

# Especificações do Projeto Final

---

## Programação para Ciência de Dados - MBA Ciência de Dados

**Disciplina:** Programação para Ciência de Dados

**Instrutor:** Cássio Pinheiro

**Universidade de Fortaleza (UNIFOR)**

---

### Objetivo do Projeto Final

Desenvolver um projeto prático completo que integre os conhecimentos adquiridos nos **Módulos 1, 2 e 3** da disciplina, demonstrando domínio em:

- Fundamentos de Python (Módulo 1)
- Manipulação de dados com Pandas e NumPy (Módulo 2)
- Visualização de Dados (Módulo 3)

O projeto deve ser funcional, bem documentado e publicado em um repositório público no GitHub.

---

### Formação de Grupos

- **Número máximo de integrantes:** 3 pessoas por grupo
  - **Número mínimo de integrantes:** 1 pessoa (trabalho individual permitido)
  - **Formação:** Os grupos devem ser formados pelos próprios alunos
  - **Divisão de trabalho:** Deve estar claramente documentada no README do projeto
- 

### 17 Prazos Importantes

Entrega do README com Especificações

- **Prazo:** Até **8/11/2025**
- **Enviar para o email** **cassiopo7@gmail.com**
- **Apenas 1 membro por equipe precisa enviar**
- **Conteúdo:** README completo com todas as especificações do projeto
- **Formato:** Markdown (.md) no repositório GitHub

Entrega Final do Projeto

- **Data limite:** **14/11/2025, às 23:59**
  - **Formato:** Repositório público no GitHub
  - **Link:** Deve estar disponível no README desde a primeira entrega
- 

### Conteúdo a Ser Cobrado

O projeto deve demonstrar o uso dos seguintes conceitos e bibliotecas:

#### Módulo 1 - Fundamentos de Programação com Python

- Tipos de dados básicos (int, float, str, bool)

- Estruturas de dados (list, dict, tuple, set)
- Estruturas de controle (if/else, for, while)
- Funções e documentação (docstrings)
- Compreensões de lista e dicionário
- Funções lambda, map, filter, reduce
- Manipulação de arquivos (leitura/escrita com `open()`)
- Tratamento básico de erros

## ✓ Módulo 2 - Manipulação Avançada com Pandas e NumPy

- Criação e manipulação de arrays NumPy
- Operações vetorizadas
- Indexação e fatiamento avançado
- Operações estatísticas básicas
- Criação e manipulação de DataFrames e Series
- Leitura de arquivos CSV
- Limpeza e tratamento de dados (valores ausentes, duplicatas)
- Seleção e filtragem de dados
- Agrupamentos e agregações (`groupby()`)
- Operações temporais (se aplicável)
- Pivot tables e transformações de dados
- Operações com strings em DataFrames

## ✓ Módulo 3 - Visualização de Dados

- Visualizações básicas com Matplotlib
  - Gráficos de linha, barras, histogramas
  - Subplots e múltiplos gráficos
  - Customização (cores, labels, títulos)
- Visualizações estatísticas com Seaborn
  - Box plots, violin plots
  - Heatmaps de correlação
  - Gráficos de distribuição

---

## Estrutura Obrigatória do README

O README do projeto deve conter **TODOS** os seguintes tópicos:

### 1. Cabeçalho e Informações Básicas

```
# Nome do Projeto

**Disciplina:** Programação para Ciência de Dados
**Curso:** MBA Ciência de Dados – UNIFOR
**Instrutor:** Cássio Pinheiro
**Integrantes:**
- Nome Completo 1 (Matrícula)
- Nome Completo 2 (Matrícula)
- Nome Completo 3 (Matrícula)

**Repositório GitHub:** [Link completo do repositório público]
**Data de Entrega:** 14/11/2024
```

## 2. 🎯 Objetivo do Projeto

- Descrição clara do propósito do projeto
- Problema que o projeto resolve
- Público-alvo ou contexto de aplicação

## 3. 📊 Diagrama de Contexto (Mermaid)

- Diagrama C4 de contexto mostrando o sistema e seus relacionamentos
- Pode usar diagramas Container ou Component se aplicável

## 4. 🛠️ Funcionalidades Implementadas

- Lista detalhada de todas as funcionalidades
- Descrição do que cada funcionalidade faz
- Indicação de quais módulos foram utilizados em cada funcionalidade

## 5. 📦 Estrutura de Dados

- Modelos de dados utilizados
- Exemplos de estruturas de entrada
- Exemplos de estruturas de saída
- Formato dos arquivos de dados (CSV, TXT, etc.)

## 6. 🖥️ Requisitos Técnicos

- Versão do Python utilizada
- Bibliotecas e dependências (com versões)
- Requisitos de sistema (se houver)
- Como instalar as dependências

## 7. 🚀 Como Executar o Projeto

- Passo a passo para instalação
- Como executar o código principal
- Exemplos de uso
- Comandos necessários

## 8. 📈 Análises Realizadas

- Descrição das análises realizadas
- Principais insights encontrados
- Visualizações criadas e seus propósitos
- Estatísticas calculadas

## 9. 📁 Estrutura do Projeto

```
nome_projeto/  
├── README.md  
├── requirements.txt  
├── main.py (ou arquivo principal)  
├── dados/  
│   ├── arquivos de dados  
├── src/ (opcional)  
│   ├── módulos do projeto  
├── relatorios/ (opcional)  
│   └── relatórios gerados
```

```
└─ notebooks/ (opcional)
   └─ notebooks de análise
```

10. 📸 Capturas de Tela / Exemplos de Saída

- Imagens das visualizações geradas
- Exemplos de saída do programa
- Interface (se houver)

11. ✏️ Testes Realizados

- Cenários testados
- Resultados obtidos
- Validações implementadas

12. 📖 Referências e Bibliografia

- Documentação consultada
- Tutoriais utilizados
- Datasets utilizados (com links)

13. 🤝 Contribuições dos Integrantes

- Divisão de trabalho
- Responsabilidades de cada integrante
- Commits principais de cada membro

14. 🔮 Próximos Passos / Melhorias Futuras

- Funcionalidades que poderiam ser adicionadas
- Melhorias técnicas possíveis
- Expansões do projeto

## 📊 Normas de Avaliação

O projeto será avaliado em **10 pontos**, distribuídos da seguinte forma:

📄 Documentação (2 pontos)

A documentação será avaliada considerando:

Critério	Pontuação	Descrição
Qualidade do README	0.5 pontos	README completo, bem formatado, com todos os tópicos obrigatórios preenchidos
Alinhamento com o Projeto	0.75 pontos	Documentação reflete fielmente o que foi implementado; exemplos correspondem ao código
Expressão do Propósito	0.75 pontos	Documentação comunica claramente o objetivo, funcionalidades e valor do projeto

Critérios de Desconto:

- README incompleto (faltam tópicos obrigatórios): -0.2 pontos
- Exemplos não funcionam ou não correspondem ao código: -0.2 pontos
- Erros de formatação ou português que dificultam leitura: -0.1 pontos

- Falta de clareza na explicação do propósito: -0.2 pontos

## Implementação Técnica (8 pontos)

A implementação será avaliada considerando:

### 1. Uso dos Conceitos dos Módulos (3 pontos)

Módulo	Pontuação	Critérios
Módulo 1 - Fundamentos	1.0 ponto	Uso adequado de estruturas de dados, funções bem definidas, compreensões, manipulação de arquivos
Módulo 2 - Pandas/NumPy	1.0 ponto	Manipulação eficiente de DataFrames, operações de agrupamento, limpeza de dados, operações vetorizadas
Módulo 3 - Visualização	1.0 ponto	Visualizações relevantes e bem formatadas, uso adequado de Matplotlib e Seaborn

#### Critérios de Desconto:

- Uso mínimo ou inadequado dos conceitos: -0.3 pontos por módulo
- Código que poderia usar conceitos do módulo mas não usa: -0.2 pontos
- Implementações ineficientes (ex: loops quando poderia usar vetorização): -0.2 pontos

### 2. Qualidade do Código (2 pontos)

Aspecto	Pontuação	Descrição
Organização e Estrutura	0.5 pontos	Código bem organizado, funções com responsabilidades claras, modularização adequada
Documentação do Código	0.5 pontos	Docstrings nas funções principais, comentários onde necessário, código legível
Tratamento de Erros	0.5 pontos	Validação de entradas, tratamento de casos extremos, mensagens de erro claras
Boas Práticas	0.5 pontos	Nomes de variáveis descritivos, código limpo, seguimento de convenções Python

#### Critérios de Desconto:

- Código desorganizado ou difícil de entender: -0.3 pontos
- Falta de documentação (docstrings): -0.2 pontos
- Sem tratamento de erros ou validações: -0.2 pontos
- Código duplicado ou não reutilizável: -0.2 pontos

### 3. Funcionalidades Implementadas (2 pontos)

Critério	Pontuação	Descrição
Compleitude das Funcionalidades	1.0 ponto	Todas as funcionalidades planejadas foram implementadas e funcionam corretamente
Robustez e Confiabilidade	0.5 pontos	Sistema funciona com diferentes entradas, trata casos extremos, não quebra facilmente
Complexidade Adequada	0.5 pontos	Projeto demonstra nível adequado de complexidade para o escopo do curso

**CrITÉrios de Desconto:**

- Funcionalidades planejadas não implementadas: -0.3 pontos por funcionalidade importante
- Bugs críticos que impedem execução: -0.5 pontos
- Sistema quebra com entradas inválidas: -0.2 pontos
- Projeto muito simples para o escopo: -0.3 pontos

**4. Visualizações (1 ponto)**

CrITÉrio	Pontuação	Descrição
Relevância das Visualizações	0.5 pontos	Visualizações relevantes para o problema, insights interessantes
Qualidade das Visualizações	0.5 pontos	Gráficos bem formatados, informativos, com títulos e labels adequados

**CrITÉrios de Desconto:**

- Visualizações sem contexto ou pouco informativas: -0.2 pontos
- Gráficos mal formatados ou sem labels: -0.2 pontos
- Visualizações superficiais ou sem insights: -0.2 pontos
- Falta de diversidade nos tipos de visualização: -0.1 pontos

**Checklist de Entrega**

Antes de submeter o projeto, verifique:

**Documentação**

- ☐ README completo com todos os tópicos obrigatórios
- ☐ Link do repositório GitHub público funcionando
- ☐ Informações dos integrantes corretas
- ☐ Exemplos de uso funcionando
- ☐ Estrutura do projeto documentada
- ☐ Capturas de tela ou exemplos de saída incluídos

**Código**

- ☐ Código funcional e executável
- ☐ `requirements.txt` com todas as dependências
- ☐ Código comentado e documentado
- ☐ Tratamento de erros implementado
- ☐ Validações de entrada implementadas

**Conteúdo dos Módulos**

- ☐ Uso adequado de conceitos do Módulo 1
- ☐ Uso adequado de Pandas e NumPy (Módulo 2)
- ☐ Visualizações com Matplotlib e Seaborn (Módulo 3)

**Repositório GitHub**

- ☐ Repositório público
- ☐ README.md na raiz
- ☐ Código organizado em pastas adequadas
- ☐ Arquivo `.gitignore` presente (se necessário)
- ☐ Commits com mensagens descritivas

- ☐ Histórico de commits mostra trabalho colaborativo (se em grupo)

---

## Observações Importantes

1. **Plágio:** Projetos copiados de outros alunos ou da internet resultarão em nota zero para todos os envolvidos.
2. **Originalidade:** Embora possam usar projetos da pasta `projetos/modulos_1` como referência, o projeto deve ter implementação própria e melhorias significativas.
3. **Colaboração:** Em grupos, todos os membros devem contribuir significativamente. A divisão de trabalho deve estar clara no README.
4. **Prazo:** Projetos entregues após o prazo terão desconto de 1 ponto por dia de atraso, até máximo de 3 dias. Após isso, não serão aceitos.
5. **Dúvidas:** Em caso de dúvidas sobre os critérios de avaliação ou requisitos, consultar o instrutor com antecedência.

---

## Exemplos de Projetos Aceitáveis

Os projetos podem ser baseados nos exemplos da pasta `projetos/modulos_1`, mas devem incluir:

- **Expansão significativa:** Adicionar funcionalidades além do básico
- **Análises com Pandas:** Manipulação avançada de dados
- **Visualizações:** Múltiplos gráficos e visualizações relevantes
- **Dataset real:** Usar dados reais ou simulados de forma adequada
- **Documentação completa:** README profissional e detalhado

---

## Contato

**Dúvidas:** Prof. Cássio Pinheiro

**E-mail:** cassiopo7@gmail.com

**Disciplina:** Programação para Ciência de Dados - MBA Ciência de Dados

---

**Boa sorte com o projeto final!** 🚀

---





