de usuário e senha.
  + Localização deste requisito no código: módulo Stockease.py. Função singup(). Linha: 69 até 87.
* RF02: Login de Usuário:
  + O sistema deve permitir que os usuários cadastrados façam login no sistema.
  + O sistema deve validar das credenciais do usuário, comparando o nome de usuário e senha fornecidos com os registros do banco de dados.
  + Após o login bem-sucedido, exibe a janela principal do sistema de gerenciamento de estoque.
  + Localização deste requisito no código: módulo Stockease.py. Função login(). Linha: 124 até 172. Função login\_account(). Linha: 91 até 115.
* RF03: CRUD de Produtos:
  + O sistema deve permitir a inserção, atualização, exclusão e seleção de produtos no banco de dados.
  + Os campos obrigatórios para inserção e atualização incluem nome, preço, quantidade e categoria do produto.
  + A categoria do produto é selecionada a partir de uma lista predefinida.
  + Localização deste requisito no código: módulo Stockease.py. Função save(). Linha: 304 até 365. Função update(). Linha: 368 até 404. Função delete(). Linha: 407 até 429. Função: select(). Linha: 432 até 446. Lista predefinida: linha 558.
* RF04: Busca de Produtos:
  + O sistema deve permitir a funcionalidade de pesquisa de produtos por nome.
  + Localização deste requisito no código: módulo Stockease.py. Função find(). Linha: 449 até 472.
* RF05: Limpeza de Campos:
  + O sistema deve permitir limpar os campos do formulário de cadastro de produtos.
  + Localização deste requisito no código: módulo Stockease.py. Função: clear(). Linha: 479 até 489.
* RF06: Validação de Permissões:
  + O sistema deve verificar se o usuário possui permissões de administrador, para que seja exibido ou não no menu a ação de deletar um registro.
  + Localização deste requisito no código: módulo Stockease.py. Função: login\_account(). Linha: 102 até 109.
* RF07: Armazenamento de Dados:
  + O sistema deve ser capaz de armazenar informações sobre itens em estoque e usuários.
  + Localização deste requisito no código: módulo Stockease.py. Função save(). Linha: 304 até 365. Função singup(). Linha: 69 até 87.
* RF08: Restrição de valores máximos para cadastro de produtos:
  + O sistema não deve permitir o cadastro de um produto com valor acima de 1 bilhão e quantidade maior que 1000.
  + Localização deste requisito no código: módulo Stockease.py. Função save(). Linha: 351 até 352.
* RF09: Totalização
  + O sistema deve mostrar o preço total dos produtos.
  + Localização deste requisito no código: módulo Stockease.py. Função: calculate\_total\_price(). Linha: 492 até 506.

Requisitos Não-funcionais:

* RNF01: Segurança:
  + O sistema deve ser capaz de armazenar as senhas dos usuários de forma segura no banco de dados, utilizando técnicas de hash com a biblioteca bcrypt do Python.
  + Localização deste requisito no código: módulo Stockease.py. Função: singup(). Linha: 79 até 83.
* RNF02: Usabilidade:
  + O sistema deve ser capaz de possuir uma interface de usuário intuitiva.
  + Localização deste requisito no código: este requisito é atendido com as seguintes implementações:
    - A interface de usuário é desenvolvida usando o framework Tkinter e CustomTkinter, proporcionando uma interface visual amigável.
    - A tela de login é criada com a janela top e os widgets como Label, Entry e Button são configurados com fontes, cores e posicionamento adequados para facilitar a navegação e interação do usuário.
    - Funções como center\_window\_display centralizam automaticamente a janela na tela, melhorando a usabilidade.
    - Labels e entradas são claramente definidas para campos de usuário e senha, além de botões intuitivos para login, cadastro e fechamento da aplicação.
    - Mensagens de aviso e erro são exibidas com messagebox para orientar o usuário em caso de entradas inválidas ou operações malsucedidas.
* RNF03: Desempenho:
  + O sistema deve ser capaz de lidar com um grande volume de dados de forma eficiente, garantindo um desempenho adequado durante as operações de CRUD.
  + Localização deste requisito no código: as funções do módulo StockEase.py de leitura e atualização de dados (refreshTable, read, read\_where) utilizam consultas SQL otimizadas, garantindo que os dados sejam recuperados e apresentados na interface do usuário de forma rápida. Além disso, o uso de cursor.connection.ping() no módulo StockEase.py antes das operações de banco de dados assegura que a conexão esteja ativa, prevenindo falhas e melhorando a robustez do sistema.
* RNF04: Portabilidade:
  + O sistema deve ser desenvolvido em Python, tornando-o altamente portátil e capaz de ser executado em diferentes plataformas, desde que o ambiente Python esteja instalado.
  + A interface gráfica do usuário deve ser desenvolvida utilizando o framework Tkinter.
  + O sistema deve utilizar o SGBD MySQL.
  + Localização deste requisito no código: nota-se que todo o código foi escrito em Python, utilizando a biblioteca Tkinter, para fazer a interface gráfica e a biblioteca pymysql, para fazer a conexão com o banco de dados. Além disso, as entidades do sistema foram feitas no SGBD MySQL.

Localização dos conceitos estudados na disciplina de Programação Funcional:

* Função lambda de alta ordem:
  + Módulo Stockease.py. Da linha 476 até a 489.
* Função lambda recursiva:
  + Módulo Stockease.py. Da linha 492 até a 506.
* Função lambda utilizando currying:
  + Módulo Stockease.py. Da linha 47 até a 51.
* List Comprehension dentro do escopo de uma Lambda:
  + Módulo Stockease.py. Da linha 558 até a 559.
* Dicionário dentro do escopo de uma função lambda:
  + Módulo Stockease.py. Da linha 510 até a 526.
* Functor map:
  + Módulo Stockease.py. Da linha 279 até a 280.
* Utilização de monad:
  + Módulo Stockease.py. Da linha 284 até a 292 e linha 333.

Perguntas feitas ao Chat GPT:

* Como eu posso usar recursão em uma totalização?
* Dê um exemplo de função lambda de alta ordem aplicada a um sistema de gerenciamento de estoque.
* Cite quais os requisitos funcionais e não-funcionais que um sistema de gerenciamento de estoque precisa possuir.
* Como posso melhorar a usabilidade de uma tela de login?
* Como utilizar currying para centralizar a tela de login?