

Readme Algoritmos

Sistema de Empacotamento 3D - 4 Inteligências

![[Python]](<https://python.org>)
![[Status]](#)
![[Algoritmo]](#)

[TARGET] Visão Geral

Sistema avançado de empacotamento 3D que combina 4 inteligências especializadas para otimização de alocação de produtos em containers, considerando ergonomia biomecânica, eficiência operacional, estabilidade física e aproveitamento máximo do espaço.

[BRAIN] As 4 Inteligências

[CHART] Performance

- Taxa de Alocação: 97%+ dos produtos
- Otimização Ergonômica: Redução de 80% em esforços inadequados
- Estabilidade: 100% dos arranjos fisicamente seguros
- Eficiência: 60% algoritmo principal + 40% recuperação Greedy

[ROCKET] Instalação Rápida

```
# Clone o repositório
git clone https://github.com/vferreirall/app-ef-log.git
cd app-ef-log

# Instale dependências
pip install -r requirements.txt

# Execute o sistema
python scripts/run_pipeline.py
```

[FOLDER] Estrutura do Projeto

```
■ app-ef-log/
■ ■ ■ ■ scripts/
■ ■ ■ ■ ■ core/
■ ■ ■ ■ ■ algorithms.py      # 4 Inteligências principais
■ ■ ■ ■ ■ models.py         # Modelos de dados
■ ■ ■ ■ ■ visualization.py  # Renderização 3D
■ ■ ■ ■ ■ config.py         # Configurações
■ ■ ■ ■ ■ run_pipeline.py   # Execução principal
■ ■ ■ ■ ■ run_packing_gpu.py # Otimização GPU
■ ■ ■ ■ ■ docs/
```

```
■ ■■■ ■ ALGORITMOS_VISUAL.md # Documentação completa
■■■ ■ data/ # Dados de entrada
■■■ ■ output/ # Resultados visuais
■■■ ■ README.md # Este arquivo
```

[THEATER] Como Funciona (Analogia Simples)

Imagine organizar uma farmácia [SHOP]:

```
■ FARMÁCIA = CONTAINER
■ REMÉDIOS = PRODUTOS
■■■■ FARMACÊUTICO = OPERADOR
■ VENDAS/MÊS = DEMANDA ABC
```

[MOVIE] Fluxo de Trabalho

1. [CHART] Gerente Comercial classifica produtos por vendas
1. [DNA] Fisioterapeuta define zonas ergonômicas
1. [CONSTRUCTION] Engenheiro valida estabilidade física
1. [ROBOT] Faxineiro aproveita espaços restantes

[CONSTRUCTION] Configuração de Container

[RULER] Dimensões Padrão

- Largura: 30cm
- Profundidade: 40cm
- Altura: 50cm (dividida em 5 zonas)

[TARGET] Zoneamento Vertical

[TOOL] Uso Avançado

[CHART] Configuração ABC

```
from scripts.core.algorithms import classificar_abc_por_giro

# Configurar limites ABC
produtos_a = classificar_abc_por_giro(produtos, limite_a=50, limite_b=15)
```

[DNA] Configuração Biomecânica

```
from scripts.core.algorithms import determinar_zona_biomecanica
```

```
# Definir operador
operador = {
    'altura': 170, # cm
    'peso_max': 20 # kg
}
```

[PALETTE] Visualização 3D

```
from scripts.core.visualization import create_3d_visualization

# Gerar visualização
fig = create_3d_visualization(resultado_final)
fig.show()
```

■ Resultados Típicos

[CHECK] Produtos Alocados por Curva ABC

- Curva A: 100% alocados (prioridade máxima)
- Curva B: 98% alocados (boa prioridade)
- Curva C: 95% alocados (aproveitamento Greedy)

[TARGET] Distribuição por Zona Ergonômica

- Zona A (Premium): 85% produtos Curva A
- Zona B (Boa): 60% produtos Curva B
- Zona Γ (Aceitável): 40% produtos Curva C

[BRAIN] Para Usuários TDAH

[TARGET] Regra de Ouro

```
MAIS VENDIDO = MAIS FÁCIL DE PEGAR
MAIS PESADO = MAIS EMBAIXO
INSTÁVEL = NÃO FICA
SOBROU = GREEDY SALVA
```

[BOOKS] Documentação Completa

- Guia Visual Completo - Explicação detalhada com exemplos visuais e analogias

■ Contribuição

1. Fork o projeto
1. Crie sua feature branch (git checkout -b feature/nova-inteligencia)

1. Commit suas mudanças (git commit -m 'Adiciona nova inteligência')
1. Push para a branch (git push origin feature/nova-inteligencia)
1. Abra um Pull Request

[DOCUMENT] Licença

Este projeto está sob a licença MIT. Veja o arquivo LICENSE para detalhes.

■ Autores

- Desenvolvedor Principal: @vferreira11
- Algoritmo 4-Inteligências: Desenvolvimento próprio
- Otimização GPU: Baseado em CUDA/Numba

■ Contato

- Issues: GitHub Issues
- Documentação: docs/ALGORITMOS_VISUAL.md
- Exemplos: scripts/examples/

■ Links Rápidos

[TARGET] Sistema projetado para máxima eficiência operacional com foco em acessibilidade e compreensão.