

Roteiro de Projeto: Algoritmos e Complexidade em Aplicação Web/Mobile

Objetivo

Desenvolver uma **aplicação web** (**frontend** + **backend**) que permita aplicar e demonstrar conceitos de **algoritmos**, **estruturas de dados** e **complexidade computacional**, além de apresentar boas práticas de versionamento e deployment em nuvem.

O trabalho busca:

- Implementar **funcionalidades** web **interativas**.
- Analisar **complexidade assintótica** de rotinas e algoritmos.
- Documentar **estruturas de dados** utilizadas.
- Explorar **técnicas de programação** avançadas quando possível.

Escopo da Aplicação Web

Os alunos devem **criar uma aplicação web** com pelo menos **uma das seguintes funcionalidades**:

1. Sistema de Cadastro e Busca:

- Usuários, produtos, serviços ou tarefas.
- Busca eficiente (ex.: **árvores de busca**, **tabelas hash**).

2. Gerenciamento de Ranking/Competição:

- Pontuação de usuários ou itens.
- **Algoritmos de ordenação** para exibir **ranking** (QuickSort, MergeSort).

3. Aplicação CRUD com Análise de Dados:

- Registro, edição e exclusão de dados.
- Cálculos estatísticos ou agregações (média, máximo, mínimo).

4. Simulação de Processos:

- Filas, pilhas ou grafos.
- Visualização de processos em tempo real.

A **escolha da funcionalidade** fica a critério do aluno, desde que permita demonstrar **algoritmos e estruturas de dados**.

Requisitos Técnicos

- **Linguagem e Frameworks:** Livre (**JavaScript/Node.js**, **Python/Django/Flask**, **Java/Spring Boot/IntelliJ IDEA**, **C#/Visual Studio**, VS Code, etc.).
- **Frontend:** HTML/CSS/JS, BootStrap, frameworks opcionais (React, Vue, Angular).
- **Backend:** APIs REST ou GraphQL(Linguagem de Consulta – Arquiteturas REST, Facebook).
- **Banco de Dados:** Relacional ou NoSQL, conforme o projeto.
- **Versionamento:** GitHub obrigatório, GitLab, BitBucket.
- **Deployment:** Aplicação **hospedada em nuvem** (Railway, Heroku, Vercel, AWS, Azure ou equivalente).

Análise de Algoritmos e Estruturas

Para cada rotina implementada, documentar:

1. Complexidade Assintótica

- Melhor caso, pior caso e caso médio (**Big O**, **Big Θ** , **Big Ω**).

2. Estruturas de Dados Utilizadas

- Arrays, listas encadeadas, filas, pilhas, árvores, grafos, hash tables, etc.
- Justificar a escolha para cada funcionalidade.

3. Equações de Recorrência

- Se houver recursão, apresentar a equação e resolver.

4. Algoritmos de Ordenação e Busca

- Indicar qual algoritmo foi usado, complexidade e motivos da escolha.

5. Recursividade

- Identificar funções recursivas e explicar sua lógica.

6. Técnicas Especiais

- Programação dinâmica
- Divisão e conquista
- Algoritmos gulosos
- Árvores ou grafos
- Problemas NP

Entrega

1. GitHub:

- Código-fonte completo e organizado.
- **README** com instruções de execução, análise de algoritmos e estruturas.

2. Deployment:

- Link funcional da aplicação na nuvem.
- Instruções para acesso e teste.

3. Documento de Análise:

- **PDF** ou **Markdown(linguagem de marcação)**, detalhando:
 - Complexidade assintótica das rotinas.
 - Estruturas de dados usadas.
 - Algoritmos implementados e técnicas aplicadas.

Avaliação – 06 pts

Critério	Peso
Funcionalidade da aplicação web	30%
Análise de complexidade e documentação	30%
Uso de algoritmos e técnicas avançadas	20%
Versionamento e deployment	10%
Criatividade e clareza na apresentação	10%

Considerações Finais

1. **Equipe:** **Individual** ou máximo 02(dois) componentes.
2. **Prazo de Entrega:** O prazo de entrega é uma semana antes da Data AV Institucional.
3. **Entrega do Trabalho:** Entregar tanto a **aplicação web/mobile** quanto a documentação do projeto.
4. **Não será aceita entrega com atraso**, a nota **PROJETO WEB** será **DESCONTADO 01 PONTO** em caso de atraso.
5. **Criação de Repositório no GitHub:** O repositório deve ser público e o link postado na plataforma SAVA do aluno.
6. **Arquivo README:** Incluir no repositório do **GitHub** um arquivo **README**, contendo informações sobre os componentes da equipe e a descrição do projeto.
7. **Seminário:** Uma semana antes da Data AV Institucional, será realizado o seminário para apresentar a **aplicação web/mobile** desenvolvida.
8. **Tempo Apresentação:** 8min. a 10min – todos os componentes da equipe devem apresentar;
9. **Postar no SAVA:** **01 semana antes do Seminário data da AV**, o link do **GitHub**.

Nota: No **GitHub** deve conter os nomes dos componentes, a parte escrita padrão ABNT, o código do projeto e o **PPT da apresentação do seminário**.