

Otimização de Programação de Manutenção

Este projeto implementa um sistema automatizado para programação semanal de Ordens de Serviço (OS), considerando:

- Prioridades operacionais (Z, A, B, C)
- Dias de parada e operação
- Dependências entre OS (predecessoras)
- Disponibilidade de recursos por habilidade

O algoritmo determina em qual dia cada OS deve ser alocada, respeitando as restrições operacionais.

Tecnologias utilizadas

- Python 3.12+
- Pandas (tratamento e agregação dos dados)
- OpenPyXL (leitura de arquivos Excel)

Estrutura do projeto

```
desafio/
├── venv/
├── src/
│   └── solucao.py
├── data/
│   └── backlog_desafio_500.xlsx
├── docs/
│   ├── fluxograma.bpm
│   ├── fluxograma.png
│   ├── doc.pdf
│   └── doc.md
├── README.md
├── README.pdf
└── requirements.txt
```

Instalação e Configuração

1. Criar o ambiente virtual

```
python -m venv .venv
```

2. Ativar o ambiente virtual

Windows

```
.venv\Scripts\activate
```

Linux ou macOS

```
source .venv/bin/activate
```

3. Instalar dependências

```
pip install pandas openpyxl
```

Ou pode usar

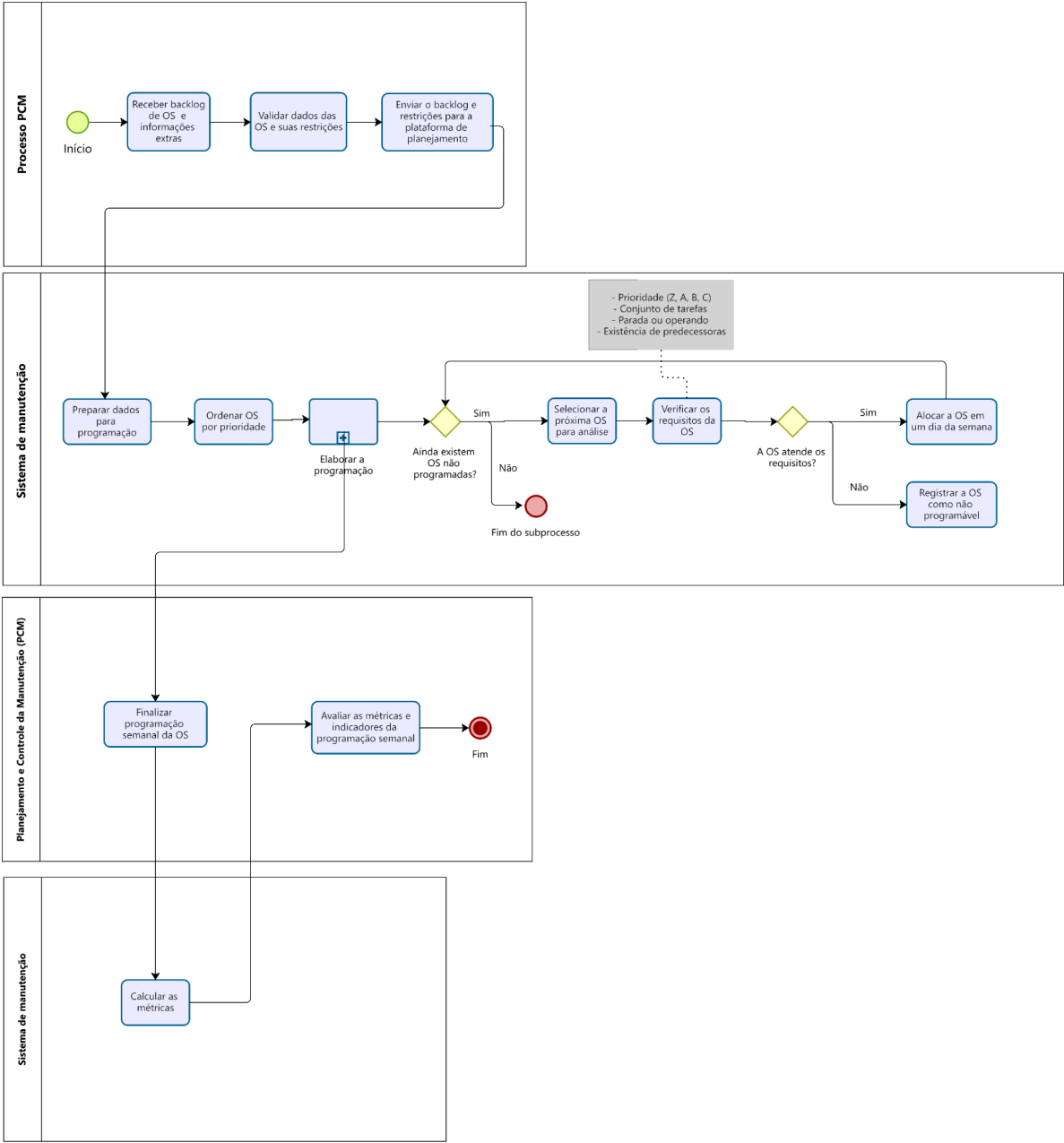
```
pip install -r requirements.txt
```

Visão Geral do Sistema

O algoritmo `create_solution()` processa as planilhas de backlog e executa automaticamente a programação das OS seguindo sete macroetapas:

1. Ler os dados do Excel
2. Calcular demanda de horas por OS e habilidade
3. Calcular duração contínua por OS
4. Ordenar por prioridade e tempo
5. Avaliar restrições da OS (parada e predecessor)
6. Verificar disponibilidade de recursos
7. Programar ou rejeitar a OS e calcular métricas finais

Fluxograma do processo - BPMN



Estrutura da saída da solução

{

```
"solution": {
  "OS_10": "1",
  "OS_381": "2",
  ...
},
"metrics": {
  "n_os": 32,
  "n_Z": 10,
  "n_A": 10,
  "n_B": 7,
  "n_C": 5,
  "utilization": {
    "Mecânico": "86.46%",
    "Elétrico": "86.51%",
    "Soldador": "92.86%",
    "Lubrificador": "93.38%"
  }
}
}
```

Documentação complementar

/docs/doc.md