

Guia para Desenvolvimento do Projeto de Programação Web

1. Escolha do Tema

- A equipe deve escolher um tema para a plataforma. Algumas sugestões:
 - E-commerce (loja virtual)
 - Rede social
 - Blog
 - Plataforma de cursos online
 - Sistema de gerenciamento de tarefas (To-Do List)
 - Plataforma de reservas (hotel, restaurante, etc.)
 - O tema escolhido deve permitir a criação das páginas e rotas solicitadas.
-

2. Definição das Rotas da API

- A equipe deve planejar as rotas da API com base nos métodos HTTP:
 - **GET**: Para buscar informações (ex: listar produtos, exibir detalhes de um item).
 - **POST**: Para criar novos recursos (ex: cadastrar usuário, adicionar item).
 - **PUT**: Para atualizar recursos existentes (ex: editar perfil do usuário, atualizar item).
 - **DELETE**: Para remover recursos (ex: excluir item, remover usuário).
 - Exemplo de rotas para um tema de e-commerce:
 - Listar todos os produtos.
 - Exibir detalhes de um produto específico.
 - Adicionar um novo produto.
 - Atualizar um produto existente.
 - Remover um produto.
-

3. Estrutura do Projeto

- Organize o projeto em pastas para facilitar o desenvolvimento:
 - **public/**: Arquivos estáticos (CSS, JS, imagens).
 - **views/**: Páginas HTML (Login, Home, Admin, etc.).
 - **dados/**: Arquivos de dados armazenados em json (usuarios.json, itens.json, etc.).
 - **app.js**: Arquivo principal do servidor Express.
 - **package.json**: Dependências do projeto.
-

4. Desenvolvimento das Páginas

- **Landing Page:** Página inicial do site, com uma breve descrição do tema e links para outras páginas.
 - **Página de Login:** Formulário para autenticação de usuários.
 - **Página do Usuário/Admin:** Após o login, o usuário deve ser redirecionado para uma página personalizada (ex: perfil do usuário ou painel de administração).
 - **Página de Detalhes:** Exibe informações detalhadas sobre um item específico (ex: detalhes de um produto, postagem, etc.).
-

5. Implementação do Servidor Express

- Instale as dependências necessárias, como o Express
 - Configure o servidor para servir arquivos estáticos e renderizar páginas.
 - Defina rotas básicas, como a rota para a landing page.
-

6. Implementação das Rotas da API

- Crie um arquivo de rotas para a API.
 - Implemente pelo menos uma rota para cada método HTTP (GET, POST, PUT, DELETE).
 - Exemplos de rotas:
 - Listar recursos (GET).
 - Adicionar um novo recurso (POST).
 - Atualizar um recurso existente (PUT).
 - Remover um recurso (DELETE).
-

7. Integração com GitHub

- Crie um repositório no GitHub e compartilhe com os membros da equipe e o professor.
 - Use branches para organizar o trabalho (ex: feature/login, feature/api).
 - Faça commits regulares com mensagens descritivas.
-

8. Testes e Apresentação

- Teste todas as rotas e páginas para garantir que tudo funcione corretamente.
-

Exemplo de Tema: E-commerce

- **Landing Page:** Página inicial com destaques de produtos.
 - **Página de Login:** Formulário para login de usuários.
 - **Página do Usuário/Admin:** Perfil do usuário ou painel de administração para gerenciar produtos.
 - **Página de Detalhes:** Exibe informações detalhadas sobre um produto.
-

Dicas Extras

- Use ferramentas como **Postman** ou **Insomnia** ou uma extensão na IDE (REST Client) para testar as rotas da API,.
- Se o tempo permitir, adicione um banco de dados (ex: SQLite, MongoDB) para persistência de dados.
- Documente o projeto no README do GitHub, explicando como executar e testar a aplicação.