

Departamento de Engenharia de Sistemas e Computação

Apresentação dos Projetos

Demonstração, explicação e critérios de avaliação

João Araujo

Algoritmos Computacionais I

Aula 34 e 35

Como serão as apresentações

- Cada grupo apresenta seu projeto para a turma: **10 a 12 minutos** + 3 minutos de perguntas
- Todos os integrantes falam
- Ordem definida por sorteio no início da aula 34
- **Entrega do código** (pasta compactada com instruções) antes da apresentação

Atenção

Teste a demonstração *no computador que será usado*. Tenha um plano B (capturas de tela ou vídeo curto) caso algo falhe.

Roteiro sugerido (10 minutos)

Tempo	Conteúdo	Quem
1 min	O problema: qual é, para quem, por que importa	integrante A
2 min	Visão geral: funcionalidades e estrutura do programa (módulos/funções)	integrante B
4 min	**Demonstração ao vivo** com dados reais	integrante A ou C
2 min	Decisões de projeto e desafios enfrentados (o que deu errado e como resolveram)	integrante C
1 min	Conclusões: limitações e próximos passos	todos

Critérios de avaliação

Funcionalidade

O programa atende aos requisitos especificados? Funciona com dados diferentes dos do exemplo? Trata entradas inválidas?

Modularização

O código está organizado em funções e módulos de forma lógica, com cada parte tendo uma responsabilidade?

Organização do código

O código é bem estruturado e fácil de entender? Nomes claros, docstrings, sem repetição?

Clareza na explicação

Os alunos explicam claramente as decisões de design e a implementação? Respondem às perguntas?

Pontos extras e pontos de atenção

Valorizamos

- Testes automáticos
- Gráficos bem feitos e interpretados
- Tratamento cuidadoso de erros
- Uso adequado de NumPy/Pandas/Matplotlib
- Instruções de uso claras (`LEIAME`)

Perdem pontos

- Código que só um integrante entende
- Programa que "quebra" em entradas simples
- Funções gigantes e repetição
- Apresentação que só lê os slides
- Estourar o tempo

Perguntas que o professor poderá fazer

- Por que escolheram *esta* estrutura de dados (lista, dicionário, DataFrame)?
- O que acontece se o arquivo não existir? E se o usuário digitar texto no lugar de número?
- Mostre a função mais importante e explique linha a linha
- Como vocês testaram? Qual foi o bug mais difícil?
- Onde há variáveis globais? Como removê-las?
- O que faria diferente se começasse hoje? O que falta para uso real?

Dicas para uma boa apresentação

- **Comece pelo problema**, não pelo código
- Poucos slides, poucas palavras; mostre o *programa rodando*
- Use entradas **realistas**; mostre também um erro sendo tratado
- Mostre trechos curtos de código (letra grande!) para ilustrar uma decisão
- Treine antes e cronometre
- Assuma as limitações: "ainda não faz X" é melhor que ser surpreendido

Material de apoio

Cap. 9 do livro e o site `https://www.araujo.eng.uerj.br/livro/`

Encerramento da disciplina

- O que aprendemos: pensar algoritmicamente, programar em Python, organizar dados, analisar e visualizar
- Isto é o **começo**: as próximas disciplinas e a vida profissional exigirão programar cada vez mais
- Continue praticando: exercícios adicionais do site, projetos pessoais, dados da sua área

Para continuar

Estruturas de dados e algoritmos · Cálculo numérico · Estatística com Python · Banco de dados · Aprendizado de máquina

Obrigado!