

Curso Superior de Desenvolvimento de Software Multiplataforma

Alexsander Ramos Ferreira, RA 3011392313001 Cláudio Vinicius de Almeida, RA 3011392313009

Gabriel Fortes Ashikaga, RA 3011392313029

João Cunha Fischer, RA 3011392313014

Joel de Farias Alves Neto, RA 3011392313033

Lucas Kauã M Sampaio, RA 3011392313016

Projeto Interdisciplinar

Programação para Dispositivos Móveis II

Segurança no Desenvolvimento de Aplicações

Computação em Nuvem I

Laboratório de Desenvolvimento para Dispositivos Móveis

Aprendizagem de Máquina

Fundamentos da Redação Técnica

CAPI.TECH

Orientadores: Emerson De Oliveira Rocha

Junio César Dos Santos Gonçalves

Felipe Santos Medeiros

Helder Pestana

Waldemar Bonventi Junior

Rosana Helena Nunes

Votorantim

Junho, 2025

📱 RESUMO

O projeto em foco visa desenvolver um aplicativo educacional gratuito com ênfase em tecnologia, utilizando ferramentas como Visual Studio Code, TypeScript, Node.js, Express.js, React Native, Expo, Yarn, GitHub, N8N, Prisma, MySQL, MongoDB, AWS, Docker, DBeaver, MongoDB Compass, Mongoose, React Icons, Flaticon, Figma, Insomnia e Postman. A plataforma é estruturada em três componentes cruciais.

Inicialmente, o aplicativo apresenta brevemente a FATEC, proporcionando aos usuários um entendimento introdutório sobre a instituição. Em seguida, oferece uma análise detalhada sobre o curso de Desenvolvimento de Software Multiplataforma e o contexto mais amplo da área de tecnologia.

A parte central do aplicativo é dedicada à disponibilização de conteúdo educacional, oferecendo aos usuários acesso a uma variedade de materiais, como videoaulas, links úteis, sugestões de cursos adicionais, além de um chatbot educativo, o CapiBot, que apoia os estudos por meio de pesquisas orientadas na área de programação.

Uma característica distintiva do projeto é a facilidade de navegação, mesmo em ambiente mobile, e a capacidade dos usuários de melhorarem seu desempenho nos estudos. Essa abordagem visa tornar o aprendizado mais acessível, responsivo e personalizado, incentivando a exploração e o aprimoramento contínuo na área da tecnologia.

Em resumo, a iniciativa busca oferecer uma plataforma aberta, acessível e gratuita para aprendizado em tecnologia, promovendo inclusão digital e formação de profissionais qualificados, por meio de uma experiência moderna, prática e alinhada às tendências atuais de desenvolvimento mobile e backend.

SUMÁRIO

Sumário

[📱 RESUMO 4](#_Toc199357439)

[SUMÁRIO 5](#_Toc199357440)

[1. Descrição do projeto 7](#_Toc199357441)

[Proposta do Software (Objetivo) 7](#_Toc199357442)

[Justificativa 7](#_Toc199357443)

[Mapa Mental 8](#_Toc199357444)

[Logomarca 9](#_Toc199357445)

[2. Requisitosdoprojeto 10](#_Toc199357446)

[Levantamento de Requisitos 10](#_Toc199357447)

[Backlogs & UserStories 11](#_Toc199357448)

[Requisitos funcionais 12](#_Toc199357449)

[Requisitos não funcionais 12](#_Toc199357450)

[Arquitetura da Aplicação 14](#_Toc199357451)

[TecnologiasUtilizadas 14](#_Toc199357452)

[DiagramadeClasse 16](#_Toc199357453)

[Implementação Das Classes 17](#_Toc199357454)

[Interfacescom ousuário 19](#_Toc199357455)

[3. ESTRATÉGIADETESTES 21](#_Toc199357456)

[4. IMPLANTAÇÃO 22](#_Toc199357457)

[5. REFERÊNCIAS 22](#_Toc199357458)

**INDICE DE IMAGENS**

[Figura 1-Mapa do site 7](#_bookmark4)

[Figura 2 - Logo e Nome do Projeto – Aprendendo com quem aprende 8](#_bookmark6)

[Figura 3 – Arquiteturade aplicação 13](#_bookmark15)

[Figura 4 – Página Inicial do Site 18](#_bookmark23)

[Figura 5 – Página de Login e Cadastro 19](#_bookmark24)

[Figura 6 – Página de Conteúdo 19](#_bookmark25)

[Figura 7- Página Vestibulares e Fatec Votorantim 20](#_bookmark26)

1. Descrição do projeto
   1. Proposta do Software (Objetivo)

O projeto consiste na elaboração de um aplicativo mobile voltado para a educação, com o objetivo de fornecer ensino gratuito e de qualidade para pessoas interessadas em ingressar nas áreas de tecnologia — mais especificamente no nicho de desenvolvimento de software.

A estrutura do aplicativo foi dividida em três partes: uma breve apresentação sobre a FATEC; uma análise sobre o curso de Desenvolvimento de Software Multiplataforma e o panorama da área de tecnologia; e, por fim, a seção de conteúdos educacionais. Esta última tem como principal objetivo disponibilizar materiais aprendidos em sala de aula, além de videoaulas, links úteis, sugestões de cursos e um chatbot educativo chamado CapiBot.

O aplicativo foi projetado para ser simples, intuitivo e responsivo, permitindo que o usuário acesse todo o conteúdo com facilidade, contribuindo assim para a melhoria de seu desempenho nos estudos e incentivando a exploração contínua do conhecimento em tecnologia.

.

* 1. Justificativa

Este projeto visa suprir a crescente demanda por educação tecnológica acessível, oferecendo um aplicativo educacional centrado no desenvolvimento de software e promovendo oportunidades de aprendizado gratuito e de qualidade.

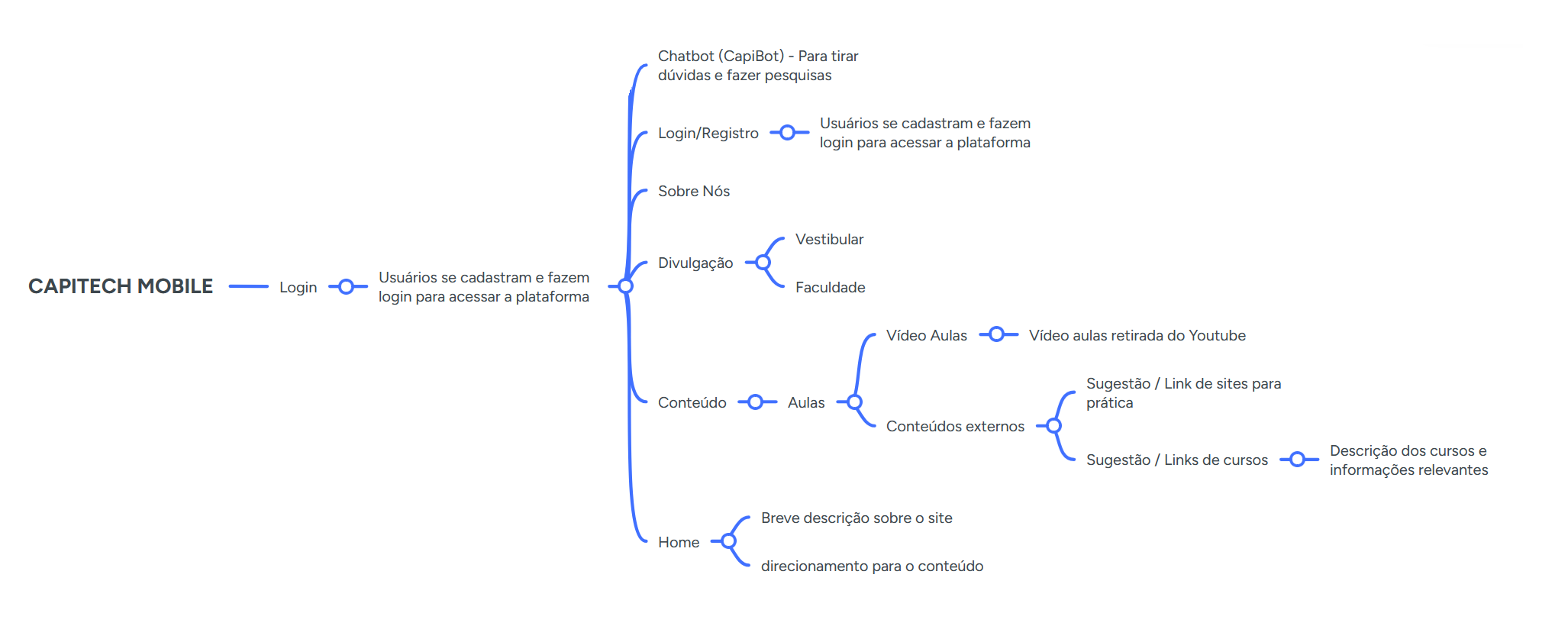
A motivação parte do reconhecimento de que a tecnologia ocupa um papel cada vez mais central na sociedade, ao mesmo tempo em que barreiras de acesso à educação ainda persistem. Assim, o projeto busca democratizar o conhecimento e capacitar indivíduos interessados na área de desenvolvimento, especialmente aqueles que não têm fácil acesso a recursos formais de ensino.

Além de transmitir conhecimento técnico, o aplicativo se propõe a impactar positivamente nossa sociedade, formando profissionais qualificados, incentivando o autoestudo, promovendo a inclusão digital e fomentando a inovação por meio da tecnologia.

Com recursos como um chatbot educativo (CapiBot), conteúdos multimídia e navegação mobile amigável, o projeto pretende ser uma ponte entre o potencial dos estudantes e as exigências do mercado de trabalho na era digital.

* 1. Mapa Mental

Figura1-Mapa do site



Fonte: Autoria Própria

* 1. Logomarca

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



Fonte: Autoria Própria

1. Requisitos doprojeto
   1. Levantamento de Requisitos

Para o levantamento de requisitos do sistema, as principais ideias foram reunidas com base na observação e análise de blogs, plataformas de cursos e soluções educacionais online já consolidadas, que auxiliam milhares de estudantes diariamente em suas jornadas de aprendizado. A partir dessa pesquisa, foi possível identificar e adaptar funcionalidades que atendem às necessidades do público-alvo, respeitando a usabilidade no ambiente mobile e a proposta pedagógica da aplicação.

O objetivo principal do levantamento foi garantir que o aplicativo Capitech Mobile ofereça uma experiência fluida, útil e acessível, tanto para iniciantes quanto para estudantes em fase mais avançada, sempre promovendo o autoestudo e a inclusão digital.

Abaixo estão listadas algumas das fontes utilizadas como inspiração e referência para a construção da proposta funcional do aplicativo:

* + - **Curso em Vídeo(Prof. Gustavo Guanabara)**: <https://www.cursoemvideo.com/>
    - **DevMedia:** <https://www.devmedia.com.br/>
    - **W3 schools:** <https://www.w3schools.com/>
    - **Alura:** https://www.alura. com.br/artigos
  1. Backlogs & UserStories

Uma user story é uma técnica amplamente utilizada no desenvolvimento ágil de software para descrever funcionalidades a partir da perspectiva do usuário final. Cada história inclui uma persona (perfil de usuário), define sua necessidade específica e explicita o valor esperado ao utilizar determinada funcionalidade dentro do aplicativo. Essas histórias são fundamentais para orientar a equipe durante o desenvolvimento, permitindo uma compreensão clara das prioridades do produto e promovendo comunicação eficiente entre desenvolvedores, designers, testadores e stakeholders.

No contexto do Capitech Mobile, as user stories foram definidas com base nas funcionalidades identificadas durante o levantamento de requisitos e organizadas em backlog, priorizando as entregas mais relevantes para os usuários.

|  |  |  |  |
| --- | --- | --- | --- |
| **User Story** | | | |
| **ID Req** | **Ator** | **Ação** | **Motivo** |
| **1** | Aluno | Acessar página principal | Ver as funcionalidades do site |
| **2** | Aluno | Acessar menu "hamburguer" | Facilitar a navegação entre abas |
| **3** | Aluno | Exibir conteúdos | Acessar os diferentes conteúdo do site |
| **4** | Aluno | Visualizar os vídeos | Para acessar outras de fontes de informação |
| **5** | Aluno | Acessar os links | Para acessar outras fontes de informação |
| **6** | Aluno | Interage com chatbot | Para tirar dúvidas e fazer pesquisas |
| **7** | Aluno | Acessa aba de noticías voltadas para tecnologia | Para se manter atualizado sobre tendências |
| **8** | Aluno | Acessar link para o vestibular FATEC | Promover a procura pela faculdade |
| **9** | Aluno | Cadastro de aluno | Para que possa criar uma conta e acessar os conteúdos |
| **10** | Aluno | Login de aluno | Para que possa fazer o login e acessar os conteúdos |
| **11** | Aluno | Alteração de senha para os alunos | Alterar a senha se necessário |

* 1. 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, visualização de conteúdo, interação com o chatbot, entre outros.

Eles são essenciais para guiar o desenvolvimento de software, garantindo que o aplicativo execute corretamente as funcionalidades previstas.

No Capitech Mobile, os requisitos funcionais foram definidos para garantir que os usuários possam acessar o conteúdo educacional, interagir com ferramentas complementares e navegar com facilidade pela plataforma:

|  |  |
| --- | --- |
| **Requisitos funcionais** | |
| **1** | Exibir a página principal comexplicações do site para o usuário. |
| **2** | Exibir a navegação entre as páginas "home", "conteúdos", "vestibular" e "sobrenos" por meio do menu hamburguer 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** | Disponibizar um Chatbot para sanar dúvidas e fazer pesquisas |
| **8** | Implementar área de cadastro para os usuários alunos. |
| **9** | Implementar área de notícias para leitura |
| **10** | Implementar área de login para os usuários alunos. |
| **11** | Implementar área de recuperação de senha para os alunos. |
| **12** | Exibir página de criação de "posts" para os alunos. |
| **13** | Usuários considerados administradores realizam a criação e alteração de novos "posts" no site. |

* 1. Requisitos não funcionais

Requisitos não funcionais são critérios que descrevem as qualidades e restrições que um sistema deve atender, indo além das funcionalidades que ele executa.

Eles abordam aspectos como desempenho, usabilidade, segurança, compatibilidade e confiabilidade, sendo essenciais para garantir que o sistema esteja de acordo com os padrões e expectativas de qualidade esperados pelos usuários e stakeholders.

Para o Capitech Mobile, os requisitos não funcionais foram definidos com base nas tecnologias utilizadas, no contexto de uso do aplicativo e nos objetivos do projeto:

|  |  |
| --- | --- |
| **Requisitos não funcionais** | |
| **1** | Utilizar **TypeScript** como linguagem base do projeto |
| **2** | Utilizar **React Native** e **Expo** para construção da interface mobile |
| **3** | Utilizar **Node.js** com **Express.js** no desenvolvimento do backend |
| **4** | Utilizar **Prisma** e **Mongoose** como **ORM** e **MySQL** e **MongoDB** como bancos de dados |
| **5** | Utilizar **Docker** para padronizar e orquestrar, simplificando o ambiente de desenvolvimento |
| **6** | Utilizar **AWS** para possíveis serviços em nuvem e escalabilidade |
| **7** | Utilizar ferramentas de apoio como **N8N**, **DBeaver** e **MongoDB Compass** |
| **8** | Utilizar **Postman** e **Insomnia** para testes de APIs |
| **9** | Interface pensada para o usuário mobile, com navegação fluida e experiência nativa |
| **10** | Interface com design responsivo e usabilidade acessível e intuitiva |
| **11** | Utilizar **Figma** para criação de protótipos e layout |
| **12** | Utilizar ícones visuais e funcionais como os do **React Icons** e **Flaticon** |
| **13** | Utilizar **GitHub** para versionamento e controle colaborativo do código |

PROJETO DO SOFTWARE

* 1. Arquitetura da Aplicação

*Figura 4 - Arquitetura de aplicação*

**ClienteWeb / APP**

**MongoDB / MySQL**

**API**



Fonte: Autoria própria

* 1. Tecnologias Utilizadas
     + **Visual Studio Code – Microsoft1**: 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;
     + **TypeScript2**- Linguagem que adiciona tipagem estática opcional ao JavaScript, ideal para projetos de larga escala, aumentando a robustez e a manutenibilidade do código.
     + **React Native3**: Framework baseado em JavaScript que permite o desenvolvimento de aplicativos mobile nativos para Android e iOS a partir de um único código base;
     + **Expo4**: Plataforma que acelera o desenvolvimento com React Native, oferecendo ferramentas e bibliotecas para build, testes e deploy de aplicativos mobile;
     + **Yarn5**: Gerenciador de pacotes rápido e seguro, utilizado para instalar bibliotecas no ambiente Node.js/React Native.

1https://code.visualstudio.com/

2https://www.typescriptlang.org/

3https://reactnative.dev/

4https://expo.dev/

5https://yarnpkg.com/

* + - **Node.js6**: Ambiente de execução JavaScript no lado do servidor, amplamente usado para construir backends escaláveis e APIs.
    - **Express.js7**: Framework minimalista para Node.js que simplifica a criação de servidores e APIs REST.
    - **Prisma:8**: ORM (Object-Relational Mapping) moderno para Node.js e TypeScript, usado para modelar, consultar e manipular bancos de dados como MySQL e MongoDB.
    - **MySQL9**: Sistema de gerenciamento de banco de dados relacional (RDBMS), amplamente utilizado para armazenar dados estruturados.
    - **MongoDB10**: Banco de dados NoSQL orientado a documentos, ideal para armazenar dados flexíveis e não estruturados.
    - **Docker11**: Plataforma de containers utilizada para padronizar ambientes de desenvolvimento e facilitar o deploy de aplicações.
    - **AWS (Amazon Web Services)12**: Plataforma de serviços em nuvem usada para escalar, hospedar e gerenciar aplicações.
    - **DBeaver:13**: Ferramenta universal de gerenciamento de banco de dados que permite visualizar e manipular dados em bancos como MySQL.
    - **MongoDB Compass14**: Interface gráfica oficial do MongoDB para exploração, visualização e gerenciamento de documentos.
    - **N8N15**: Plataforma de automação de fluxos (workflow automation) que permite conectar APIs e serviços com pouca ou nenhuma programação.
    - **Figma16**: Ferramenta colaborativa para design de interface e prototipação, ideal para criar wireframes e layouts responsivos.
    - **Flaticon:17**: Biblioteca online de ícones vetoriais em diversos estilos e formatos, utilizada para melhorar a experiência visual da interface.
    - **React Icons18**: Coleção de pacotes de ícones prontos para uso com React e React Native, oferecendo integração simples com bibliotecas populares como FontAwesome.
    - **Postman19**: Plataforma para criação, teste e documentação de APIs RESTful, utilizada para validar endpoints do backend.
    - **Insomnia20**: Ferramenta de código aberto para desenvolvimento, testes e organização de APIs, útil durante a etapa de testes e integração.

6https://nodejs.org/

7https://expressjs.com/pt-br/

8https://www.prisma.io/

9https://www.mysql.com/

10https://www.mongodb.com/

11https://www.docker.com/

12https://aws.amazon.com/pt/

13https://dbeaver.io/

14https://www.mongodb.com/products/compass

15https://n8n.io/

16https://www.figma.com/

17https://www.flaticon.com/br/

18https://react-icons.github.io/react-icons/

19https://www.postman.com/

20https://insomnia.rest

* 1. Interfaces com o usuário

*Figura 8-Página Inicial do Site*

**

*Fonte:Autoria Própria*

*Figura9-Páginade Login e Cadastro*

**

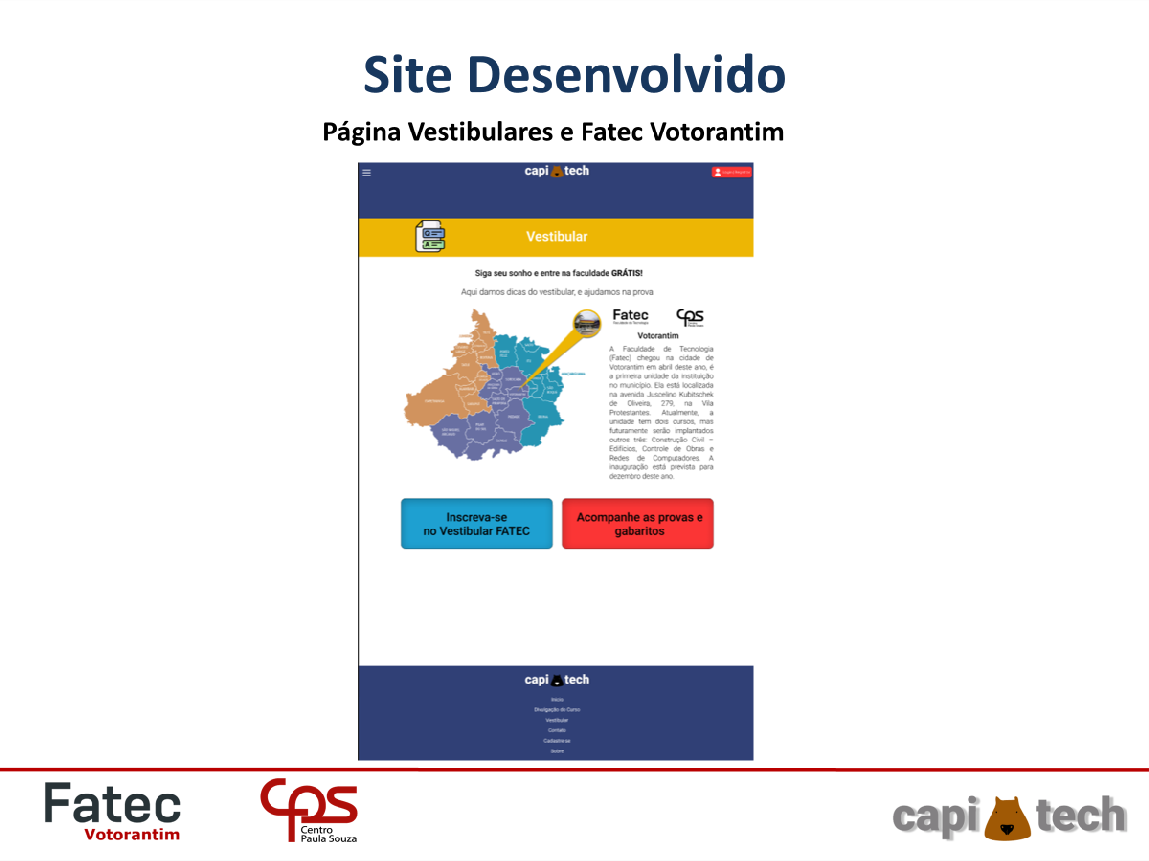
*Fonte:Autoria Própria*

*Figura10-Página de Conteúdo*

**

*Fonte:Autoria Própria*

*Figura11-PáginaVestibulares e Fatec Votorantim*

**

*Fonte:Autoria Própria*

1. ESTRATÉGIA DE TESTES

A estratégia de testes adotada visa validar a qualidade, integridade e funcionalidade do aplicativo Capitech Mobile, identificando possíveis erros de implementação ou falhas de design. Foram utilizados diferentes tipos de testes ao longo do desenvolvimento:

**Testes Unitários:** Foram realizados testes unitários sobre métodos responsáveis por operações básicas como inserção, atualização, exclusão e visualização de dados, especialmente nas rotas do backend (Node.js + Express + Prisma).

Para esses testes, utilizamos ferramentas como Insomnia e Postman, que permitem validar requisições HTTP e verificar se os dados retornados do banco (MySQL ou MongoDB) correspondem às operações esperadas.

**Teste de Sistema:** Os testes de sistema avaliam o comportamento geral do aplicativo, verificando não só os requisitos funcionais, mas também aspectos de desempenho, segurança e usabilidade (requisitos não funcionais).

Realizamos testes com um usuário externo ao time de desenvolvimento para simular o uso real do app, identificando comportamentos inesperados, como:

* Tentativa de login sem preenchimento de dados
* Acesso a funcionalidades restritas sem autenticação
* Verificação de links e redirecionamentos
* Navegação entre telas e menus

**Teste de Usabilidade:** A usabilidade foi avaliada com foco na experiência do usuário mobile, observando sua facilidade de navegação, clareza visual e tempo de aprendizado com o aplicativo.

Utilizamos a técnica de caixa-preta, onde observamos o comportamento de usuários reais ao interagir com o app em situações comuns, a fim de identificar pontos de fricção e oportunidades de melhoria.

1. IMPLANTAÇÃO

O código-fonte do aplicativo Capitech Mobile está disponível publicamente no GitHub, contendo todo o histórico de desenvolvimento e instruções de execução:

* 📘 **Readme**: <https://github.com/CapitechDev/capitech-mobile/blob/main/README.md>
* 💻 **Repositório principal**: <https://github.com/CapitechDev/capitech-mobile>
* 🌐 **Versão hospedada (frontend web anterior)**: <https://capitech-front-final.vercel.app/>

Atualmente, o aplicativo está em fase de desenvolvimento contínuo. Futuramente, caso atinja maturidade para produção e uso em larga escala, será publicado nas lojas (Google Play e App Store) e ampliado com novas funcionalidades.

1. REFERÊNCIAS

¹conformedisponívelem: <<http://pmkb.com.br/sig/padroes-frameworks/pmbok-pmi/>>. Acessoem: 10 jul. 2020.

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

PRESSMAN, Roger; MAXIM, Bruce. Engenharia de Software. 8 ed. São Paulo: Mc Graw 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.