

Como Executar o Projeto

Pré-requisitos

- Python 3.8 ou superior
- Oracle Database (pode ser Oracle XE)

Início Rápido

Opção 1: Script Automático (RECOMENDADO)

```
.\iniciar.ps1
```

Este script faz tudo automaticamente:

- ✓ Cria o ambiente virtual (se não existir)
- ✓ Ativa o ambiente virtual
- ✓ Instala todas as dependências
- ✓ Verifica se o `.env` está configurado
- ✓ Inicia o sistema

Apenas certifique-se de configurar o `config\.env` antes!

Opção 2: Instalação Manual

1. Clone o repositório e navegue até a pasta:

```
cd Fase6Modificada
```

2. Crie e ative um ambiente virtual:

```
python -m venv .venv  
.\venv\Scripts\Activate.ps1
```

3. Instale as dependências:

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

4. Configure o banco de dados:

- Edite o arquivo `config/.env` com suas credenciais Oracle
- Use o arquivo `config/.env.example` como referência

5. Execute o sistema:

```
python -m src.main
```

Executando o Projeto

Executar o sistema principal:

```
python -m src.main
```

Executar testes de conexão:

```
python -m scripts.teste_conexao
```

Estrutura do Projeto

- **src/** - Código fonte principal
 - **main.py** - Programa principal com menu interativo
 - **banco.py** - Conexão e operações com Oracle Database
 - **colheita.py** - Registro de colheitas (apenas perdas)
 - **relatorios.py** - Geração de relatórios em JSON, TXT e CSV
- **scripts/** - Scripts auxiliares
 - **utils.py** - Funções utilitárias (validação, formatação de tabelas)
 - **teste_conexao.py** - Teste de conexão com banco
- **config/** - Configurações
 - **.env** - Variáveis de ambiente (não commitar!)
 - **.env.example** - Exemplo de configuração
- **document/** - Documentação complementar

Funcionalidades

1. Registrar Colheita

O sistema permite registrar dados de perdas na colheita:

- Tipo (manual ou mecânica)
- Produtividade estimada e real (t/ha)
- Valor por tonelada

Importante: Apenas registros com **perda** são salvos. Se a produtividade real superar a estimada, o sistema informa que houve ganho e não salva o registro.

2. Gerar Relatórios

Gera automaticamente relatórios em três formatos na pasta `document/`:

- `relatorio.json` - Dados estruturados
- `relatorio.txt` - Relatório completo com sumário
- `relatorio.csv` - Planilha para Excel

Notas Importantes

- O arquivo `.env` com credenciais reais **NÃO** deve ser commitado no Git
- Os relatórios são salvos automaticamente na pasta `document/`
- Execute sempre a partir da raiz do projeto usando `python -m`
- O sistema **só registra perdas**: se a produtividade real for maior que a estimada, o registro não é salvo
- Verifique mensagens de sucesso/erro ao registrar colheitas:
 - [OK] = dados salvos com sucesso
 - [ERRO] = problema na conexão ou banco de dados