



## Curso Superior de Desenvolvimento de Software Multiplataforma

Alexsander Ramos Ferreira, RA 3011392313001

Ali Haidar Pucci Sidani, RA 3011392313004

Cláudio Vinicius de Almeida, RA 3011392313009

João Cunha Fischer, RA 3011392313014

Joel de Farias Alves Neto, RA 3011392313033

### Projeto Interdisciplinar

Gestão ágil de Projetos de Software

Interação Humano Computador

Desenvolvimento Web III

Técnicas de Programação II

Banco de Dados Não Relacional:

### CAPI.TECH

Orientadores: Prof. Jones Artur Gonçalves

Profª Maria Janaína da Silva Ferreira

Prof. Ricardo Roberto Leme

Prof. Rodrigo de Paula Diver

Votorantim

Junho, 2024

## RESUMO

O projeto em foco visa desenvolver um site educacional gratuito com ênfase em tecnologia, utilizando ferramentas como Visual Studio Code, HTML, CSS, JavaScript, TypeScript, Node.js, Express.js, React Icons, Flaticon, Figma, Insomnia, Postman. A plataforma abrangente é estruturada em três componentes cruciais. Inicialmente, apresenta brevemente a FATEC, proporcionando aos usuários um entendimento inicial sobre a instituição. Em seguida, realiza uma análise detalhada do curso e do contexto mais amplo da área de tecnologia.

A parte central do site é dedicada à disponibilização de conteúdo educacional, oferecendo aos usuários acesso a uma variedade de materiais, como vídeos aulas, links úteis e sugestões de cursos adicionais. Essa seção não apenas fornece informações, mas também cria um ambiente de aprendizado envolvente e acessível.

Uma característica distintiva do projeto é a facilidade de navegação e a capacidade dos usuários melhorarem seu desempenho nos estudos. Essa abordagem visa tornar o aprendizado mais acessível e personalizado, incentivando os usuários a explorarem e se aprimorarem na área de tecnologia.

Em resumo, a iniciativa busca oferecer uma plataforma aberta e gratuita para aprendizado em tecnologia, utilizando ferramentas como Visual Studio Code, HTML, CSS, JavaScript, TypeScript, Node.js, Express.js, React Icons, Flaticon, Figma, Insomnia, Postman, com o objetivo não apenas de transmitir conhecimento, mas também de promover a inclusão digital e a formação de profissionais qualificados na área.

## SUMÁRIO

1.Descrição do projeto .....	6
1.1. Proposta do Software (Objetivo) .....	6
1.2. Justificativa.....	6
1.3. Mapa Mental.....	7
1.4. Logomarca .....	8
2. Requisitos do projeto .....	9
2.1. Levantamento de Requisitos .....	9
2.2. Backlogs & User Stories .....	10
2.3. Requisitos funcionais .....	11
2.4. Requisitos não funcionais.....	11
2.5. Diagrama de Caso de Uso .....	12
2.6. Arquitetura da Aplicação.....	13
2.7. Tecnologias Utilizadas .....	13
2.8. Diagrama de Classe .....	15
2.9. Implementação Das Classes .....	16
2.10. Interfaces com o usuário .....	18
3. ESTRATÉGIA DE TESTES .....	20
4. IMPLANTAÇÃO.....	21
5. REFERÊNCIAS.....	21

**INDICE DE IMAGENS**

Figura 1 - Mapa do site .....	7
Figura 2 - Logo e Nome do Projeto - Aprendendo com quem aprende.....	8
Figura 3 - Diagrama de caso de uso .....	12
Figura 4 - Arquitetura de aplicação .....	13
Figura 5 – Diagrama de Classe .....	15
Figura 6 - Implementação da classe Trail .....	16
Figura 7 - Implementação da classe User .....	17
Figura 8 - Página Inicial do Site.....	18
Figura 9 - Página de Login e Cadastro .....	19
Figura 10 - Página de Conteúdo .....	19
Figura 11 - Página Vestibulares e Fatec Votorantim .....	20

## **1.Descrição do projeto**

### **1.1. Proposta do Software (Objetivo)**

O projeto consiste na elaboração de um site voltado para a educação, com o objetivo de fornecer ensino gratuito e de qualidade para pessoas que tem interesse em ingressar nas áreas de tecnologia (mais especificamente no nicho de desenvolvimento).

A estrutura desenhada para o site foi dividida em três partes: uma breve explicação sobre a FATEC, assim como uma análise do curso e da área, e a parte de conteúdos, esta que tem como maior objetivo fornecer materiais aprendidos em sala de aula, assim como vídeo aulas, links e sugestões de cursos. O usuário pode acessar todo o conteúdo de maneira fácil, além de melhorar seu desempenho durante os estudos.

### **1.2. Justificativa**

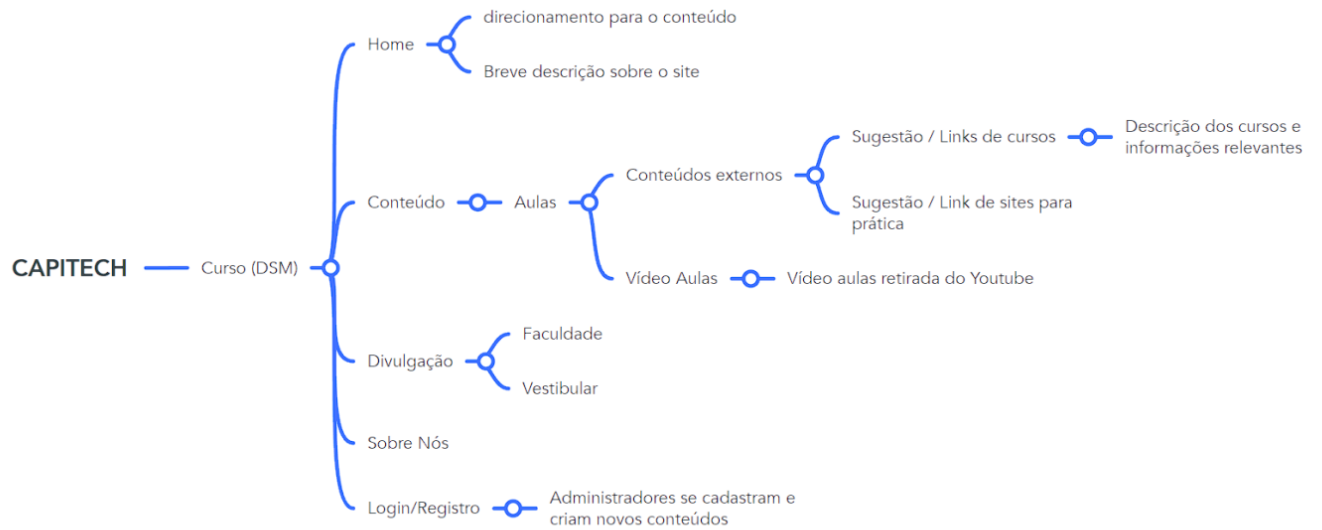
Este projeto visa suprir a demanda por educação tecnológica acessível, oferecendo uma plataforma centrada no desenvolvimento e fornecendo oportunidades de aprendizado gratuito na área. Motivação: A crescente importância da tecnologia e as limitações no acesso à educação inspiram este projeto, visando democratizar conhecimento e capacitar aqueles interessados no desenvolvimento.

Além de fornecer conhecimento técnico, a plataforma busca impactar positivamente a sociedade, formando profissionais qualificados e promovendo inovação e inclusão digital.

A crescente importância da tecnologia e as limitações no acesso à educação inspiram este projeto, visando democratizar conhecimento e capacitar aqueles interessados no desenvolvimento. Além de fornecer conhecimento técnico, a plataforma busca impactar positivamente a sociedade, formando profissionais qualificados e promovendo inovação e inclusão digital.

### 1.3. Mapa Mental

Figura 1 - Mapa do site



Fonte: Autoria Própria

## 1.4. Logomarca

Figura 2 - Logo e Nome do Projeto - Aprendendo com quem aprende



Fonte: Autoria Própria



## 2. Requisitos do projeto

### 2.1. Levantamento de Requisitos

Para o levantamento de requisitos do sistema, as principais ideias foram reunidas utilizando como base alguns blogs e sites de cursos que já estão na web e que ajudam, diariamente, inúmeros alunos em seus respectivos estudos acadêmicos. Dessa forma, foi possível elaborar as funcionalidades que o software deve ter.

Abaixo, estão listados os exemplos de alguns sites e blogs utilizados para a definição do projeto:

- **Curso em Vídeo (Prof. Gustavo Guanabara):** <https://www.cursoemvideo.com/>
- **DevMedia:** <https://www.devmedia.com.br/>
- **W3schools:** <https://www.w3schools.com/>
- **Alura:** <https://www.alura.com.br/artigos>

## 2.2. Backlogs & User Stories

Um "user story" é uma técnica usada no desenvolvimento ágil de software para descrever uma funcionalidade do ponto de vista do usuário final. Cada história inclui uma persona do usuário, descreve sua necessidade específica e o valor esperado ao utilizar essa funcionalidade. Essas histórias ajudam a equipe a entender e priorizar o desenvolvimento com foco nas necessidades reais dos usuários, promovendo uma comunicação clara e alinhada entre todos os membros do projeto.

User Story			
ID Req	Ator	Ação	Motivo
1	Aluno/ Admin	Acessar página principal	Ver as funcionalidades do site
2	Aluno/ Admin	Acessar menu "hamburger"	Facilitar a navegação entre abas
3	Aluno/ Admin	Exibir conteúdos	Acessar as diferentes conteúdo do site
4	Admin	Disponibilizar vídeos	Para acessar outras de fontes de informação
5	Admin	Disponibilizar links	Para acessar outras de fontes de informação
6	Admin	Disponibilizar link para o vestibular FATEC	Promover a procura pela faculdade
7	Admin	Cadastro de ADMINS	Para que os conteúdos possam ser atualizados com segurança
8	Admin	Login de ADMINS	Para que os admins possam entrar e alterar os conteúdos
9	Admin	Alteração de senha para ADMINS	Alterar a senha se necessário
10	Admin	Exibir página de criação de posts	Facilitar a criação de conteúdo do site

## 2.3. Requisitos funcionais

Requisitos funcionais são descrições específicas das ações que um sistema deve realizar em resposta a entradas do usuário ou de outros sistemas, como login, busca de produtos e gestão de pedidos. Eles são essenciais para guiar o desenvolvimento de software, garantindo que o sistema execute as funcionalidades necessárias de maneira adequada.

Requisitos funcionais	
1	Exibir a página principal com explicações do site para o usuário.
2	Exibir a navegação entre as páginas "home", "conteúdos", "vestibular" e "sobre nos" por meio do menu hambúrguer e do rodapé do site.
3	Exibir os conteúdos
4	Disponibilizar vídeos explicativos da web sobre os conteúdos.
5	Disponibilizar links para arquivos de estudo sobre os conteúdos.
6	Disponibilizar o link de inscrição no vestibular Fatec.
7	Implementar área de cadastro para os usuários administradores.
8	Implementar área de login para os usuários administradores.
9	Implementar área de recuperação de senha para os usuários administradores.
10	Exibir página de criação de "posts" para os usuários com acesso de administradores.
11	Usuários considerados administradores realizam a criação e alteração de novos "posts" no site.

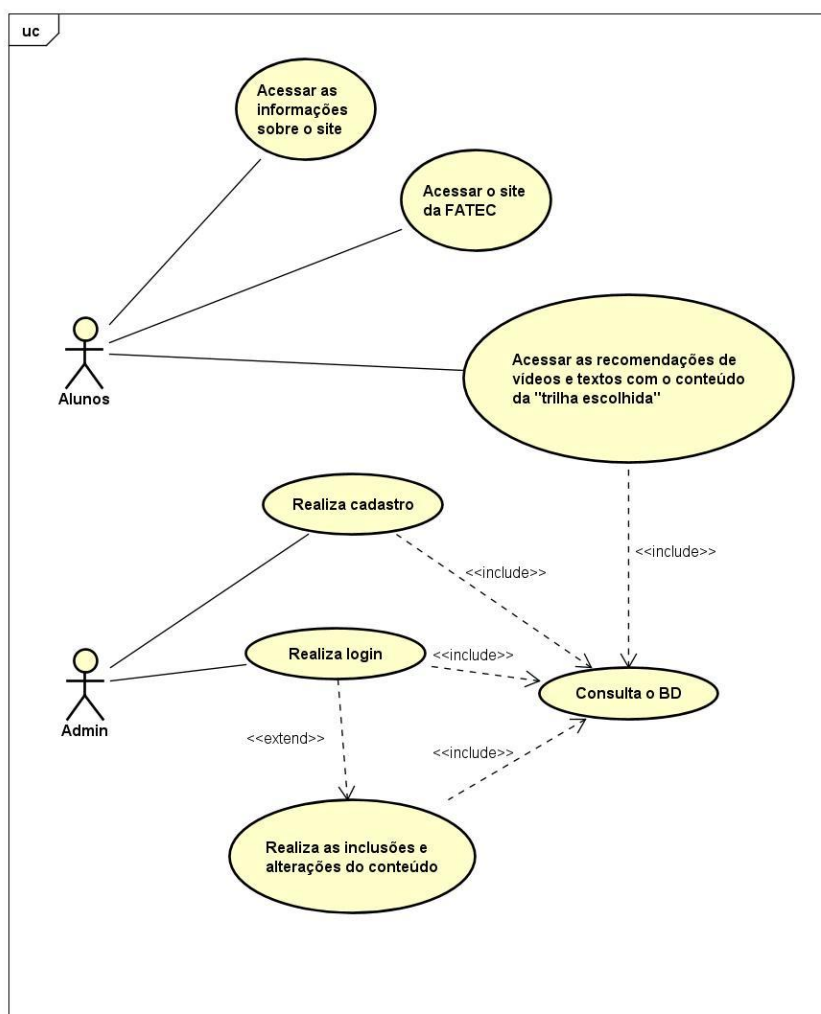
## 2.4. Requisitos não funcionais

Requisitos não funcionais são critérios que descrevem as qualidades e as restrições que um sistema deve possuir, além das funcionalidades específicas que ele deve executar. Esses requisitos geralmente abordam aspectos como desempenho, segurança, usabilidade, confiabilidade e compatibilidade, e são cruciais para garantir que o sistema atenda aos padrões e expectativas de qualidade e eficiência esperados pelos usuários e stakeholders.

Requisitos não funcionais	
1	utilizar HTML/CSS/JavaScript e TypeScript
2	utilizar Node.js para o backend
3	utilizar banco de dados MongoDB
4	utilizar framework Express.js para desenvolvimento backend
5	utilizar frameworks React e TailWind para a construção do site
6	Interface principal voltada para o usuário
7	Interface agradável e com fácil visualização
8	Interface com fontes de fácil compreensão
9	Software hospedado na web, acesso somente em ambientes com internet
10	Interface do sistema responsiva

## 2.5. Diagrama de Caso de Uso

Figura 3 - Diagrama de caso de uso

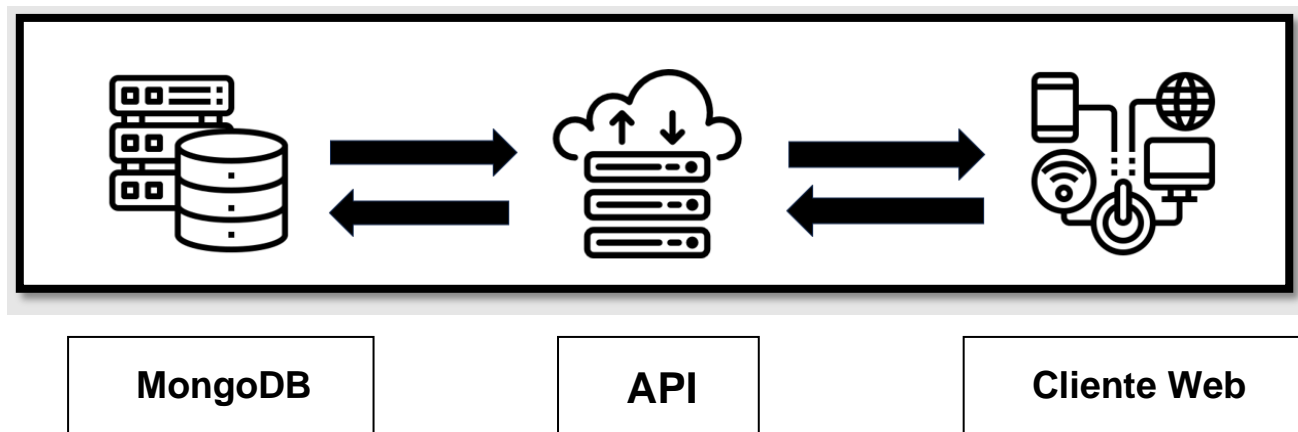


Fonte: Autoria Própria

## PROJETO DO SOFTWARE

### 2.6. Arquitetura da Aplicação

Figura 4 - Arquitetura de aplicação



Fonte: Autoria própria

### 2.7. Tecnologias Utilizadas

- Visual Studio Code – Microsoft<sup>1</sup>: Um editor de código-fonte desenvolvido pela Microsoft para Windows, Linux e MacOS. Inclui suporte para depuração, controle de versionamento Git incorporado, realce de sintaxe, complementação inteligente de código, snippets e refatoração de código;
- HTML<sup>2</sup> - Linguagem de Marcação de Hipertexto: Hypertext Markup Language (Linguagem de Marcação de Hipertexto). É fundamental para a internet que conhecemos atualmente por trazer recursos que facilitaram a estruturação e apresentação dos conteúdos na web;
- CSS – Folhas de Estilo Em Cascata<sup>3</sup>: usada para descrever a apresentação de um documento escrito em HTML. O CSS descreve como elementos são mostrados na tela, no papel, na fala ou em outras mídias;
- Javascript – Linguagem de Programação<sup>4</sup>: É uma linguagem de programação de comportamento que permite a criação de conteúdos dinâmicos, controle de mídias e animações;
- React<sup>5</sup>: React é uma biblioteca JavaScript que permite criar UIs interativas com componentes declarativos, baseados em componentes e flexíveis.

<sup>1</sup> <https://code.visualstudio.com/>

<sup>2</sup> <https://www.techtudo.com.br/listas/2023/06/o-que-e-html-5-edsoftwares.ghtml>

<sup>3</sup> <https://developer.mozilla.org/pt-BR/docs/Web/CSS>

<sup>4</sup> <https://www.hostinger.com.br/tutoriais/o-que-e-javascript>

<sup>5</sup> <https://react.dev/>

- Flaticon<sup>6</sup>: O Flaticon é uma plataforma online de ícones animadores e em diferentes formatos, gratuitos ou pagos
- Figma<sup>7</sup>: Um editor gráfico de vetor e prototipagem de projetos de design baseado principalmente no navegador web, com ferramentas offline adicionais para aplicações desktop
- Insomnia<sup>8</sup>: Uma plataforma colaborativa de desenvolvimento de API de código aberto que facilita a construção de APIs de alta qualidade.
- TypeScript<sup>9</sup>:: Uma linguagem de programação que adiciona tipos estáticos opcionais ao JavaScript, sendo amplamente utilizada em projetos JavaScript de grande escala para melhorar a escalabilidade e manutenção do código
- Node.js<sup>10</sup>: Um ambiente de execução JavaScript baseado no motor V8 do Chrome, que permite executar JavaScript fora de um navegador. É usado para construir aplicativos de rede escaláveis.
- Express.js<sup>11</sup>: Um framework web rápido, não opinativo e minimalista para Node.js, que facilita a construção de APIs e aplicativos web.
- Postman<sup>12</sup>: Uma ferramenta colaborativa de desenvolvimento de APIs que simplifica a criação, compartilhamento, teste e documentação de APIs.

---

<sup>6</sup> <https://www.flaticon.com/br/>

<sup>7</sup> <https://www.figma.com/>

<sup>8</sup> <https://insomnia.rest/>

<sup>9</sup> <https://www.typescriptlang.org/>

<sup>10</sup> <https://nodejs.org/>

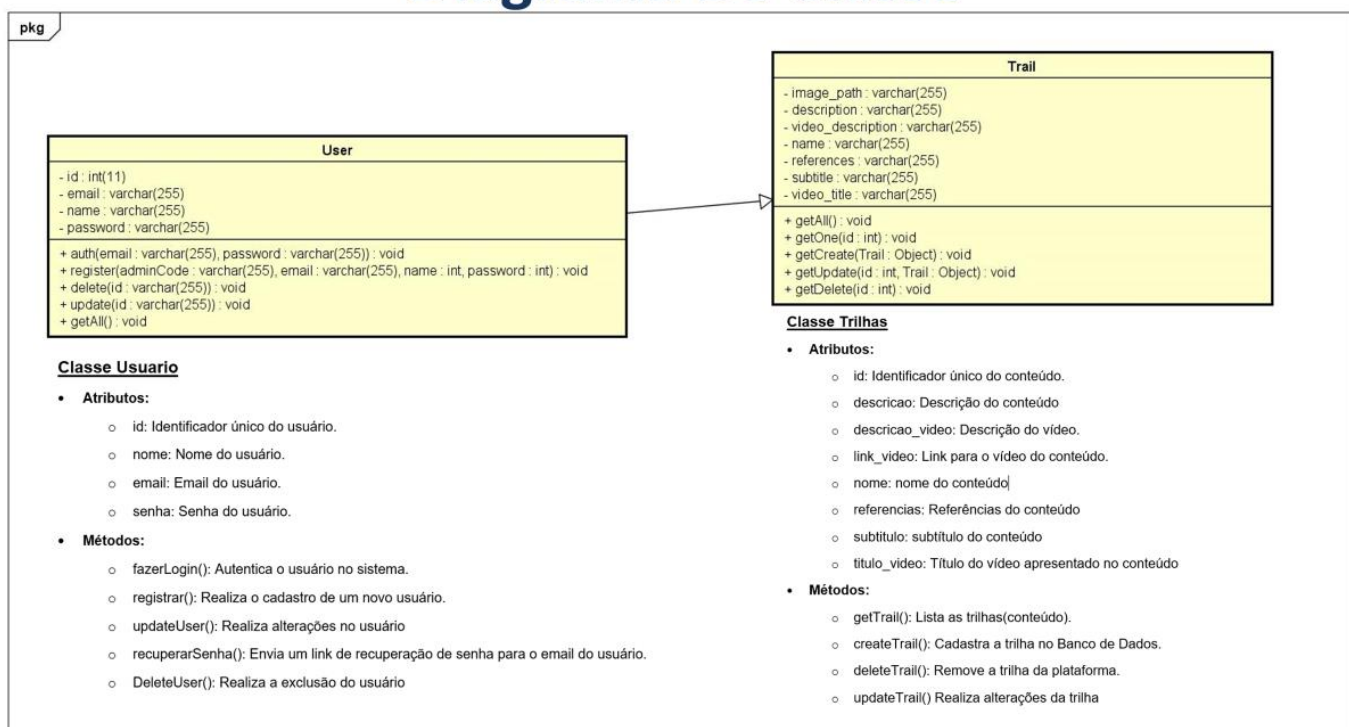
<sup>11</sup> <https://expressjs.com/pt-br/>

<sup>12</sup> <https://www.postman.com/>

## 2.8. Diagrama de Classe

Figura 5 – Diagrama de Classe

# Diagrama De Classe



powered by Astah

## 2.9. Implementação Das Classes

Para elucidar um pouco mais sobre como seria a implementação das classes nós trouxemos aqui dois exemplos de nosso projeto.

Para tal, utilizamos o método `getOne()` que está dentro da classe “Trilhas”. Como descrevemos acima, esse método retorna um registro do curso, mas vamos detalhar isso utilizando essa próxima imagem.

Figura 6 - Implementação da classe Trail

# Implementação das Classes

Classe: Trail

Método: `getOne()`

```
async getOne(req: Request<{ id: string }, unknown, unknown>, res: Response) {
  try {
    const { id } = req.params;

    const trail = await Trail.findById(id);

    if (!trail) {
      return SendResponse.error(res, 404, 'Trilha não encontrada');
    }

    return SendResponse.success(
      res,
      200,
      'Trilha encontrada com sucesso',
      trail
    );
  } catch (error) {
    console.log('❗ ~ TrailController ~ getOne ~ error:', error);
    return SendResponse.error(res, 500, 'Erro ao buscar trilha');
  }
}
```



Utilizaremos também um método da classe “Usuário”, trata-se do método: register(), esse método é utilizado para realizar o cadastro de um novo usuário. Ele também será detalhado melhor com o auxílio da próxima imagem.

Figura 7 - Implementação da classe User

# Implementação das Classes

## Classe: User

### Método: register()

### Método register() escrito

```
async register(
  req: Request<unknown, unknown, AuthRegisterProps>,
  res: Response
) {
  try {
    const { adminCode, email, password, name } = req.body;

    if (!adminCode || !email || !password) {
      return SendResponse.error(
        res,
        404,
        'Campos obrigatórios não preenchidos'
      );
    }

    const userExists = await User.findOne().where('email').equals(email);

    if (userExists) {
      return SendResponse.error(res, 400, 'Usuário com e-mail já cadastrado');
    }

    const saltRounds = 10;
    const salt = bcrypt.genSaltSync(saltRounds);
    const hashedPassword = bcrypt.hashSync(password, salt);
    // You, há 4 dias • ↳ Fast: user controller, model and routes
    const newUser = { email, password: hashedPassword, name };

    const createdUser = await User.create(newUser);

    return SendResponse.success(
      res,
      201,
      'Usuário criado com sucesso',
      createdUser
    );
  } catch (error) {
    console.log('❌ ~ UserController ~ auth ~ error:', error);
    return SendResponse.error(res, 500, 'Erro ao cadastrar usuário');
  }
}
```

### Página de cadastro

## 2.10. Interfaces com o usuário

Figura 8 - Página Inicial do Site

# Site Desenvolvido

## Página inicial



**Fatec**  
Votorantim

**CPQ**  
Centro  
Paula Souza

**capi tech**

Fonte: Autoria Própria

Figura 9 - Página de Login e Cadastro

## Site Desenvolvido



Fatec  
Votorantim

CPS  
Centro  
Paula Souza

capi tech

Fonte: Autoria Própria

Figura 10 - Página de Conteúdo

## Site Desenvolvido

### Página de conteúdo



Fatec  
Votorantim

CPS  
Centro  
Paula Souza

capi tech

Fonte: Autoria Própria

Figura 11 - Página Vestibulares e Fatec Votorantim



Fonte: Autoria Própria

### 3. ESTRATÉGIA DE TESTES

A estratégia de testes é executar diferentes tipos de testes para descobrir erros de implementação ou de projeto.

**Testes Unitário:** consiste em verificar os métodos de inserção, deleção, edição e visualização de dados de acordo com as rotas criadas.

Para realização de teste foi utilizada a ferramenta Insomnia. O objetivo dessa ferramenta é facilitar a criação e testes das rotas de requisições HTTP (Hypertext Transfer Protocol), verificando se os métodos chamados no banco de dados correspondentes às rotas criadas funcionam da maneira esperada.

**Teste de Sistema:** Os testes de sistema não se limitam aos requisitos funcionais, mas também objetiva testar os requisitos não-funcionais. Os testes de aceitação são, em geral, uma extensão dos testes de sistema. Durante esta fase, o objetivo é verificar se o software está pronto e pode ser usado pelo usuário final.

Para realização deste teste foi utilizado uma pessoa que não tem envolvimento em nosso desenvolvimento para testar todas as funcionalidades que envolvessem o usuário, como menus, links e possíveis erros que pudessem ocorrer como: realizar login sem nenhuma senha; tentar alterar conteúdo sem ser administrador.

**Teste de Usabilidade:** é uma técnica de caixa-preta. O objetivo é observar usuários reais usando o produto para descobrir problemas e pontos de melhorias.

O teste de usabilidade é uma técnica que visa avaliar a qualidade de uma interface ao colocá-la em uso em situações comuns. Ele busca entender como o sistema se comporta no dia a dia, na naturalidade da utilização, e se ele atende aos requisitos pensados e estabelecidos. Em outras palavras, a usabilidade é a facilidade com que os usuários lidam com uma ferramenta, site ou produto.

## 4. IMPLANTAÇÃO

- O código fonte da aplicação pode ser acessado no repositório do GitHub no seguinte endereço:

Readme: <https://github.com/FischerJoao/readme-Capitech/blob/main/README.md>

Back-end: <https://github.com/alihpss/capitech-final>

Front-end: <https://github.com/Joel-Neto/capitech-front-final>

- Atualmente hospedado no endereço do Vercel: <https://capitech-front-final.vercel.app/>
- No futuro, caso o site seja desenvolvido para acesso a grande massa, vai ser hospedado e aberto para todos com novas funcionalidades.

## 5. REFERÊNCIAS

<sup>1</sup> conforme disponível em: < <http://pmkb.com.br/sig/padroes-frameworks/pmbok-pmi/>>. Acesso em: 10 jul. 2020.

SOMMERVILLE, Ian. **Engenharia de Software**. 8ª ed. Pearson, 2007

PRESSMAN, Roger; MAXIM, Bruce. Engenharia de Software. 8 ed. São Paulo: McGraw Hill Brasil, 2016.

SOMMERVILLE, Ian. Engenharia De Software. 10 ed. São Paulo: Pearson Brasil, 2019

STELLMAN, Andrew; GREENE, Jennifer. Use a Cabeça! Ágil. Rio de Janeiro: Alta Books, 2019.