

Departamento de Engenharia de Sistemas e Computação

Lançamento do Projeto Final

Temas, grupos, escopo e cronograma

João Araujo

Algoritmos Computacionais I

Aula 26

O projeto final

- Programa **completo** que resolve um problema real da sua área de engenharia
- Em **grupos de 2 ou 3 alunos**
- Desenvolvido em **etapas**, com feedback contínuo em cada aula
- Integra tudo o que vimos: funções, listas e dicionários, arquivos, exceções, NumPy, Pandas e Matplotlib

Objetivo

Passar de "resolver exercícios" para construir um programa que outra pessoa consiga usar e entender.

O que deve aparecer no projeto

- **Pense antes de programar** (Cap. 9): problema bem definido, esboço, casos-limite
- **Estruturas de controle** (Cap. 3): decisões e repetições
- **Funções** (Cap. 4): programa dividido em funções coesas, módulos organizados
- **Estruturas de dados** (Cap. 5): listas, dicionários, tuplas; matrizes
- **Arquivos** (Cap. 6): leitura e/ou gravação (texto, CSV)
- **Tratamento de erros**: entradas inválidas e arquivos ausentes
- **NumPy / Pandas / Matplotlib** quando aplicável: cálculo, análise e gráficos

Requisito mínimo

Pelo menos **5 funções**, **1 arquivo de dados**, **tratamento de erros** e, quando houver dados numéricos, **2 gráficos**.

Sugestões de temas (gerais)

- Sistema de biblioteca simples
- Controle financeiro pessoal
- Jogo textual (forca, velha, labirinto)
- Simulador simples: fila, elevador, semáforo
- Agenda de contatos / de tarefas
- Análise de dados em arquivo CSV (esportes, clima, finanças)
- Calculadora científica / de unidades
- Quiz com banco de perguntas em arquivo
- Gerador de relatórios (texto + gráficos)

Sugestões por engenharia

Civil / Produção	Gestão de insumos e custos com dicionários e arquivos; controle de estoque; análise de produção e paradas (CSV + Pandas); orçamento de obra.
Elétrica / Mecânica	Simulador de circuitos (leis de Kirchhoff com NumPy); tabelas termodinâmicas; análise de ensaios (ajuste e gráfico); dimensionamento de componentes.
Ambiental / Cartográfica	Séries históricas de temperatura/chuva; conversor de coordenadas geográficas; cálculo de distâncias e áreas; mapas de calor em matriz.
Computação	Jogos com matrizes (velha, campo minado); simulações; recursão e ordenação aplicadas; mini-sistema com módulos.

Dados para analisar

- Arquivos de exemplo na pasta do curso: produção de peças (`dadosproducao.csv`) e produção/paradas mensais
- Dados abertos: IBGE, INMET (clima), Banco Mundial, Gapminder, Portal da Transparência
- **Seus próprios dados**: medições de laboratório, planilhas de estágio ou de iniciação científica (com autorização)

Boa pergunta de projeto

"Qual máquina/turno tem a maior taxa de defeitos?" · "Como a temperatura média evoluiu em 30 anos?" · "Qual a resistência de um componente a partir de ensaios?"
Uma pergunta clara ⇒ um projeto focado.

Hoje: formação dos grupos e esboço inicial

1. **Forme o grupo** (2 ou 3 pessoas) e escolha um nome
2. **Escolha o tema** (ou proponha o seu) e responda por escrito:
   - Qual problema vamos resolver? Para quem?
   - Quais as entradas e as saídas?
   - Que dados usaremos? De onde vêm?
3. **Esboço inicial** (meia página): 3 a 5 funcionalidades principais
4. Entregue o esboço ao professor ao final da aula

Cronograma do projeto

Aula	Etapa	Entrega
26	Lançamento	grupos e esboço
27–29	Matplotlib e Pandas (ferramentas)	exercícios de laboratório
30	Especificação	documento de especificação
31	Implementação 1	esqueleto funcionando
32	Implementação 2	arquivos e erros
33	Refatoração e testes	código limpo e testes
34–35	Apresentação	código + demonstração

Critérios de avaliação

Funcionalidade

O programa atende aos requisitos especificados e funciona sem "quebrar"?

Modularização

Código organizado em funções e módulos de forma lógica?

Organização do código

Bem estruturado, nomes claros, fácil de entender?

Clareza na explicação

Os alunos explicam as decisões de projeto e a implementação?

Todos os integrantes devem saber explicar *qualquer* parte do código.

Dicas para um bom começo

- **Comece pequeno**: uma versão simples que funciona vale mais que uma ambiciosa que não roda
- Crie a pasta do projeto, um arquivo principal e use controle de versões se conhecer Git
- Divida as tarefas, mas **integrem cedo e com frequência**
- Peça feedback ao professor em toda aula de implementação
- Código em equipe: combinem o estilo (nomes, indentação, docstrings)

Próxima aula (30)

Especificação detalhada do problema, das funcionalidades e da estrutura do programa.