Relatório Técnico-Científico

Sistema de Gerenciamento de Pedidos

Tia Lu Delivery - Bahia

Aloisio Caldas da Silva Junior, Eduardo Sousa da Silva, Eveny Castro de Almeida, Iran Pablo Santos Martins e Thiago Sanches Hohlenwerger

Tutor: Lucas Almeida Silva

Sistema de Informação – Centro Universitário de Excelência (Unex) Praça José Bastos, 55 – Osvaldo Cruz, Itabuna – BA – 45600-080 – Brasil

Abstract. *Data sorting is an essential step in the development of systems that handle large lists of information, as it ensures greater efficiency, organization, and clarity in the presentation of results. This report details the implementation of the Bubble Sort algorithm, chosen to sort the orders registered in the TIA Lu Delivery project system – phase 2. In addition to presenting the theoretical foundations, the algorithm's strategies, its pseudocode, how it was integrated into the modularized system, and the behaviors observed during execution are discussed. The report includes practical examples, flowcharts, and collected screenshots, demonstrating the operation of the adopted solution.*

Resumo. *A ordenação de dados é uma etapa essencial no desenvolvimento de sistemas que manipulam grandes listas de informações, pois garante maior eficiência, organização e clareza na apresentação dos resultados. Neste relatório, é detalhada a implementação do algoritmo de ordenação Bubble Sort, escolhido para ordenar os pedidos registrados no sistema do projeto TIA Lu Delivery – etapa 2. Além de apresentar os fundamentos teóricos, são discutidas as estratégias do algoritmo, seu pseudocódigo, a forma como foi integrado ao sistema modularizado e os comportamentos observados durante a execução. O relatório inclui exemplos práticos, fluxogramas e telas coletadas, evidenciando o funcionamento da solução adotada.*

1. Introdução

A ordenação de dados é uma atividade fundamental nos sistemas computacionais modernos, sendo empregada em mecanismos de busca, geração de relatórios, listagens organizadas e análises de desempenho. Nos sistemas de informação, especialmente aqueles que lidam com cadastros, pedidos ou registros, a organização dos dados facilita a visualização e acelera o acesso a informações específicas[1].Diversos algoritmos foram propostos ao longo da história para resolver o problema da ordenação, cada um com estratégias, vantagens e limitações próprias. Entre eles, destacam-se abordagens clássicas como Bubble Sort, Selection Sort, Merge Sort e Quick Sort. Cada técnica possui custo computacional distinto e se adapta melhor a diferentes cenários[2].No contexto deste projeto, o objetivo principal do relatório é apresentar, explicar e analisar o algoritmo Bubble Sort, escolhido pela equipe Bahia como mecanismo de ordenação dos pedidos registrados no sistema de gerenciamento TIA Lu Delivery – Etapa 2. Além disso, serão demonstrados o comportamento da solução implementada e sua integração

à arquitetura refatorada do sistema, agora modularizado e com persistência de dados em arquivo JSON.

2. Fundamentação Teórica

2.1 O Problema da Ordenação

Ordenar significa reorganizar elementos de uma estrutura (lista, vetor, arquivo) segundo um critério definido, geralmente crescente ou decrescente. Esse processo, embora pareça simples, é essencial para otimizar consultas, reduzir o tempo de busca e melhorar a apresentação dos dados[3].

2.2 O Algoritmo Bubble Sort

O Bubble Sort é um dos algoritmos de ordenação mais conhecidos na computação e frequentemente utilizado em contextos educacionais por sua simplicidade e clareza de implementação[4]. Sua estratégia consiste em percorrer repetidas vezes a lista comparando pares de elementos adjacentes. Quando identifica um par fora de ordem, realiza uma troca. Esse processo se repete até que nenhum elemento seja alterado.

2.3 Características principais

Estratégia de comparação simples.

Método estável (não altera a posição relativa de elementos iguais).

Ordenação feita in-place (não exige estruturas auxiliares significativas).

Complexidade:

$O(n^2)$ no pior e médio caso;

$O(n)$ no melhor caso (lista já ordenada).

2.4 Pseudocódigo

O pseudocódigo clássico do Bubble Sort pode ser escrito assim[5]:

para i de 0 até n-1:

para j de 0 até n-1-i:

se lista[j] > lista[j+1]:

trocar(lista[j], lista[j+1])

2.5 Estratégia de Ordenação

O algoritmo faz trocas sucessivas empurrando, “borbulhando”, o maior elemento para o final da lista a cada ciclo externo, o que lhe dá o nome de Bubble. Embora seja menos eficiente que algoritmos mais modernos, demonstra bem o conceito de ordenação por comparação e é suficiente para ordenação de pequenas listas[6].

3. Metodologia

Nesta etapa do projeto, o sistema anterior foi totalmente refatorado para incluir modularização, armazenamento persistente e organização estruturada dos dados. A implementação do Bubble Sort foi integrada seguindo estes passos:

3.1 Modularização

O sistema foi dividido em arquivos independentes:

main.py — interface de console e menu principal

pedidos.py — funções de manipulação de pedidos

itens.py — manipulação de itens

json_io.py — leitura e escrita do arquivo JSON

ordenacao_bubble_sort.py — implementação do Bubble Sort

arvore_avl.py — indexador utilizado na carga dos dados

Essa separação melhora a legibilidade, manutenção e expansão futura do código[7].

3.2 Armazenamento em JSON

Todos os itens e pedidos passaram a ser persistidos no arquivo dados.json. A leitura ocorre na inicialização do programa, e a gravação após qualquer alteração relevante.

3.3 Representação dos Dados

Itens e pedidos foram convertidos para dicionários (mapas), exemplo:

```
pedido = {
"codigo": 12,
"cliente": "Maria",
"total": 54.90,
"status": "ENTREGUE"
}
```

3.4 Integração do Bubble Sort

A função de ordenação foi implementada em arquivo próprio:

```
def bubble_sort(lista):
n = len(lista)
for i in range(n - 1):
for j in range(n - 1 - i):
if lista[j]["codigo"] > lista[j+1]["codigo"]:
lista[j], lista[j+1] = lista[j+1], lista[j]
return lista
```

Na geração de relatórios, a lista de pedidos é ordenada pelo campo codigo.

4. Resultados e Discussões

4.1 Funcionamento Geral do Algoritmo

O Bubble Sort se mostrou adequado à proposta, ordenando corretamente a lista de pedidos de acordo com seus códigos, garantindo assim a organização das informações antes da exibição ao usuário.

4.2 Comportamento do Algoritmo

O algoritmo apresentou:

movimentação progressiva dos maiores valores para o final da lista;

ordenação estável e previsível;

execução mais lenta em listas maiores, característica esperada.

4.3 Demonstrações

👉 [ESPAÇO PARA PRINT DA LISTA ANTES DA ORDENAÇÃO]

👉 [ESPAÇO PARA PRINT DA LISTA DEPOIS DA ORDENAÇÃO]

👉 [ESPAÇO PARA PRINT DO RELATÓRIO FINAL]

4.4 Fluxograma

👉 [ESPAÇO PARA INSERIR O FLUXOGRAMA DO BUBBLE SORT]

5. Considerações Finais

A utilização do algoritmo Bubble Sort permitiu explorar conceitos fundamentais da ordenação computacional, além de reforçar a importância da modularização e da organização dos dados em sistemas reais.

Os principais desafios envolveram a integração do algoritmo à estrutura refatorada, a necessidade de manipulação adequada dos dados em JSON e a implementação de testes para assegurar a ordenação correta.

Com mais tempo, seria possível integrar algoritmos mais eficientes, como Merge Sort ou Quick Sort, além de explorar métricas de desempenho comparando diferentes métodos.

6. Referências

[1] SWEIGART, A. Automatize tarefas maçantes com Python. Novatec Editora, 2017.

[2] FORBELLONE, A.; EBERSPÄCHER, H. Lógica de Programação. Pearson, 2016.

[3] PAIVA, D. Estruturas de Dados em Python. Fundação Bradesco, 2020.

[4] DEV MEDIA. Algoritmos de Ordenação – Bubble Sort. DevMedia, 2021.

[5] MANZANO, J. Algoritmos: Lógica para Desenvolvimento de Programação. Saraiva, 2018.

[6] W3SCHOOLS. Sorting Algorithms. Disponível em: www.w3schools.com.

[7] SILVA, F. Modularização em Python e Boas Práticas de Desenvolvimento. USP, 2021.