



INFORMAÇÕES SOBRE A AVALIAÇÃO

LAB01	Laboratório 01 - 15 pontos
-------	----------------------------

INFORMAÇÕES DOCENTE						
CURSO: ENGENHARIA DE SOFTWARE	DISCIPLINA: LABORATÓRIO DE DESENVOLVIMENTO DE SOFTWARE	TURNO	MANHÃ	TARDE	NOITE	PERÍODO/SALA: 4º
			x		x	
PROFESSOR (A): João Paulo Carneiro Aramuni						

Portfólio Profissional

Neste laboratório, vamos projetar e desenvolver um website do seu portfólio profissional. Leia atentamente a descrição fornecida pelo Product Owner e elabore o projeto detalhado do sistema, incluindo front-end, back-end e hospedagem na nuvem.

Descrição do Sistema:

O objetivo é desenvolver um website de portfólio profissional para você, que deseja apresentar sua trajetória, habilidades, projetos e formas de contato de maneira moderna e acessível. O sistema deverá conter as seguintes seções acessadas por um menu de navegação:

1. Sobre Mim

- Breve apresentação em português e inglês, destacando sua formação, área de atuação, interesses e objetivos profissionais.

2. Projetos

- Seção com linha do tempo de projetos, do mais antigo ao mais recente.
- Cada projeto deverá conter:
 - Nome e descrição;
 - Tecnologias utilizadas;
 - Link para o repositório no GitHub;
 - Imagem ou GIF com o projeto em funcionamento (telas, login, interações).

3. Experiências

- Espaço para relatar experiências profissionais, estágios, freelas, participações em projetos open source ou eventos técnicos.
- Para cada item: nome da empresa/instituição, cargo ou atividade, período e breve descrição.



PUC Minas

4. Contato

- Página com: Ícones clicáveis para e-mail, WhatsApp, LinkedIn, entre outros;
- Formulário com campos de nome, e-mail e mensagem, com funcionalidade de envio por e-mail.

O sistema deve ter um design responsivo, interface amigável e identidade visual coerente com seu perfil profissional. Ao final, o site deverá estar hospedado gratuitamente em nuvem (ex: Render, Vercel, Heroku, Fly.io), e o repositório no GitHub deverá conter um README completo, incluindo:

- Listagem das tecnologias utilizadas;
- Relação das dependências e bibliotecas/frameworks usados;
- Estrutura de diretórios do projeto;
- Instruções de instalação e execução do sistema localmente;
- Link de acesso para o site publicado na nuvem.

Apresentação Final:

Ao final da Sprint 03, você deverá apresentar o sistema final, hospedado e funcional, explicando as escolhas de design, arquitetura e as funcionalidades implementadas.

A avaliação considerará:

- Qualidade do site (visual, responsividade, conteúdo e funcionalidades);
- Organização do repositório e clareza do README;
- Alinhamento com os protótipos e funcionalidades definidas;
- Funcionamento da hospedagem na nuvem.

Processo de Desenvolvimento:

Lab01S01: Planejamento e prototipação do site: (4 pontos)

- Criação do repositório GitHub com README inicial;
- Wireframes das páginas no figma (média fidelidade);
- Protótipo inicial do front-end (HTML/CSS/JS ou frameworks modernos como React/Vue/Mantine/Material-UI);
- Implementação da navegação (estrutura de páginas e links entre seções) e do layout principal (organização visual base do site, com cabeçalho, rodapé e área de conteúdo).

Entrega: README com imagens dos protótipos, descrição do projeto, tecnologias previstas e estrutura inicial do site.



PUC Minas

Lab01S02: Implementação das funcionalidades principais: (4 pontos)

- Página "Sobre Mim" com versões em português e inglês;
- Página "Projetos" com timeline dinâmica;
- Página "Experiências" com dados organizados;
- Página "Contato" com ícones e formulário funcional (ex: envio de e-mail);
- Validações básicas e responsividade.

Entrega: Versão funcional local ou com preview em ambiente temporário (ex: Vercel Preview).

Lab01S03: Hospedagem e finalização do sistema: (7 pontos)

- Deploy completo em Render, Vercel, Heroku, Fly.io ou similar;
- Ajustes visuais e de usabilidade;
- Inserção de imagens/GIFs dos projetos em execução;
- README final com: Tecnologias utilizadas; Link para o site publicado; Instruções de uso e desenvolvimento.

Para este projeto, utilize o template de **README.md** disponível aqui: https://github.com/joaopauloaramuni/laboratorio-de-desenvolvimento-de-software/blob/main/TEMPLATES/template_README.md

Prazo final (Acesse o cronograma): <https://github.com/joaopauloaramuni/laboratorio-de-desenvolvimento-de-software/tree/main/CRONOGRAMA>

Valor total: 15 pontos | Desconto de 1.0 ponto por dia de atraso

Atenção: semanalmente, todos os grupos deverão apresentar o andamento das entregas durante a aula. A não participação do grupo implicará na perda automática de 50% dos pontos da sprint.

Observação: Cada integrante do grupo deverá apresentar seu próprio portfólio, com conteúdo personalizado (Sobre Mim, Projetos, Experiências e Contato). No entanto, a estrutura do front-end (como layout, navegação e componentes reutilizáveis) poderá ser compartilhada entre os membros do grupo, desde que o conteúdo e o repositório final de cada aluno sejam individuais, em seu próprio GitHub.

Links úteis:

Protótipos e design:

- Figma (modelos prontos) – Criação de protótipos de interface e wireframes <https://www.figma.com/pt-br/modelos/>



PUC Minas

Bibliotecas de Componentes UI:

- Material UI (MUI) – Componentes React com design do Google
<https://mui.com/material-ui/>
- Mantine – UI moderna para React
<https://mantine.dev/>
- shadcn/ui – Componentes estilizados com Tailwind CSS
<https://ui.shadcn.com/>

Frameworks e bibliotecas JavaScript:

- React – Biblioteca JavaScript para construção de interfaces
<https://react.dev/>
- Vue.js – Framework progressivo para construção de interfaces
<https://vuejs.org/>
- Next.js – Framework React fullstack com suporte a SSR e API routes
<https://nextjs.org/>
- Three.js – Biblioteca JavaScript para gráficos 3D na web
<https://threejs.org/>

Funcionalidades / Integrações úteis:

- Tailwind CSS – Framework CSS utilitário para estilização rápida
<https://tailwindcss.com/>
- Framer Motion – Animações para interfaces React
<https://www.framer.com/motion/>
- React Hook Form – Biblioteca para manipulação de formulários em React
<https://react-hook-form.com/>
- EmailJS – Envio de e-mail diretamente do front-end
<https://www.emailjs.com/>

Hospedagens gratuitas:

- Vercel – Hospedagem para front-end, ideal para projetos em Next.js
<https://vercel.com/>
- Render – Hospedagem fullstack para aplicações web e APIs
<https://render.com/>
- GitHub Pages – Hospedagem gratuita para sites estáticos
<https://pages.github.com/>
Exemplo: <https://pythonmg.github.io/>

Hospedagens pagas:

- Fly.io – Hospedagem distribuída de apps back-end ou fullstack
<https://fly.io/>



PUC Minas

- Heroku – Hospedagem de back-end com suporte a diversas linguagens
<https://www.heroku.com/>

Repositório e versionamento:

- GitHub – Hospedagem de código e controle de versão
<https://github.com/>

Exemplos de portfólio:

- Flávio Júnior
 - <https://flaviojunior-portfolio.vercel.app/>
- Gabriel Victor
 - <https://gabrielvictor.web.app/>
- Pedro Braga
 - <https://www.pedrobragadev.com/>
- Luca Azalim
 - <https://azal.im/>