

Projeto Projeto Interdisciplinar – Desenvolvimento Web Full Stack	Curso Análise Desenvolvimento de Sistemas	Turma 2º Semestre, 2025
Tema Desenvolvimento de um Web Site	Título Desenvolvimento de um site institucional para o Instituto Alma, organização voltada a causas sociais e desenvolvimento humano.	

Site Institucional – Instituto Alma

Projeto Interdisciplinar – 2ADS – 2º SEMESTRE 2025

1. Apresentação do Projeto

Este projeto tem como objetivo o desenvolvimento de um site institucional criativo, funcional e responsivo para o Instituto Alma (@almainstituto_oficial), organização voltada a causas sociais e desenvolvimento humano. O site irá concentrar informações essenciais sobre o Instituto, suas atividades, formas de colaboração e prestação de contas com a sociedade.

O projeto será conduzido por alunos do 2º semestre de Análise e Desenvolvimento de Sistemas, permitindo aplicar conhecimentos de front-end e back-end adquiridos em disciplinas como Desenvolvimento Web Full Stack, Programação Orientada a Objetos, Algoritmos e Estrutura de Dados, Design de Interface Digital, e Modelagem de Software e Arquitetura de Sistemas.

2. Objetivos

Objetivo Geral:

Criar um site institucional funcional e de fácil navegação, com conteúdo dinâmico e integração a redes sociais e meios de pagamento online.

Objetivos Específicos:

- Apresentar o Instituto Alma, suas metas e valores;
- Disponibilizar um portal para divulgação de atividades e eventos;
- Criar um canal para doações com pagamento online;
- Garantir a transparência por meio da publicação de relatórios e documentos;
- Estabelecer uma ouvidoria com formulário de contato direto;
- Integrar redes sociais para divulgação automática de eventos.
- Administrar conteúdo dinâmico a partir de uma URL específica mediante a *login* e senha.

3. Estrutura do Site



3.1. Menu Principal

- **Sobre Nós:** Página institucional com texto sobre o Instituto, sua história, missão, visão e valores.
- **Nossas Atividades:** Página dinâmica com publicações de ações, projetos e atividades desenvolvidas pelo Instituto, com título, descrição e imagens.
- **Portal do Doador:** Área para realização de doações com meios de pagamento online (ex: Pix, cartão de crédito).
- **Transparência:** Página com upload e exibição de relatórios em PDF.
- **Ouvidoria:** Formulário de contato para envio de sugestões, dúvidas e reclamações.
- **Eventos:** Página dinâmica com lista de eventos realizados e futuros com integração com redes sociais.
- **Administração do Site:** Página com URL específica com acesso a login e senha para atualizar o conteúdo das páginas Nossas Atividades, Transparência e Eventos. Também acompanhar as doações feitas no Portal do Doador e as mensagens da ouvidoria

4. Tecnologias Sugeridas

- **Front-End:** HTML5, CSS3, JavaScript, Bootstrap
- **Back-End:** C# ou Node.js (express)
- **Banco de Dados:** MySQL ou SQLite
- **Ferramentas/Serviços Externos:** API de pagamento (PagSeguro, Mercado Pago ou Stripe), API do Instagram/Facebook, Firebase ou Nodemailer

5. Funcionalidades Esperadas

Página	Funcionalidade
Sobre Nós	Texto institucional estático
Nossas Atividades	CRUD de postagens com título, texto e imagem
Portal do Doador	Integração com plataforma de pagamentos
Transparéncia	Upload e visualização de arquivos PDF
Ouvidoria	Formulário de contato com envio por e-mail
Eventos	Cadastro e exibição de eventos + integração com redes sociais

6. Requisitos Não Funcionais

- Site responsivo para dispositivos móveis
- Interface amigável (UX/UI)
- Sistema de login para área administrativa
- Proteção de dados sensíveis (HTTPS, validações)

7. Etapas do Projeto

1. Planejamento e levantamento de requisitos
2. Criação do protótipo da interface
3. Modelagem do banco de dados
4. Implementação front-end
5. Desenvolvimento do back-end
6. Integração com APIs externas
7. Testes e validações
8. Implantação do site
9. Documentação e apresentação final

8. Possíveis Extensões da Solução

- Área de voluntariado
- Blog com atualizações periódicas
- Histórico de contribuições para doadores
- Newsletter automatizada

Obs: Cada equipe pode propor e implementar suas próprias extensões.

Entregáveis

Data das entregas	22/set.	10/nov.	17/11 - 25/11
Disciplinas	Entrega 1 (3,0 Pontos)	Entrega 2 (4,0 Pontos)	Apresentação (3,0 Pontos)
Algoritmos e Estrutura de dados	<p>Formato de Entrega: Implementação de Algoritmo de Ordenação na Lista Recebida</p> <p>Objetivo: Os alunos devem DEFINIR UMA ESTRUTURA DE DADOS QUE REPRESENTE para as páginas dinâmicas NOSSAS ATIVIDADES e TRANSPARÊNCIA. Criar uma lista com estas estrutura de dados</p>	<p>Formato de Entrega: Implementação da solução em um arquivo ZIPADO</p> <p>Objetivo: Fazer leitura do Banco de Dados e gerar um lista de ATIVIDADES e de DOCUMENTOS de TRANSPARÊNCIA</p>	<p>Formato de entrega: Banner e Pitch de 4 minutos.</p> <p>Critérios: Criatividade, Impacto Social, Tempo de apresentação e Embasamento.</p>
Full Stack	<u>ANEXO1</u>	<u>ANEXO 2</u>	
Design de Interface digital	<p>Formato de Entrega:</p> <ul style="list-style-type: none"> - Criação do layout e prototipação da Home no Figma. - Definição da paleta de cores. - Elaboração do Guia de Estilo do projeto. <p>Descrição: Os alunos devem desenvolver o layout e o protótipo da Home no Figma, garantindo que a prototipação represente o design final com todos os elementos de interface bem definidos (botões, navegação, campos de entrada, etc.).</p> <p>Além disso, é necessário definir uma paleta de cores coerente e acessível, proporcionando uma experiência</p>	<p>Formato de Entrega:</p> <ul style="list-style-type: none"> - Mapa de navegação; - Prototipação navegável das páginas do projeto no Figma. - Elaboração do Design System do projeto. <p>Descrição: Os alunos devem desenvolver um protótipo interativo no Figma, representando visualmente o layout, o design e a interatividade do site ou sistema. O protótipo permitirá validar o fluxo de navegação e a experiência do usuário com os componentes da interface.</p> <p>Objetivos:</p>	

	<p>visual agradável ao usuário. O Guia de Estilo deve estabelecer padrões visuais, tipográficos e de layout, garantindo consistência no desenvolvimento do sistema.</p> <p>Objetivos:</p> <ul style="list-style-type: none">- Definir a Identidade Visual: Criar um padrão de estilo para os componentes do sistema, incluindo botões, menus, tipografia e ícones.- Padronizar Elementos de Interface: Estabelecer diretrizes para espaçamentos, alinhamentos e interações, garantindo boas práticas de UI/UX.- Assegurar Coerência Visual: Harmonizar os elementos do design para proporcionar uma experiência intuitiva e agradável.- Definir a Paleta de Cores: Escolher e justificar as cores do projeto, considerando o público-alvo e a acessibilidade, assegurando combinações equilibradas para fundo, textos, botões e demais elementos.	<ul style="list-style-type: none">- Criar Protótipos Navegáveis: Representar as páginas principais do projeto de forma interativa.- Definir Fluxos de Navegação: Assegurar que todas as páginas estejam conectadas, garantindo uma navegação clara e fluida.- Refinar Design e Funcionalidade: Aplicar padrões de design e melhorar a usabilidade para tornar a interação intuitiva.
Modelagem de Software ESSA DISCIPLINA FOI PARA O 1 SEM.		
POO	<p>Instruções para Entrega: Estruturação básica do backend (com classes e banco de dados) e com o upload para servidor em nuvem.</p> <p>Objetivo: Os alunos devem implementar e entregar a classe principal do sistema, que será responsável por integrar as funcionalidades já implementadas e representar a estrutura central do sistema. Esta classe deve orquestrar a interação entre diferentes partes do sistema e garantir o fluxo adequado das operações.</p>	<p>Instruções para Entrega: Sistema completamente funcional rodando na nuvem.</p> <p>Objetivo: Os alunos devem realizar a subida do backend do projeto para o Azure, garantindo que o sistema esteja disponível e acessível via a plataforma de nuvem Azure. Isso envolve a configuração de um serviço de backend (por exemplo, Web App, Azure Functions, ou App Service) e a</p>

		correta implementação de conexões com o banco de dados, se necessário.	
Banco de Dados	Apresentar em documento a criação de índices para melhorar a performance das consultas e justificar tal criação.	Apresentar em documento a criação de pelo menos uma view englobando consultas.	

ANEXO 1

1ª Entrega do Projeto Interdisciplinar (PI)

Nome da Entrega: **Estrutura Inicial do Projeto Web Full Stack**

Objetivo da Entrega:

Entregar a primeira versão funcional do projeto web, contendo:

- Frontend estático e reativo com HTML, CSS, JavaScript e React
- Backend funcional com Node.js, Express e MySQL
- Integração parcial entre frontend e backend
- Estrutura organizada e modularizada, com navegação e interatividade

Conteúdos contemplados (Aulas 1 a 8):

- Configuração do ambiente (VS Code, extensões)
- HTML5 semântico e responsivo
- CSS com Flexbox e Grid
- JavaScript com manipulação do DOM
- Introdução ao React (JSX, componentes, props, estado)
- Criação de formulários controlados
- Criação de API com Express
- Operações de CRUD básicas com MySQL

Requisitos Técnicos

Frontend:

1. Projeto React funcional com ao menos 3 componentes (ex: Header, Home, Card).
2. Layout responsivo com CSS Flexbox ou Grid.
3. Página inicial navegável e formulário funcional (ex: cadastro de item).
4. Manipulação de eventos e uso de useState.
5. Estilização consistente e sem redundâncias.

1. Estrutura Node.js com Express, organizada em rotas e controladores.
2. Banco de dados MySQL com ao menos uma tabela (ex: notícias, produtos, usuários).
3. Implementação das operações **Create, Read, Update e Delete**.
4. Uso de Postman para testes locais das rotas.

Integração (mínima e parcial):

1. O frontend deve **listar ou enviar dados** para o backend (ex: cadastrar e exibir notícias).
2. Uso de `fetch()` ou axios no React para conectar à API criada.

Ferramentas Recomendadas:

- Visual Studio Code
- React + Vite
- Node.js + Express
- MySQL + Workbench
- Postman
- Netlify
- Firebase, Railway, Vercel

Deploy obrigatório

- **Frontend:** Netlify ou Vercel
- **Backend:** Railway, Render, ou similar
- Alunos devem apresentar os **dois links públicos funcionando**
- Repositório Git com código-fonte deve conter **README estruturado**, incluindo:
 - Descrição do projeto
 - Rotas da API
 - Como rodar o projeto localmente
 - Vídeo demonstrativo

- Cada grupo deve apresentar o projeto funcionando, acessível por links públicos.
- O código deve estar organizado, legível e modularizado.
- O professor poderá solicitar acesso ao código-fonte e à API via navegador ou Postman.
- O professor poderá fazer perguntas técnicas para verificar domínio do conteúdo.
- A versão final do projeto será complementada na segunda entrega (PI2), com recursos como autenticação, upload de imagens e deploy.

ANEXO 2

2ª Entrega do Projeto Interdisciplinar (PI)

Nome da Entrega: Projeto Web Full Stack Finalizado com Funcionalidades Avançadas e Deploy

Objetivo da Entrega

Entregar a versão final do projeto integrador Full Stack com todas as funcionalidades implementadas, integrando **frontend com React** e **backend com Node.js + MySQL**, com **upload de imagens, autenticação com JWT, rotas protegidas, estilização aprimorada, e deploy completo** das duas camadas da aplicação.

Conteúdos contemplados (Aulas 10 a 16):

- Upload de imagens com Multer
- Autenticação com Bcrypt e JWT
- Integração real Frontend ↔ Backend com fetch/axios
- React Router DOM e navegação entre rotas
- Estilização com CSS avançado e acessibilidade básica
- Testes de funcionalidades e validações
- Deploy completo do sistema

Requisitos Técnicos

Frontend (React):

- SPA completa com navegação via React Router DOM
- Componentes reutilizáveis com `props` e `useState`
- Consumo de API com `fetch` ou `axios`, incluindo rotas autenticadas
- Armazenamento e uso do token JWT
- Layout responsivo com Flexbox/Grid e transições visuais
- Mensagens de erro e loading.
- Formulários controlados com validações básicas
- Deploy no Netlify ou Vercel

Backend (Node.js + Express + MySQL):

- API REST completa com rotas protegidas por middleware JWT
- Upload de imagens com Multer (armazenamento local ou remoto)
- Criptografia de senhas com Bcrypt
- Conexão com banco de dados relacional
- Organização modular: rotas, controladores, middlewares, models
- Respostas estruturadas com status HTTP apropriados
- Variáveis de ambiente separadas
- Deploy no Render, Railway ou outro serviço compatível

Integração obrigatória:

- Frontend deve consumir a API hospedada
- Funcionalidades de login e acesso autenticado
- CRUD completo e funcional
- Envio de imagens via formulário

Deploy obrigatório

- **Frontend:** Netlify ou Vercel
- **Backend:** Railway, Render, ou similar
- Alunos devem apresentar os **dois links públicos funcionando**
- Repositório Git com código-fonte deve conter **README estruturado**, incluindo:
 - Descrição do projeto
 - Rotas da API
 - Como rodar o projeto localmente
 - Vídeo demonstrativo

Dicas e Recomendações

- Teste o projeto completo antes da apresentação.
- Deixe o código limpo, sem console.log ou arquivos não utilizados.
- O professor poderá testar o sistema ao vivo e solicitar explicações sobre trechos do código.