

Controle de Estoque Hospitalar

Aprimorado

Este documento descreve a solução aprimorada para o gerenciamento de estoque hospitalar, desenvolvida com base no código Python fornecido e nas melhores práticas de programação e documentação.

Identificação do Grupo (Contexto Original)

As informações abaixo referem-se ao grupo que desenvolveu a versão inicial do trabalho, mantidas para referência contextual:

- **Turma:** 2ESPY
- **Integrantes:**
 - Guilherme (RA: 554962)
 - Pedro (RA: 555556)
 - Fabrício (RA: 558216)
 - Vitor (RA: 554893)
 - Matheus (RA: 555447)

Descrição do Projeto Aprimorado

Esta versão do projeto foca em um sistema unificado e aprimorado para o controle de estoque hospitalar, utilizando Python. O script

`estoque_hospitalar_aprimorado.py` implementa uma estrutura de dados baseada em lista de dicionários para representar os insumos e oferece as seguintes funcionalidades principais:

- **Gerenciamento CRUD:** Adição, leitura (individual ou total), atualização de quantidade e remoção de insumos do estoque.
- **Ordenação Eficiente:** Utiliza o algoritmo **Merge Sort** (complexidade $O(N \log N)$) para ordenar os insumos por nome, o que é um pré-requisito para a busca otimizada.
- **Busca Rápida:** Implementa a **Busca Binária** (complexidade $O(\log N)$) para localizar insumos rapidamente na lista ordenada.

- **Análise com Grafos:** Emprega a biblioteca `networkx` para modelar o estoque como um grafo. Neste modelo:
 - Cada **nó** representa um insumo, com seus atributos (local, quantidade atual, ideal, unidade).
 - As **arestas** conectam insumos que estão armazenados no mesmo local (depósito).
 - Funções analisam o grafo para identificar insumos com estoque **abaixo ou acima do ideal** e listar insumos vizinhos (no mesmo local).
- **Melhorias Gerais:** O código foi refatorado para maior clareza, robustez e eficiência, incluindo a remoção de pausas desnecessárias (`time.sleep`), adição de validações (ex: não adicionar insumos duplicados), estruturação lógica e uso de boas práticas como type hints e docstrings detalhadas.

Estrutura dos Arquivos Gerados

- `estoque_hospitalar_aprimorado.py` : O script Python principal contendo toda a lógica e funcionalidades aprimoradas.
- `README_aprimorado.md` : Documento em formato Markdown com a descrição detalhada do projeto, melhorias, funcionalidades e instruções (utilizado como base para este documento).
- `documento_final_estoque.pdf` : Este documento final em formato PDF.

Como Executar o Script

Para executar o script e observar suas funcionalidades, siga os passos:

1. **Instalar Dependência:** É necessário ter a biblioteca `networkx` instalada. Caso não a tenha, instale via pip: `bash pip install networkx`
2. **Executar o Script:** Navegue até o diretório onde salvou o arquivo e execute-o com Python 3: `bash python3 estoque_hospitalar_aprimorado.py`

O script contém um bloco de exemplo (`if __name__ == "__main__":`) que demonstra as operações de adição, busca, atualização, remoção e a análise via grafo, imprimindo os resultados diretamente no console.

Documentação do Código

O código fonte (`estoque_hospitalar_aprimorado.py`) está extensivamente documentado internamente:

- **Docstrings:** Todas as funções possuem docstrings que explicam seu propósito, os parâmetros que recebem, o que retornam e a complexidade algorítmica (quando relevante, como no Merge Sort e Busca Binária).
- **Type Hints:** As assinaturas das funções incluem anotações de tipo, facilitando o entendimento e a manutenção do código.
- **Comentários:** Comentários pontuais explicam decisões de design ou partes específicas da lógica.

Referências

- Documentação oficial do Python 3.
- Documentação da biblioteca NetworkX.
- Conceitos de algoritmos de ordenação (Merge Sort) e busca (Busca Binária).
- Materiais de referência sobre estruturas de dados e boas práticas de programação.