

DASA – Desafio 1

Controle de Estoque com Sensores de Peso, ESP e Dashboard

DYNAMIC PROGRAMMING

Integrantes:

- Lucca Borges (RM554608)
- Ruan Vieira (RM557599)
- Rodrigo Carnevale (RM558148)

1. Hipóteses e Dados Considerados

- Origem dos dados: sensores de peso (células de carga) conectados a ESP32 via MQTT
- Precisão dos sensores: $\pm 0,02$ kg
- Intervalo de leitura: 1 leitura a cada 5 segundos
- Latência típica: até 500 ms entre ESP32 e servidor Python

2. Modelo de Dados

O fluxo de dados é:

1. JSON recebido do ESP32:

```
{  
  "sensor_id": "balanca1",  
  "produto": "arroz",  
  "peso_atual": 2.35,  
  "peso_ideal": 5.00  
}
```

2. Conversão para dicionário Python usando `json.loads`.
3. Atualização do dicionário `estoque`:

```
mensagem = json.loads(json_string)
estoque[mensagem['produto']] = [mensagem['peso_atual'],
mensagem['peso_ideal']]
```

4. 4. Estrutura em Python: `estoque: dict[str, list[float]]` onde:

- índice 0: quantidade atual
- índice 1: quantidade recomendada

3. Análise de Complexidade (O-Grande)

- `diferenca(item)`: $O(1)$
- `produtos_em_falta / produtos_sobrando`: $O(n)$
- `busca_binaria`: $O(\log n)$
- Ordenação inicial das chaves: $O(n \log n)$

Discussão:

- Busca binária reduz o tempo de busca de $O(n)$ para $O(\log n)$.
- Memoização com `@lru_cache` evita recomputações, mantendo $O(n)$.

4. Referências Bibliográficas

- Python Documentation. `functools.lru_cache`. Disponível em: <https://docs.python.org/3/library/functools.html>
- Manual do módulo HX711 (células de carga).
- HiveMQ. MQTT Essentials. Disponível em: <https://www.hivemq.com/mqtt-essentials/>