

Conecta Empresas

Resumo do Projeto: Desenvolvimento de Sistema de Cadastro e Login com MongoDB, Express e Node.js

Introdução: O objetivo deste projeto foi criar um sistema básico de cadastro e login de usuários utilizando as tecnologias Node.js, Express, MongoDB e o framework de front-end HTML/CSS. O sistema foi implementado para permitir que usuários se registrem, façam login e sejam redirecionados para uma página inicial (index.html) após a autenticação bem-sucedida.

Fase 1: Planejamento e Levantamento de Requisitos:

Reunião Inicial com Stakeholders(público alvo):

2. Levantamento de Requisitos:

Tecnologias Utilizadas:

- 1. Node.js: Plataforma de desenvolvimento utilizada para a criação do servidor. Node.js permite a execução de código JavaScript no lado do servidor, proporcionando uma base eficiente e escalável para aplicações web.**
- 2. Express.js: Framework web minimalista para Node.js. O Express simplifica a criação de APIs e o gerenciamento de rotas, facilitando o desenvolvimento de aplicações web com Node.js.**
- 3. MongoDB: Banco de dados NoSQL utilizado para armazenar informações dos usuários, como nome de usuário e senha. O MongoDB é uma escolha popular por sua flexibilidade e escalabilidade em projetos web.**
- 4. Mongoose: Ferramenta que facilita a interação entre o Node.js e o MongoDB, permitindo a modelagem de dados e**

a execução de operações CRUD (Create, Read, Update, Delete) de forma simplificada.

- 5. bcrypt: Biblioteca utilizada para hashing de senhas. O bcrypt é uma ferramenta segura e confiável para criptografar senhas antes de armazená-las no banco de dados, garantindo que os dados sensíveis dos usuários sejam protegidos.**
- 6. HTML/CSS: Tecnologias padrão de marcação e estilização usadas para criar as páginas de cadastro (cadastro.html) e login (login.html). O CSS foi utilizado para estilizar os formulários, tornando a interface amigável e fácil de usar.**
- 7. JavaScript: Linguagem de programação utilizada para adicionar funcionalidades dinâmicas às páginas HTML. O JavaScript foi usado no front-end para validar o formulário, enviar dados para o servidor e lidar com as respostas da API.**

Funcionalidades Implementadas:

1. Cadastro de Usuário:

- **Página HTML (cadastro.html) com um formulário para entrada de dados de usuário e senha.**
- **Validação de campos no front-end, assegurando que as senhas coincidam.**
- **Envio dos dados para o servidor através de uma requisição POST usando fetch.**
- **No servidor, os dados são recebidos e a senha é criptografada usando bcrypt antes de ser armazenada no MongoDB.**
- **Após o cadastro bem-sucedido, o usuário é redirecionado para a página de login.**

2. Login de Usuário:

- **Página HTML (login.html) com um formulário para entrada de nome de usuário e senha.**

- **Envio dos dados de login para o servidor através de uma requisição POST.**
- **No servidor, a senha fornecida é comparada com a senha criptografada armazenada no banco de dados usando bcrypt.**
- **Em caso de sucesso, o usuário é redirecionado para a página index.html.**

3. Roteamento e Servidor:

- **Configuração do servidor Express para lidar com rotas de registro (/register) e login (/login).**
- **Uso de res.redirect para redirecionar o usuário para páginas específicas após o login ou registro bem-sucedido.**

4. Configuração de Ambiente (.env):

- **Utilização de um arquivo .env para armazenar variáveis de ambiente, como a URI de conexão ao MongoDB, mantendo essas informações seguras e fora do código-fonte.**

Conclusão: Este projeto demonstrou a integração de várias tecnologias para criar um sistema web completo com funcionalidades de autenticação de usuário. Ao utilizar Node.js e Express para o back-end, junto com MongoDB para o banco de dados e HTML/CSS/JavaScript para o front-end, foi possível desenvolver um sistema eficiente, seguro e escalável, que pode servir de base para aplicações mais complexas no futuro.

Cronograma e Planejamento de Recursos:

Sprint 1: Configuração Inicial e Back-end

Duração: 1 semana

Pessoa 1: Configuração do Servidor e Integração com MongoDB

- **Configurar o servidor Node.js com Express.**

- **Configurar a conexão com o MongoDB utilizando Mongoose.**
- **Criar a estrutura básica do servidor com rotas para cadastro e login.**
- **Milestone: Servidor configurado e rotas básicas criadas.**

Pessoa 2: Implementação das Funções de Cadastro e Login

- **Implementar a função de registro de usuário, incluindo o hash de senha com bcrypt.**
- **Implementar a função de login de usuário, incluindo verificação de senha com bcrypt.**
- **Milestone: Funcionalidade de cadastro e login implementada no servidor.**

Pessoa 3: Criação do Banco de Dados

- **Configurar e criar o banco de dados no MongoDB Atlas.**
- **Criar a coleção de usuários no banco de dados com os campos necessários (username e password).**
- **Milestone: Banco de dados configurado e coleção de usuários criada.**

Sprint 2: Desenvolvimento do Front-end

Duração: 1 semana

Pessoa 1: Desenvolvimento das Páginas HTML e Estrutura de Estilos

- **Criar a página de cadastro (cadastro.html) com estrutura HTML.**
- **Criar a página de login (login.html) com estrutura HTML.**
- **Desenvolver o arquivo CSS (styles.css) para estilização das páginas.**
- **Milestone: Páginas de cadastro e login criadas e estilizadas.**

Pessoa 2: Implementação da Lógica de Front-end com JavaScript

- **Implementar a validação de formulários no front-end (comparação de senhas no cadastro).**
- **Implementar a lógica de envio de dados (fetch) para o servidor nas páginas de cadastro e login.**
- **Milestone: Funcionalidades de validação e envio de dados implementadas.**

Pessoa 3: Integração Front-end/Back-end

- **Testar a integração entre as páginas front-end e as rotas do servidor.**
- **Garantir que os dados inseridos nos formulários são corretamente enviados e recebidos pelo servidor.**
- **Milestone: Integração completa entre front-end e back-end.**

Sprint 3: Finalização e Testes

Duração: 2 dias

Pessoa 1: Redirecionamento e Feedback ao Usuário

- **Implementar redirecionamento do usuário para a página de login após o cadastro bem-sucedido.**
- **Implementar redirecionamento para a página index.html após login bem-sucedido.**
- **Milestone: Redirecionamentos implementados e testados.**

Pessoa 2: Testes de Funcionalidade e Segurança

- **Realizar testes de funcionalidade para verificar o funcionamento correto de cadastro e login.**
- **Testar a segurança do sistema, incluindo o hash de senha e proteção contra ataques comuns.**
- **Milestone: Funcionalidades testadas e sistema validado.**

Pessoa 3: Documentação e Refinamento Final

- **Documentar o processo de desenvolvimento e as principais decisões técnicas.**
- **Revisar o código para possíveis melhorias e refinamentos finais.**

- **Milestone: Documentação concluída e refinamentos aplicados.**