

# FIAP - Faculdade de Informática e Administração Paulista

---



## Fase 3: Cap 1 - Etapas de uma Máquina Agrícola

---

DeepThinker's

### Integrantes Grupo 11:

- André Pessoa Gaidzakian - RM567877
- Erick Prados Pereira - RM566833
- Guilherme Ferreira Santos - RM568523
- Viviane de Castro - RM567367

### Professores:

Tutor(a)

- Sabrina Otoni

Coordenador(a)

- André Godoi Chiovato

### Descrição

### [Link do GitHub](#)

#### 1. Introdução

O agronegócio brasileiro é um dos setores mais dinâmicos e estratégicos da economia nacional, representando cerca de 27% do PIB (IBGE, 2024). A modernização do campo, impulsionada por tecnologias digitais, sensores e automação, permite ganhos de produtividade, sustentabilidade e eficiência. Neste contexto, a FarmTech Solutions, startup fictícia, desenvolveu um sistema de monitoramento e automação agrícola, integrando sensores ambientais, lógica de decisão e banco de dados Oracle para armazenamento e análise dos dados coletados.

Este relatório detalha o processo de importação dos dados coletados na Fase 2 para o Oracle SQL Developer, explorando as etapas, boas práticas e evidências do funcionamento do sistema.

#### 2. Objetivo

Demonstrar, de forma prática e documentada, a importação, consulta e manipulação dos dados agrícolas simulados (Fase 2) em um banco de dados Oracle, utilizando o Oracle SQL Developer, conforme orientações do PBL do curso de Inteligência Artificial.

### 3. Materiais Utilizados

- **Base de dados:** dados\_agro.xlsx (simulada a partir de dados plausíveis do agro brasileiro, conforme CONAB, IBGE, Embrapa, etc.)
- **Ferramenta de banco de dados:** Oracle SQL Developer
- **Documentação e códigos:** Repositório GitHub do grupo
- **Sistema de sensores:** ESP32, sensores DHT22, botões NPK, LDR (pH), módulo relé (detalhado na Fase 2)
- **Códigos de integração:** C/C++ (ESP32) e Python (API clima)

### 4. Vídeo do projeto

## 📁 Estrutura de pastas

Dentre os arquivos e pastas presentes na raiz do projeto, definem-se:

- **assets:** aqui estão os arquivos relacionados a elementos não-estruturados deste repositório, como imagens.
- **config:** Posicione aqui arquivos de configuração que são usados para definir parâmetros e ajustes do projeto.
- **document:** aqui estão todos os documentos do projeto que as atividades poderão pedir. Na subpasta "other", adicione documentos complementares e menos importantes.
- **scripts:** Posicione aqui scripts auxiliares para tarefas específicas do seu projeto. Exemplo: deploy, migrações de banco de dados, backups.
- **src:** Todo o código fonte criado para o desenvolvimento do projeto ao longo das 7 fases.
- **README.md:** arquivo que serve como guia e explicação geral sobre o projeto (o mesmo que você está lendo agora).

## 🔧 Como executar o código

### Pré-requisitos

Antes de executar o projeto, certifique-se de ter instalado:

- **Python 3.8 ou superior** - [Download Python](#)
- **Oracle Database** - Oracle XE 21c ou superior ([Download Oracle XE](#))
- **Oracle SQL Developer** (opcional, para gerenciar o banco) - [Download SQL Developer](#)
- **Git** - Para clonar o repositório

### Bibliotecas Python utilizadas

- **oracledb** - Conexão com Oracle Database

- `python-dotenv` - Gerenciamento de variáveis de ambiente
  - `tabulate` - Formatação de tabelas no terminal
- 

## Passo 1: Clonar o Repositório

```
git clone https://github.com/DeepThinker-s/Fase3-Cap1-Etapas-de-uma-maquina-agricola
```

```
cd Fase3-Cap1-Etapas-de-uma-maquina-agricola
```

## Passo 2: Configurar Ambiente Virtual

Crie e ative um ambiente virtual Python para isolar as dependências:

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

**Nota:** Se houver erro de política de execução, execute: `Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Scope CurrentUser`

## Passo 3: Instalar Dependências

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

## Passo 4: Configurar o Banco de Dados Oracle

### 1. Copie o arquivo de exemplo de configuração:

```
copy config\.env.example config\.env
```

### 2. Edite o arquivo `config/.env` com suas credenciais Oracle:

```
DB_USER=seu_usuario  
DB_PASSWORD=sua_senha  
DB_DSN=localhost:1521/XE
```

**Importante:** O arquivo `.env` contém credenciais sensíveis e **NÃO** deve ser commitado no Git (já está no `.gitignore`).

### 3. Teste a conexão com o banco:

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

Passo 5: Executar o Sistema

### Executar o programa principal (menu interativo):

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

O sistema apresentará um menu com as seguintes opções:

- **1. Registrar colheita** - Cadastra nova colheita no banco
- **2. Gerar relatório** - Gera relatórios em JSON, TXT e CSV
- **3. Sair** - Encerra o sistema

---

## Funcionalidades Principais

### Registrar Colheita

Permite registrar dados de colheita com informações de:

- Tipo de colheita (manual ou mecânica)
- Produtividade estimada (t/ha)
- Produtividade real (t/ha)
- Valor por tonelada (R\$)

**Observação importante:** O sistema **apenas registra perdas**. Se a produtividade real for maior que a estimada (ganho), o sistema informará o usuário e **não salvará** o registro, pois o foco é monitorar perdas agrícolas. Verifique as mensagens de confirmação:

- [OK] = Dados salvos com sucesso
- [ERRO] = Problema na conexão/banco de dados

### Gerar Relatórios

Gera relatórios automáticos em três formatos:

- **JSON** ([document/relatorio.json](#)) - Dados estruturados
- **TXT** ([document/relatorio.txt](#)) - Relatório completo com sumário executivo
- **CSV** ([document/relatorio.csv](#)) - Planilha para análise

---

## Estrutura de Execução por Fase

### Fase 1 - Coleta de Dados com Sensores

- Sistema físico com ESP32 e sensores (DHT22, LDR, botões NPK)
- Código Arduino/C++ para leitura de sensores

## Fase 2 - Integração com Oracle Database

- Conexão com Oracle Database via `oracledb`
- CRUD completo de colheitas
- Geração de relatórios analíticos
- Sistema de menu interativo

## 📷 Evidências de Execução

### Evidência 1 - Exportação do relatório no programa

```
> TERMINAL
⚠ [4/4] Verificando configuração...
    Configuração encontrada!

=====
Iniciando o sistema...
=====

*** Monitoramento de Perdas na Colheita ***
1. Registrar colheita
2. Gerar relatório
3. Sair

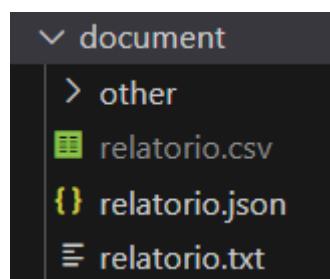
Opção: 2

*** REGISTROS ***
+---+-----+-----+-----+-----+-----+-----+-----+
| ID | PROD_ESTIMADA | TIPO_COLHEITA | PROD_REAL | VALOR_POR_TONELADA | PERDA_PERCENTUAL | PREJUIZO | DATA_HORA |
+---+-----+-----+-----+-----+-----+-----+-----+
| 21 | 11250 | Manual | 1 | 3 | 99.99 | 33747 | 2025-10-05 17:37:57 |
+---+-----+-----+-----+-----+-----+-----+-----+
| 41 | 20000 | Manual | 11250 | 150 | 43.75 | 1.3125e+06 | 2025-10-05 17:40:49 |
+---+-----+-----+-----+-----+-----+-----+-----+
| 42 | 12000 | Manual | 11250 | 150 | 6.25 | 112500 | 2025-10-05 17:41:31 |
+---+-----+-----+-----+-----+-----+-----+-----+
| 61 | 100 | Mecanica | 85 | 200 | 15 | 3000 | 2025-11-02 18:51:26 |
+---+-----+-----+-----+-----+-----+-----+-----+
| 81 | 1200 | Manual | 560 | 90 | 53.33 | 57600 | 2025-11-02 19:30:10 |
+---+-----+-----+-----+-----+-----+-----+-----+

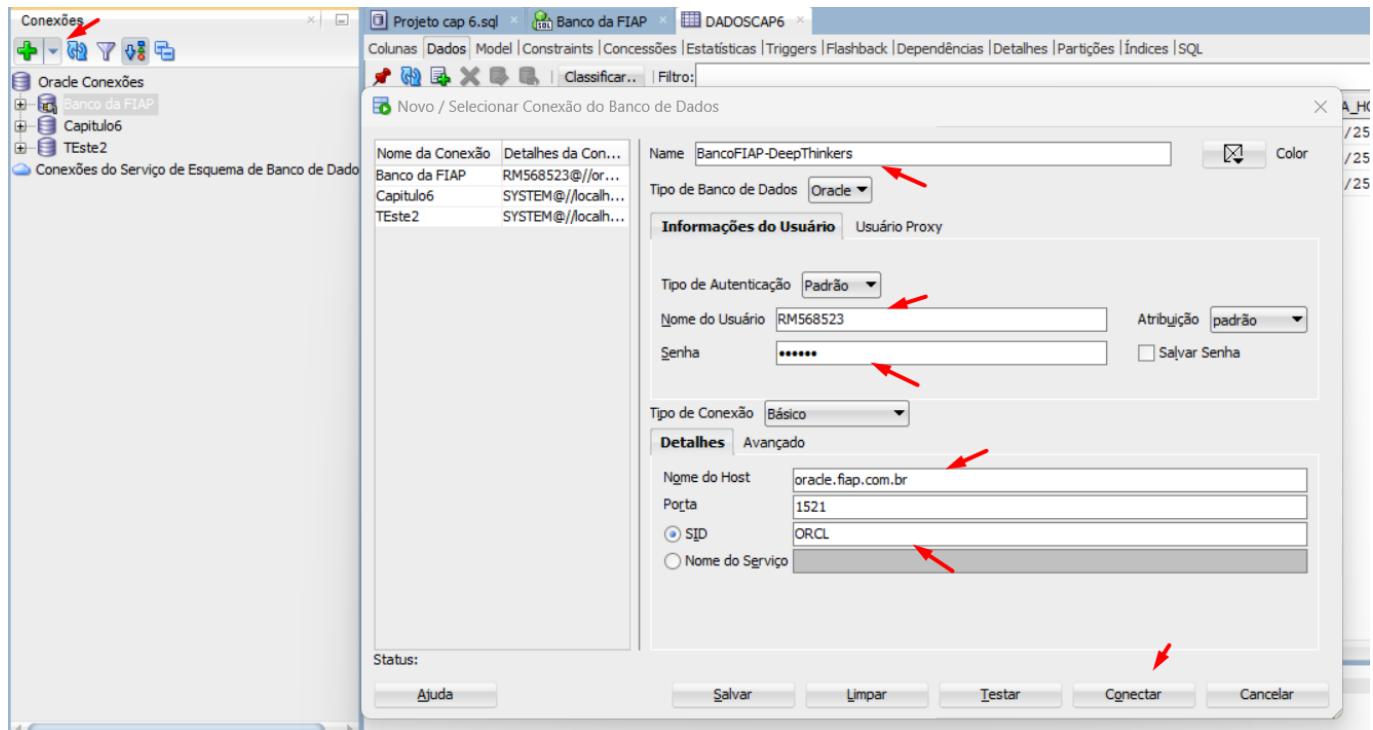
Total: 5 | Perda média: 43.66% | Prejuízo: R$ 1,519,347.00

[OK] Relatórios gerados com sucesso!
Localização: document/
- relatorio.json (dados estruturados)
- relatorio.txt (relatório completo)
- relatorio.csv (planilha)
```

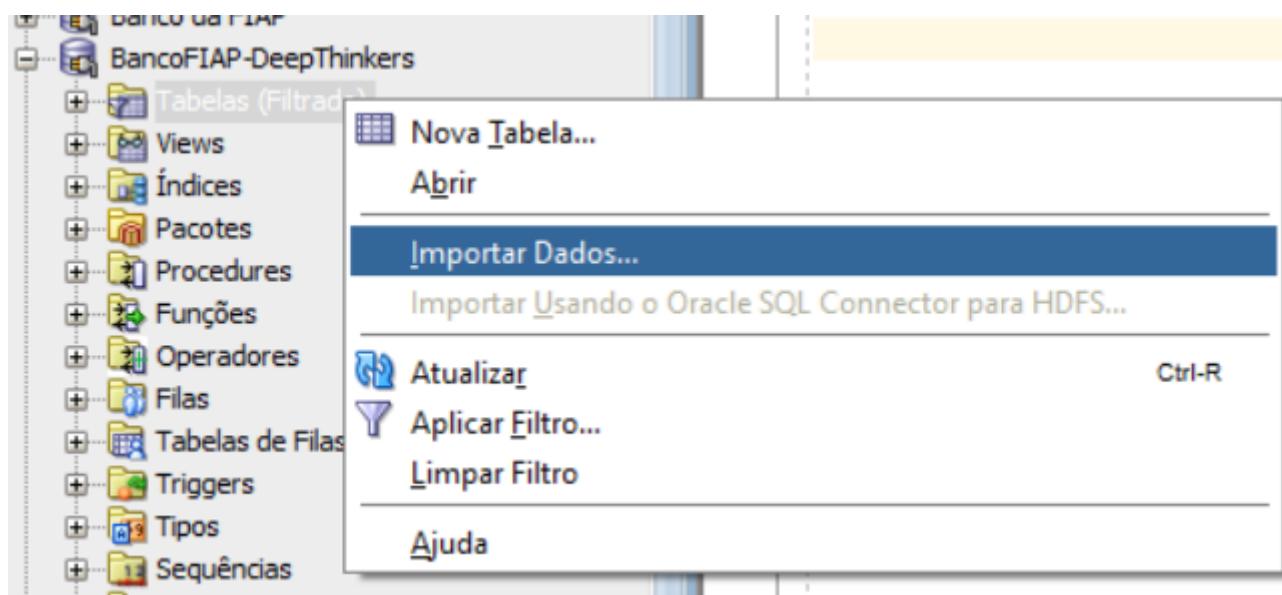
### Evidência 2 - Local dos arquivos



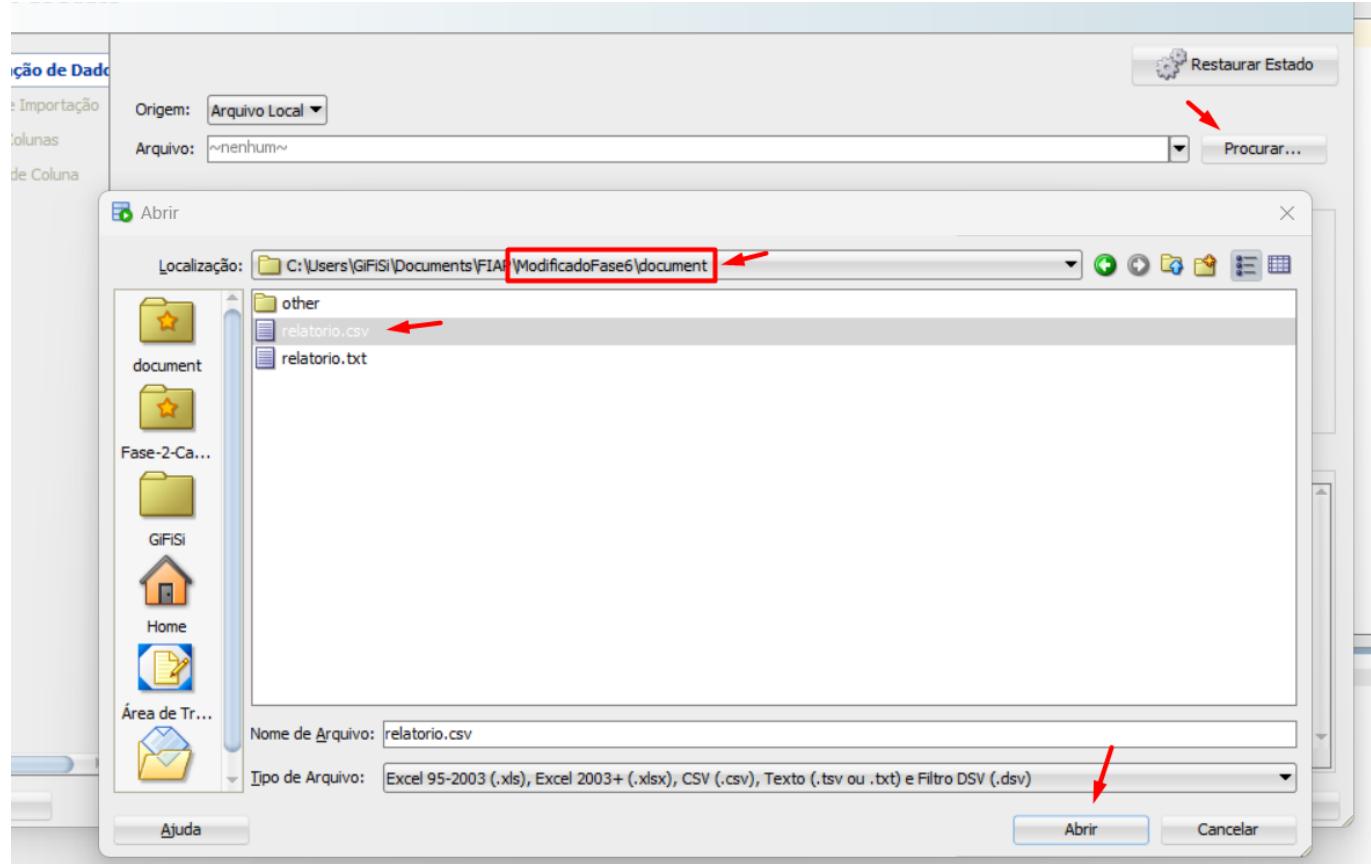
### Evidência 3 - Conexão com o banco de dados da FIAP via Oracle



#### Evidência 4 - Conectado e importando dados para a DB



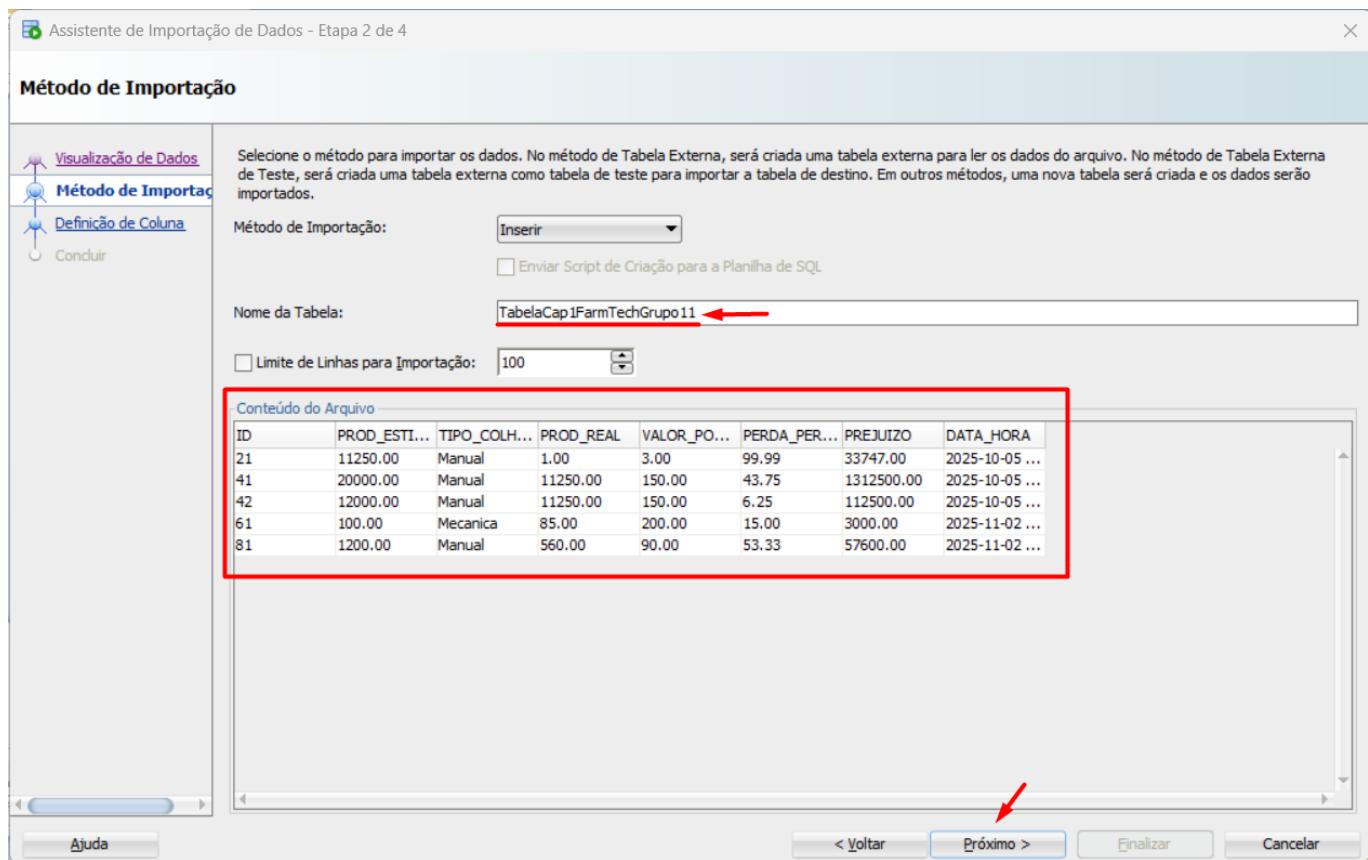
#### Evidência 5 - Buscando arquivos de registro



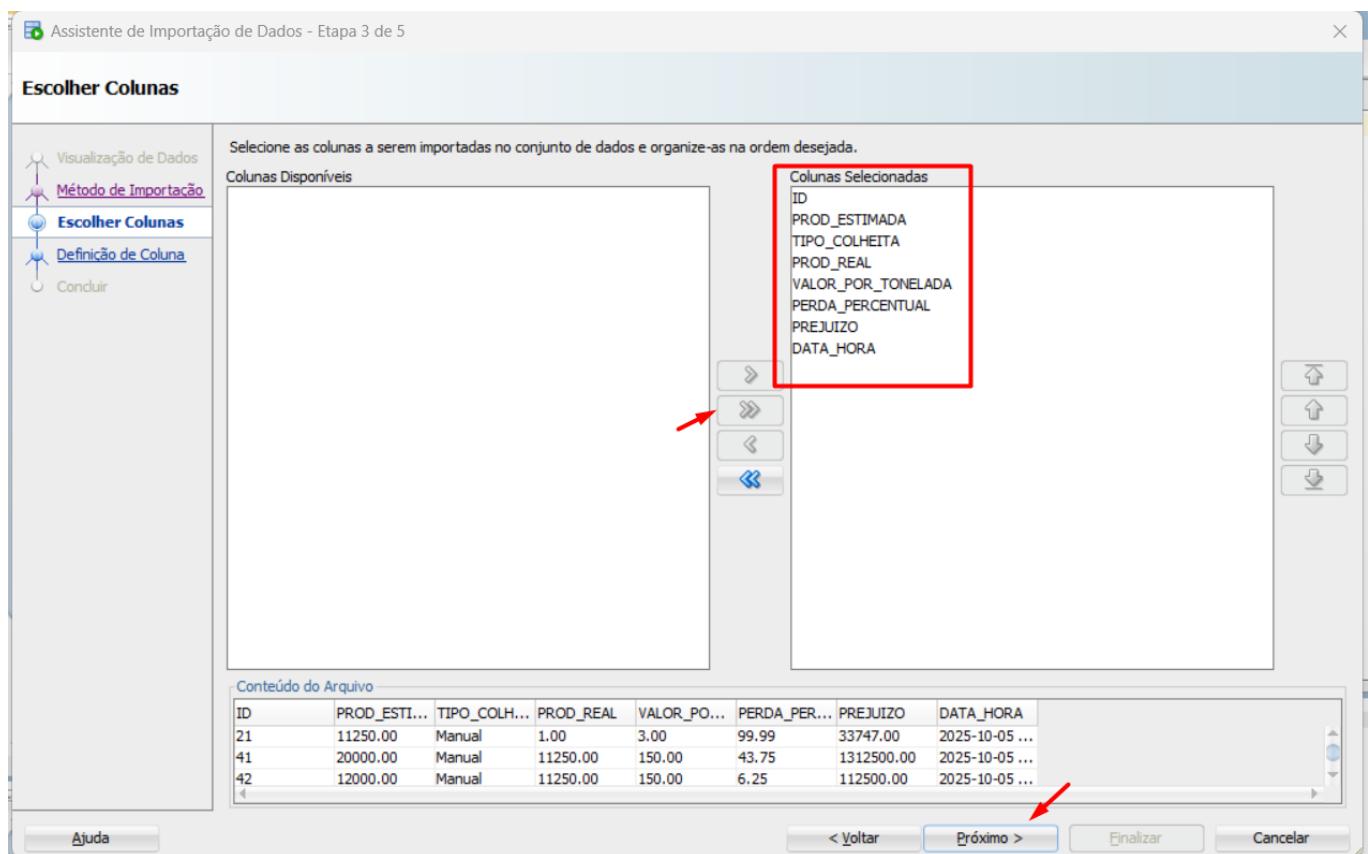
## Evidência 6 - Verificando dados

ID	PROD_ESTI...	TIPO_COLH...	PROD_REAL	VALOR_PO...	PERDA_PER...	PREJUIZO	DATA_HORA
21	11250.00	Manual	1.00	3.00	99.99	33747.00	2025-10-05 ...
41	20000.00	Manual	11250.00	150.00	43.75	1312500.00	2025-10-05 ...
42	12000.00	Manual	11250.00	150.00	6.25	112500.00	2025-10-05 ...
61	100.00	Mecanica	85.00	200.00	15.00	3000.00	2025-11-02 ...
81	1200.00	Manual	560.00	90.00	53.33	57600.00	2025-11-02 ...

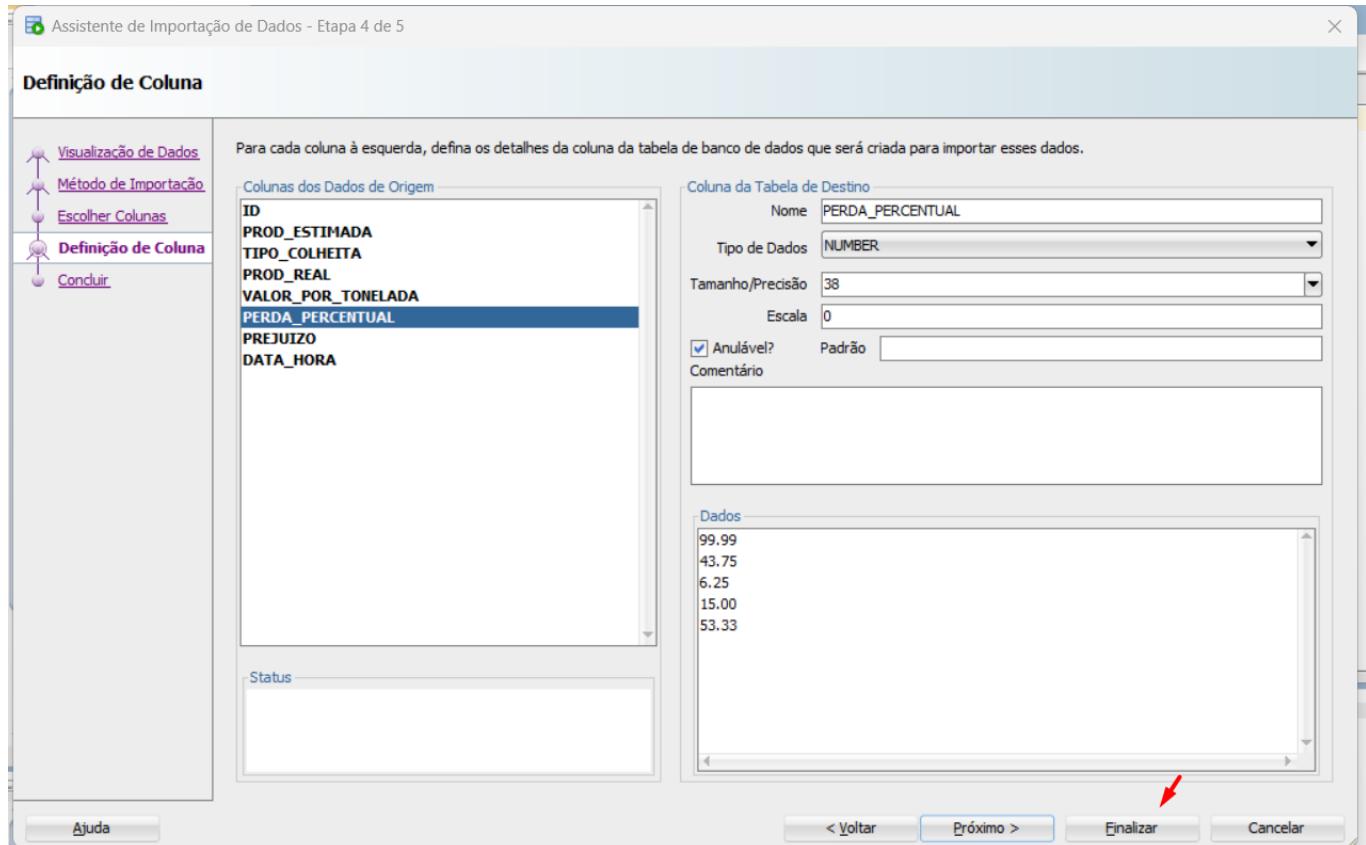
## Evidência 7 - Criando tabela



## Evidência 8 - Confirmando colunas



## Evidência 9 - Finalizando exportação



## Evidência 10 - Dados importados

ID	PROD_ESTIMADA	TIPO_COLHEITA	PROD_REAL	VALOR_POR_TONELADA	PERDA_PERCENTUAL	PREJUIZO	DATA_HORA
1	21	1125000 Manual	100	300	9999	3374700	05/10/25
2	41	2000000 Manual	1125000	15000	4375	131250000	05/10/25
3	42	1200000 Manual	1125000	15000	625	11250000	05/10/25
4	61	10000 Mecanica	8500	20000	1500	300000	02/11/25
5	81	120000 Manual	56000	9000	5333	5760000	02/11/25

## Comandos Úteis

### Verificar versão do Python:

```
python --version
```

### Listar pacotes instalados:

```
pip list
```

### Atualizar pip:

```
python -m pip install --upgrade pip
```

### Desativar ambiente virtual:

```
deactivate
```

## Solução de Problemas

### Erro ao ativar ambiente virtual:

- Execute: `Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Scope CurrentUser`

### Erro de conexão Oracle:

- Verifique se o Oracle Database está rodando
- Confirme as credenciais no arquivo `.env`
- Teste com: `python -m scripts.teste_conexao`

### Módulo não encontrado:

- Certifique-se de que o ambiente virtual está ativado
- Reinstale as dependências: `pip install -r requirements.txt`

## Histórico de lançamentos

- 2.0.0 - 02/11/2025 (Versão Atual)
  - **Refatoração completa baseada no feedback do professor**
  - Migração de `cx_Oracle` para `oracledb` (biblioteca moderna)
  - Implementação de variáveis de ambiente (`.env`) para credenciais Oracle
  - Tratamento robusto de exceções (try/except) em todas operações de I/O e banco
  - Sistema de commit/rollback para garantir integridade transacional
  - Funções modulares com docstrings, parâmetros e retornos bem definidos
  - Operações CRUD completas em memória com listas/dicionários
  - Formatação tabular de consultas (biblioteca `tabulate`)
  - Geração de relatórios em três formatos: JSON, TXT e CSV
  - Estrutura modular organizada (`src/`, `config/`, `document/`, `scripts/`)
  - Scripts de teste de conexão e utilitários
  - Documentação completa de execução no README
  - Evidências de execução com exemplos de relatórios
- 1.0.0 - 12/10/2025 (Primeira Entrega)
  - Versão inicial do sistema de monitoramento agrícola
  - Integração básica com Oracle Database
  - Problema bem definido e solução coerente para o agronegócio

- Sistema de colheita com registro de perdas
- Geração de relatórios em TXT e JSON
- Organização modular do código
- Persistência básica de dados
- Menu interativo para operações do sistema
- **Recebeu ponto extra por equipe com 4 integrantes**

## Licença



MODELO GIT FIAP por [Fiap](#) está licenciado sobre [Attribution 4.0 International](#).