

CENTRO PAULA SOUZA
FACULDADE DE TECNOLOGIA DE FRANCA
“Dr. THOMAZ NOVELINO”

TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

FÁBIO ASSUNÇÃO VITORINO

**“PROTOTIPAGEM DE SIMULADOR DE CONCURSOS PÚBLICOS
COM FOCO EM TECNOLOGIA DA INFORMAÇÃO”**

Trabalho de Graduação apresentado à Faculdade de Tecnologia de Franca - “Dr. Thomaz Novelino”, como parte dos requisitos obrigatórios para obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Orientador: Prof. Me. Carlos Alberto Lucas

FRANCA/SP

2025

“PROTOTIPAGEM DE SIMULADOR DE CONCURSOS PÚBLICOS COM FOCO EM TECNOLOGIA DA INFORMAÇÃO”

Fábio Assunção Vitorino

Resumo

Este trabalho apresenta a prototipagem de telas para um aplicativo móvel chamado Simulador de Concursos, voltado para auxiliar estudantes na preparação para concursos públicos com foco em Tecnologia da Informação. As telas permitem simular a realização de simulados personalizados, com escolha do número de questões e tempo limite, além de incluir interfaces para gerenciamento de questões, visualização de relatórios de desempenho e autenticação de usuários. A interface minimalista, com paleta de cores em branco e azul royal, prioriza usabilidade e acessibilidade, enquanto o design suporta um banco de dados local com até 100 questões, visando praticidade. A prototipagem foi desenvolvida com HTML, CSS e JavaScript, estruturada com base em requisitos funcionais, não funcionais e regras de negócio, sem a implementação do sistema completo. O objetivo é oferecer uma representação visual e interativa que facilite o estudo em momentos livres, com navegação intuitiva e relatórios detalhados para promover o autoconhecimento. As telas atendem às necessidades de estudantes com diferentes níveis de familiaridade tecnológica, garantindo conformidade com padrões de design.

Palavras-chave: Aplicativo móvel, Concursos públicos, Prototipagem, Simulados, Tecnologia da Informação.

Abstract

This work focuses on the prototyping of mobile application screens for the Simulador de Concursos, ed to support students preparing for public service exams in Information Technology. The prototyped screens enable users to simulate customized tests by selecting the number of questions and time limits, manage question banks, view performance reports, and authenticate securely. Featuring a minimalist interface with a white and royal blue color scheme, the prioritizes usability and accessibility, catering to users with varying technological familiarity. The prototypes, developed using HTML, CSS, and JavaScript, are based on functional and non-functional requirements and business rules, simulating a local database of up to 100 questions for efficiency. The goal is to provide an interactive visual representation that optimizes study during spare moments, offering intuitive navigation and detailed reports to enhance self-awareness. These screens ensure compliance with standards, facilitating validation within the academic scope of the project.

Keywords: Application screens, Information technology, Prototyping, Public servisse exams, Simulated tests.

SUMÁRIO

• 1. INTRODUÇÃO	4
1.1 Termo de Abertura do Projeto (TAP)	6
1.2 Estrutura Analítica do Projeto (EAP)	10
• 2. FUNDAMENTAÇÃO TEÓRICA	13
2.1 Tecnologia Educacional	13
2.2 Atores Envolvidos	14
2.3 Processos Operacionais	14
2.4 Tecnologias de Prototipagem	15
2.5 Ferramentas de Implementação	15
• 3. VIABILIDADE DO PROJETO	16
3.1 Canvas de Modelo de Negócio	18
3.2 Análise SWOT	19
3.3 Plano de Ação 5W2H	21
• 4. LEVANTAMENTO DE REQUISITOS	24
4.1 Ferramentas Utilizadas	25
4.2 Justificativa das Ferramentas	27
4.3 Métodos de Execução	28
4.4 Elicitação de Requisitos	30
4.5 BPMN	32
4.6 Requisitos Funcionais	33
4.7 Requisitos Não Funcionais	37
4.8 Regras de Negócio	41
4.9 Casos de Uso	45
4.10 Diagrama de Classes	50
• 5. DESENVOLVIMENTO DO PROTÓTIPO	52
5.1 Conceitos e Importância de UX e IHC	52
5.2 Prototipagem: Tipos e Benefícios	52
5.3 Processo de Desenvolvimento do Protótipo	53
• 6. RESULTADOS E DISCUSSÃO	55
6.1 Capturas das Telas e Funcionalidades	56
6.2 Documento de Análise de Portabilidade	60
6.3 Discussão dos Resultados	61
• 7. CONSIDERAÇÕES FINAIS	62
REFERÊNCIAS	64
ANEXOS	65
Anexo 1: Formulário Entrevistas	65
Anexo 2: Questões do Formulário Google Forms	69
Anexo 3: Códigos html e css das Telas Prototipadas	70

1 Introdução

A crescente demanda por ferramentas de preparação para concursos públicos, especialmente na área de Tecnologia da Informação, evidencia a necessidade de soluções de estudo acessíveis e eficientes. O Simulador de Concursos propõe a prototipagem de telas de um aplicativo móvel para auxiliar estudantes na prática para esses exames. O tema centra-se em tecnologia educacional, especificamente plataformas de aprendizado móvel que permitem estudo flexível e autogerenciado. A questão problema é a falta de tempo e ferramentas práticas para estudantes se prepararem de forma eficaz, frequentemente limitados por métodos tradicionais de estudo que consomem tempo ou não são acessíveis durante a rotina diária. A hipótese sugere que uma interface móvel minimalista e amigável, com funcionalidades como simulados personalizáveis, gerenciamento de questões e relatórios de desempenho, pode otimizar a eficiência e o engajamento no estudo. O projeto envolve estudantes como atores principais, interagindo com processos como realização de simulados, visualização de dados e autenticação de usuários, utilizando HTML, CSS e JavaScript para criar protótipos de telas responsivas e interativas.

A importância do problema é multifacetada. Tecnicamente, o projeto aborda o desafio de projetar interfaces intuitivas que funcionem eficientemente em dispositivos móveis, garantindo compatibilidade com resoluções a partir de 720x1280 pixels. Academicamente, contribui para a formação em engenharia de software, aplicando conceitos de análise de requisitos, prototipagem e usabilidade em um escopo delimitado. Socioeconomicamente, apoia estudantes de diferentes contextos, incluindo aqueles com acesso limitado a cursinhos preparatórios caros, oferecendo uma ferramenta gratuita e acessível, aumentando sua competitividade no mercado de trabalho público. Pesquisas sobre trabalhos correlatos indicam que plataformas existentes de preparação para concursos, como aplicativos móveis de questões, muitas vezes carecem de personalização ou foco em Tecnologia da Informação, reforçando a necessidade de soluções específicas. Essas descobertas destacam a relevância de um esforço direcionado à prototipagem.

O objetivo geral é entregar um conjunto de telas prototipadas para o aplicativo Simulador de Concursos, proporcionando uma representação visual e interativa de uma ferramenta de estudo para concursos de Tecnologia da Informação. Os objetivos específicos incluem: (1) definir requisitos funcionais e não funcionais para orientar o design das telas; (2) criar *Wire frames* e protótipos com HTML, CSS e *Javascript* para

simular interações do usuário; (3) modelar processos com diagramas (ex.: Caso de Uso, BPMN) para documentar os fluxos das telas; (4) validar a usabilidade do protótipo por meio de ajustes iterativos no design; e (5) documentar o processo de prototipagem conforme padrões de engenharia de software, como o Termo de Abertura do Projeto (TAP) e a Estrutura Analítica do Projeto (EAP). Essas atividades alinham-se às fases de Análise e Design do Ciclo de Vida do Desenvolvimento de Software (CVDS), com ênfase em elicitación de requisitos, design de interface e validação, sem implementação completa do sistema.

A relevância do projeto reside em seu potencial para capacitar estudantes, oferecendo uma ferramenta de estudo prática e baseada em dispositivos móveis, que aborda restrições de tempo e melhora a eficiência da preparação. Ao focar em concursos de Tecnologia da Informação, o projeto preenche uma lacuna específica, fornecendo uma solução personalizada que apoia o avanço profissional em um campo competitivo. A abordagem de prototipagem garante um entregável de baixo custo e alta fidelidade, adequado para avaliação acadêmica e possíveis desenvolvimentos futuros.

As metodologias empregadas incluem princípios Ágeis, com ciclos iterativos de prototipagem, e práticas de engenharia de software do curso, como especificação de requisitos (TAP), estruturação de tarefas (EAP) e modelagem de processos (BPMN). Ferramentas como *Figma* para iniciais, *Visual Studio Code* para codificação e *Git* para controle de versão suportam o desenvolvimento dos protótipos em *HTML/CSS/JavaScript*. Testes de usabilidade, embora simulados, informam refinamentos no design para atender às necessidades dos usuários.

O documento está estruturado em capítulos para apresentar o projeto de forma abrangente:

- **Capítulo 1 – Introdução:** Contextualiza o tema, problema, hipótese, objetivos, justificativas e metodologias, conforme detalhado aqui.
- **Capítulo 2 – Referencial Teórico:** Aborda conceitos de tecnologia educacional, aprendizado móvel e prototipagem, fundamentando o projeto.
- **Capítulo 3 – Desenvolvimento do Projeto:** Detalha o processo de prototipagem, incluindo requisitos (funcionais, não funcionais, regras de negócio), design das telas e modelos de processos.

- **Capítulo 4 – Resultados e Discussão:** Avalia a usabilidade do protótipo, conformidade com requisitos e impacto potencial, com demonstrações dos *mockups*.
- **Capítulo 5 – Conclusão:** Resume os achados, limitações e recomendações para trabalhos futuros, refletindo sobre as contribuições acadêmicas e práticas do projeto.

Essa estrutura garante um fluxo claro e lógico, atendendo às expectativas dos avaliadores quanto ao contexto, definição do problema, objetivos, justificativa, contribuição para a comunidade e organização do documento.

1.1 Termo da Abertura do Projeto (TAP)

O Termo de Abertura do Projeto (TAP) é um documento formal que autoriza oficialmente o início de um projeto, definindo seu propósito, objetivos, escopo, partes interessadas, premissas, restrições e principais entregáveis. Segundo o *PMBOK Guide (PMI, 2017)*, o TAP é essencial para alinhar expectativas entre as partes envolvidas, fornecer uma visão clara do projeto e servir como referência para decisões futuras. Ele estabelece a justificativa do projeto, documenta os requisitos iniciais e formaliza o comprometimento organizacional, sendo um marco inicial no gerenciamento de projetos. No contexto acadêmico, como em um Trabalho de Graduação (TG), o TAP é crucial para delimitar o escopo, justificar a relevância e orientar o desenvolvimento, garantindo que o projeto atenda aos objetivos educacionais e às expectativas dos avaliadores.

No projeto **Simulador de Concursos**, o TAP formaliza a prototipagem de telas de um aplicativo móvel voltado para estudantes que se preparam para concursos públicos na área de Tecnologia da Informação. Abaixo, é apresentado o TAP, estruturado em uma tabela (Tabela 1), que detalha os elementos essenciais do projeto, conforme as diretrizes do PMBOK e as práticas de engenharia de software do curso.

Tabela 1: Termo de Abertura do Projeto – Simulador de Concursos

Item	Descrição
Nome do Projeto	Simulador de Concursos
Gerente do Projeto	Fábio Assunção Vitorino

Data de Abertura	12 de maio de 2025
Justificativa	A falta de ferramentas práticas e acessíveis para estudantes prepararem-se para concursos públicos em Tecnologia da Informação, combinada com restrições de tempo, demanda uma solução móvel que otimize o estudo. A prototipagem de telas oferece uma representação visual e interativa, facilitando a validação acadêmica e atendendo às necessidades de estudantes com diferentes níveis de familiaridade tecnológica.
Objetivo Geral	Prototipar telas de um aplicativo móvel que auxilie estudantes na preparação para concursos públicos em Tecnologia da Informação, com funcionalidades como simulados personalizáveis, gerenciamento de questões e relatórios de desempenho.
Objetivos Específicos	1. Definir requisitos funcionais e não funcionais para orientar o design das telas. 2. Desenvolver protótipos interativos usando <i>HTML</i>, <i>CSS</i> e <i>JavaScript</i>. 3. Modelar processos com diagramas (ex.: Caso de Uso, <i>BPMN</i>) para documentar fluxos. 4. Validar a usabilidade do protótipo por meio de ajustes iterativos. 5. Documentar o processo de prototipagem conforme padrões de engenharia de software (TAP, EAP).
Escopo	Inclui: Prototipagem de telas para login, cadastro, tela inicial, configuração de simulados, realização de simulados, resultados, relatórios/histórico com dashboard de gráfico e gerenciamento de questões, utilizando <i>HTML</i> , <i>CSS</i> e <i>JavaScript</i> . Design

	minimalista com paleta branco e azul royal, compatível com resoluções a partir de 720x1280px, simulando um banco de até 100 questões. Não inclui: Desenvolvimento completo do aplicativo, implementação de <i>backend</i> (ex.: <i>SQLite</i>), integração com banco de dados real ou <i>deploy</i> em dispositivos móveis.
Entregáveis	1. Conjunto de telas prototipadas (arquivos <i>HTML/CSS/JavaScript</i>). 2. Documentação do projeto (requisitos, regras de negócio, diagramas). 3. Relatório final com análise de usabilidade e resultados da prototipagem.
Partes Interessadas	- Gerente do Projeto: Fábio Assunção Vitorino, responsável pela execução. - Orientador: Carlos Alberto Lucas, responsável por supervisão acadêmica. - Avaliadores: Banca examinadora da FATEC, responsável por validação. - Usuários-alvo: Estudantes de Tecnologia da Informação (indiretamente beneficiados).
Premissas	1. O gerente do projeto possui conhecimentos básicos em <i>HTML</i>, <i>CSS</i> e <i>JavaScript</i>. 2. Ferramentas como <i>Visual Studio Code</i> e <i>Figma</i> estão disponíveis. 3. O prazo de entrega (junho de 2025) é suficiente para a prototipagem.

		4. Estudantes-alvo têm acesso a dispositivos móveis com resolução mínima de 720x1280px.
Restrições		1. Prazo limitado até junho de 2025. 2. Desenvolvimento individual, sem equipe de apoio. 3. Escopo restrito à prototipagem, sem implementação funcional completa. 4. Banco de questões limitado a 100, <i>mockado</i> nos protótipos.
Riscos Iniciais		1. Atrasos devido à curva de aprendizado em ferramentas de prototipagem. 2. Dificuldade em garantir usabilidade para todos os níveis de usuários. 3. Limitações técnicas em simular interatividade avançada sem <i>backend</i>. Mitigações: Planejamento detalhado (EAP), testes iterativos de usabilidade, uso de <i>mockups</i> estáticos para funcionalidades complexas.
Orçamento	Estimado	Sem custos financeiros diretos, pois utiliza ferramentas gratuitas (<i>Figma</i> , <i>VS Code</i> , <i>Git</i>) e recursos acadêmicos. Custo indireto: tempo do gerente (estimado em 200 horas).
Cronograma Inicial		- Maio 2025: Definição de requisitos e . - Junho 2025: Prototipagem, testes de usabilidade, documentação e entrega final.
Autorização		Fábio Assunção Vitorino, Gerente do Projeto Carlos Alberto Lucas, Orientador Acadêmico

1.2 Estrutura Analítica do Projeto (EAP)

A Estrutura Analítica do Projeto (EAP) é uma ferramenta de gerenciamento de projetos que decompõe o escopo total do projeto em partes menores e gerenciáveis, organizadas hierarquicamente. De acordo com o *PMBOK Guide (PMI, 2017)*, a EAP é essencial para definir claramente todas as entregas e atividades necessárias, facilitando o planejamento, a execução e o controle do projeto. Ela permite identificar os pacotes de trabalho, alocar recursos de forma eficiente e monitorar o progresso, garantindo que nenhum aspecto do escopo seja negligenciado. No contexto de um Trabalho de Graduação (TG), a EAP é crucial para estruturar as tarefas de forma lógica, alinhando-as aos objetivos do projeto e respeitando restrições como prazo e recursos limitados. No projeto **Simulador de Concursos**, a EAP organiza as atividades de prototipagem de telas do aplicativo móvel, detalhando as fases de planejamento, design, desenvolvimento, validação e documentação, conforme os requisitos e objetivos definidos no Termo de Abertura do Projeto (TAP).

A EAP do **Simulador de Concursos**, apresentada na Tabela 2, é estruturada em níveis hierárquicos que refletem o ciclo de vida do projeto. O nível 1 representa o projeto como um todo, enquanto o nível 2 divide o trabalho em cinco macrofases: Planejamento, Design de Interface, Prototipagem, Validação e Documentação. Cada macrofase é desdobrada em pacotes de trabalho (nível 3) e, quando necessário, em atividades específicas (nível 4), garantindo granularidade suficiente para o gerenciamento individual do projeto, considerando o prazo até junho de 2025 e o desenvolvimento solo. A estrutura foi elaborada para cobrir os objetivos específicos do TAP, como definição de requisitos, criação de protótipos em *HTML/CSS/JavaScript*, modelagem de processos e validação de usabilidade, mantendo o foco na prototipagem sem implementação completa.

Tabela 2: Estrutura Analítica do Projeto – Simulador de Concursos

Nível	Código	Descrição	Entregáveis
1	1.0	Simulador de Concursos	Conjunto de telas prototipadas e documentação completa
2	1.1	Planejamento	Documentos de escopo e requisitos

3	1.1.1	Definição de Requisitos	Lista de requisitos funcionais e não funcionais
4	1.1.1.1	Elicitação de Requisitos	Questionário, análise SWOT, 5W2H
4	1.1.1.2	Documentação de Requisitos	Especificação de RF001–RF005, RNF001–RNF006, RN001–RN022
3	1.1.2	Elaboração do TAP	Termo de Abertura do Projeto
3	1.1.3	Planejamento de Tarefas	EAP, cronograma inicial
2	1.2	Design de Interface	<i>Wireframes</i> e especificações de design
3	1.2.1	Criação de <i>Wireframes</i>	<i>Wireframes</i> das telas (login, home, simulado, etc.)
3	1.2.2	Definição de Estilo Visual	Paleta de cores (branco, azul royal), tipografia (Roboto)
3	1.2.3	Modelagem de Processos	Diagramas de Caso de Uso e BPMN
2	1.3	Prototipagem	Telas interativas em <i>HTML/CSS/JavaScript</i>
3	1.3.1	Desenvolvimento de Telas	Arquivos <i>HTML/CSS</i> para 8 telas (login, cadastro, home, etc.)
4	1.3.1.1	Codificação <i>HTML/CSS</i>	Estrutura e estilos das telas
4	1.3.1.2	Interatividade com <i>JavaScript</i>	Scripts para login, recuperação de senha, navegação
3	1.3.2	Simulação de Dados	<i>Mockup</i> de banco com 100 questões
2	1.4	Validação	Relatório de usabilidade
3	1.4.1	Testes de Usabilidade	Feedback simulado de navegação e interface

3	1.4.2	Ajustes no Protótipo	Refinamentos com base em testes
2	1.5	Documentação	Relatório final do TG
3	1.5.1	Redação dos Capítulos	Introdução, referencial teórico, desenvolvimento, resultados, conclusão
3	1.5.2	Consolidação de Diagramas	Inclusão de diagramas na documentação
3	1.5.3	Revisão Final	Correção e formatação do documento

Comentário sobre a Estrutura da EAP

A EAP apresentada na Tabela 2 foi elaborada com base nas diretrizes do *PMBOK Guide* (PMI, 2017) e adaptada ao contexto acadêmico do TG, considerando o escopo de prototipagem do Simulador de Concursos. A estrutura é dividida em cinco macrofases (Planejamento, *Design* de Interface, Prototipagem, Validação e Documentação), que abrangem todas as atividades necessárias para alcançar os objetivos do projeto. O Planejamento inclui a definição de requisitos e a elaboração do TAP, garantindo uma base sólida. O Design de Interface foca na criação de *wireframes* e na modelagem de processos, alinhando-se aos requisitos de usabilidade (RN015–RN016). A Prototipagem concentra-se no desenvolvimento das telas em *HTML/CSS/JavaScript*, com interatividade básica (ex.: login, recuperação de senha), e opção de login com redes sociais, conforme RN001–RN014. A Validação assegura que o protótipo atenda às expectativas de usabilidade, enquanto a Documentação consolida os resultados para avaliação acadêmica. A granularidade até o nível 4 em algumas atividades (ex.: codificação HTML/CSS, scripts JavaScript) facilita o gerenciamento individual, respeitando as restrições de prazo (junho de 2025) e desenvolvimento solo.

2 Fundamentação Teórica

A fundamentação teórica deste Trabalho de Graduação (TG) sobre o Simulador de Concursos apresenta os conceitos, fundamentos e características relacionados à área de negócio, atores, processos, tecnologias e ferramentas aplicáveis à prototipagem de telas de um aplicativo móvel voltado para preparação de concursos públicos na área de Tecnologia da Informação. O objetivo é contextualizar a questão problema – a falta de ferramentas práticas e acessíveis para estudantes otimizar seus estudos – e sustentar a hipótese de solução, que propõe telas minimalistas e interativas para simulados personalizáveis. Organizada em subseções, esta seção aborda a tecnologia educacional como área de negócio, os atores envolvidos, os processos passíveis de automação, as tecnologias de prototipagem e as ferramentas utilizadas, alinhando-se aos requisitos e ao escopo definidos no Termo de Abertura do Projeto (TAP) e na Estrutura Analítica do Projeto (EAP).

2.1 Tecnologia Educacional

A tecnologia educacional refere-se à aplicação de recursos tecnológicos para facilitar o aprendizado, promovendo acesso, eficiência e personalização no ensino. Segundo Moore e Kearsley (2012), ela abrange plataformas digitais, como aplicativos móveis, que transformam a experiência educacional ao permitir estudo autogerenciado e flexível. No contexto de concursos públicos, a tecnologia educacional atua como um serviço que capacita candidatos por meio de ferramentas de prática e avaliação, como simulados online. Este projeto se insere no setor de serviços educacionais, atendendo à demanda por preparação acessível para concursos de Tecnologia da Informação, um campo com alta competitividade devido às oportunidades no setor público. A relevância do tema está na capacidade de atender estudantes com restrições de tempo e recursos, oferecendo uma solução que se adapta a rotinas dinâmicas e promove aprendizado contínuo.

2.1.1 Concursos Públicos em Tecnologia da Informação

Concursos públicos na área de Tecnologia da Informação exigem conhecimentos em programação, banco de dados, redes e segurança, entre outros. A preparação envolve a prática de questões objetivas sob condições que simulam o exame real, como tempo limitado e diversidade de tópicos. Segundo dados do mercado educacional brasileiro, a procura por concursos aumentou na última década,

impulsionada por estabilidade profissional e salários atrativos no setor público. Ferramentas de tecnologia educacional, como aplicativos de simulados, são cruciais para organizar o estudo, permitindo que candidatos testem conhecimentos e monitorem progresso. O Simulador de Concursos foca nesse nicho, prototipando telas que simulam a experiência de estudo, com funcionalidades como personalização de simulados e relatórios de desempenho.

2.2 Atores Envolvidos

Os atores no contexto do Simulador de Concursos incluem os estudantes como usuários finais, que buscam uma ferramenta prática para preparação. Esses usuários variam em nível de familiaridade tecnológica, desde iniciantes até profissionais experientes em Tecnologia da Informação, exigindo uma interface intuitiva e acessível. Outros atores indiretos incluem instituições educacionais, que podem adotar a ferramenta como complemento ao ensino, e organizadoras de concursos, que definem os padrões das provas. Não há cadeia produtiva complexa, pois o projeto foca na prototipagem, mas, em um cenário de implementação, fornecedores de conteúdo (ex.: autores de questões) e desenvolvedores de software seriam relevantes. A interação dos atores com o sistema ocorre por meio de processos como autenticação, realização de simulados e consulta de relatórios, todos representados nas telas prototipadas.

2.3 Processos Operacionais

Os processos operacionais passíveis de automação no Simulador de Concursos são aqueles que estruturam a experiência do usuário. Conforme Sommerville (2016), a automação de processos em sistemas educacionais envolve a execução de tarefas repetitivas e a organização de dados para melhorar a eficiência. Os principais processos identificados incluem:

- **Autenticação de Usuários:** Permite login e cadastro, garantindo acesso seguro (RN001–RN003).
- **Gerenciamento de Questões:** Envolve criar, editar e excluir questões, com validação de uma única resposta correta (RN004–RN006).
- **Realização de Simulados:** Inclui configuração do número de questões e tempo, exibição de questões com feedback visual e cronômetro (RN007–RN008, RN014).

- Geração de Relatórios: Apresenta desempenho com acertos, erros e tempo, salvos automaticamente (RN009–RN012). Esses processos foram mapeados nos requisitos funcionais (RF001–RF005) e são simulados nas telas prototipadas, utilizando interações mockadas para representar a automação.

2.4 Tecnologias de Prototipagem

As tecnologias empregadas na prototipagem do Simulador de Concursos são selecionadas para atender aos requisitos de usabilidade e compatibilidade (RNF001–RNF006). Pressman (2015) destaca que a escolha de tecnologias em projetos de software deve considerar escalabilidade, acessibilidade e facilidade de implementação. As principais tecnologias incluem:

- Arquitetura Web: As telas são desenvolvidas com HTML5 para estrutura, CSS3 para estilo e JavaScript para interatividade, formando uma arquitetura cliente-side adequada para prototipagem rápida.
- Simulação de Banco de Dados: Um banco de dados local com até 100 questões é mockado em JavaScript, simulando o comportamento de um SGBD como SQLite, sem implementação real (RN005).
- Compatibilidade Móvel: A interface é responsiva, otimizada para resoluções a partir de 720x1280px, utilizando media queries em CSS (RN019).

2.5 Ferramentas de Implementação

As ferramentas escolhidas para a prototipagem são amplamente utilizadas em desenvolvimento web e design de interfaces. Segundo Sommerville (2016), ferramentas de prototipagem devem ser acessíveis e alinhadas ao nível de habilidade do desenvolvedor. As principais ferramentas incluem:

- Figma: Utilizado para criar wireframes iniciais, definindo o layout e a paleta de cores (branco #FFFFFF, azul royal #0052CC, RN015–RN016).
- HTML, CSS e JavaScript: HTML5 estrutura as telas, CSS3 aplica estilos minimalistas (ex.: fonte Roboto, sombreado em cinza #D3D3D3 e verde #008000, RN014), e JavaScript adiciona interatividade básica, como validação de login e recuperação de senha.
- Visual Studio Code: Editor de código para desenvolvimento e depuração, com extensões como Live Server para testes locais.

- Git: Controle de versão para gerenciar alterações nos arquivos de prototipagem. Essas ferramentas são gratuitas, atendendo à restrição de orçamento zero (TAP), e suportam o desenvolvimento individual até junho de 2025.

3 Viabilidade do Projeto

A análise de viabilidade do projeto Simulador de Concursos é essencial para garantir que a prototipagem de telas do aplicativo móvel seja realizável dentro das restrições de prazo (junho de 2025), recursos e escopo definidos no Termo de Abertura do Projeto (TAP) e na Estrutura Analítica do Projeto (EAP). Para avaliar a viabilidade, utiliza-se o modelo Canvas de Proposta de Valor (Value Proposition Canvas), que mapeia como o projeto atende às necessidades dos usuários e cria valor, alinhando-se ao foco em prototipagem de telas com *HTML*, *CSS* e *JavaScript*, sem desenvolvimento completo do sistema. Este modelo, proposto por *Osterwalder et al. (2014)*, divide a análise em duas áreas: o perfil do cliente (necessidades e dores dos usuários) e a proposta de valor (soluções oferecidas pelas telas prototipadas). A seguir, a viabilidade do projeto é apresentada por meio do Canvas, detalhando cada componente em relação ao contexto do Simulador de Concursos, conforme descrito na Tabela 3.

O Canvas foi escolhido por sua capacidade de alinhar o escopo do projeto às expectativas dos usuários-alvo (estudantes de Tecnologia da Informação que se preparam para concursos públicos) e por ser adequado à natureza acadêmica do TG, que prioriza a validação de conceitos e a entrega de protótipos de alta fidelidade. A Tabela 3 estrutura a análise, situando o projeto em cada área do Canvas, com foco na prototipagem de telas que simulam funcionalidades como login, simulados personalizáveis, relatórios de desempenho e gerenciamento de questões.

Tabela 3: Canvas de Proposta de Valor – Simulador de Concursos

Área do Canvas	Descrição
Tarefas do Cliente	Estudantes precisam estudar para concursos públicos de Tecnologia da Informação, praticando questões objetivas sob condições que

	simulam o exame real (ex.: tempo limitado). Eles buscam organizar o estudo, monitorar desempenho e acessar materiais em momentos livres, usando dispositivos móveis.
Dores do Cliente	- Falta de tempo para estudar devido a rotinas corridas. - Acesso limitado a ferramentas gratuitas e especializadas em Tecnologia da Informação. - Dificuldade em encontrar simulados personalizáveis que atendam a diferentes níveis de preparo. - Interfaces complexas ou pouco intuitivas em aplicativos existentes, dificultando o uso por iniciantes.
Ganhos do Cliente	- Estudo eficiente em curtos períodos, com acesso rápido a simulados. - Feedback imediato sobre desempenho, com relatórios claros. - Interface amigável que reduz a curva de aprendizado. - Preparação focada em Tecnologia da Informação, aumentando a confiança para o exame.
Produtos e Serviços	Protótipos de telas de um aplicativo móvel, desenvolvidos em HTML, CSS e JavaScript, incluindo: login/cadastro, tela inicial, configuração de simulados, realização de simulados, resultados, relatórios/histórico e gerenciamento de questões. As telas simulam um banco de até 100 questões e são responsivas (resoluções a partir de 720x1280px).
Aliviadores de Dores	- Telas minimalistas com paleta branco e azul royal, fonte Roboto e feedback visual (ex.: sombreamento cinza #D3D3D3, verde #008000 para respostas corretas), facilitando o uso (RN015–RN016). - Simulados personalizáveis (número de questões, tempo limite), otimizando o estudo (RN007–RN008). - Mockups de relatórios com acertos, erros e tempo, acessíveis em poucos cliques (RN009–RN012). - Prototipagem em HTML/CSS/JavaScript, permitindo testes rápidos sem necessidade de backend.
Criadores de Ganhos	- Navegação intuitiva, com botões claros (ex.: “Novo Simulado”, “Resultados”) e telas com poucas informações, acelerando o acesso às funcionalidades (RN013–RN014).

	- Simulação de banco de questões focado em Tecnologia da Informação, alinhado às demandas do público-alvo (RN005). - Protótipos responsivos, garantindo compatibilidade com dispositivos móveis e usabilidade em diferentes contextos (RN019). - Relatórios detalhados (mockados) que promovem autoconhecimento, incentivando a melhoria contínua.
--	--

3.1 Canvas de Negócio (*Business Model Canvas* - BMC)

O *Business Model Canvas* (BMC) é uma ferramenta estratégica que permite visualizar e estruturar o modelo de negócio de um projeto ou produto em um único diagrama, dividido em nove blocos interconectados. Desenvolvido por *Osterwalder e Pigneur (2010)*, o BMC é amplamente utilizado para mapear como uma organização cria, entrega e captura valor, sendo particularmente útil em projetos inovadores ou acadêmicos, como o Trabalho de Graduação (TG). Segundo os autores, sua importância reside na capacidade de alinhar objetivos, identificar oportunidades e planejar a viabilidade de forma clara e colaborativa, facilitando a comunicação entre stakeholders. No contexto do Simulador de Concursos, o BMC é aplicado para analisar a viabilidade da prototipagem de telas de um aplicativo móvel voltado para estudantes que se preparam para concursos públicos em Tecnologia da Informação, definindo como o projeto atende às necessidades dos usuários dentro do escopo acadêmico, sem implementação comercial.

A seguir, o BMC do Simulador de Concursos é apresentado na Tabela 4, estruturado para refletir o foco na prototipagem com HTML, CSS e JavaScript, conforme delimitado no Termo de Abertura do Projeto (TAP) e na Estrutura Analítica do Projeto (EAP). Cada bloco do Canvas é comentado na sequência, detalhando sua aplicação ao projeto e conectando-o aos requisitos funcionais (RF001–RF005), não funcionais (RNF001–RNF006) e regras de negócio (RN001–RN022).

Tabela 4: Business Model Canvas – Simulador de Concursos

Bloco	Descrição
Segmento de Clientes	Estudantes que se preparam para concursos públicos em Tecnologia da Informação, incluindo iniciantes e profissionais

	experientes, com idades entre 18 e 40 anos, que buscam ferramentas práticas para estudar em momentos livres.
Proposta de Valor	Telas prototipadas de um aplicativo móvel que oferecem simulados personalizáveis, gerenciamento de questões, relatórios de desempenho e autenticação segura, com interface minimalista (branco e azul royal, fonte Roboto) para estudo eficiente e acessível.
Canais	Distribuição acadêmica via apresentação do TG na FATEC, com acesso às telas prototipadas por meio de arquivos HTML/CSS/JavaScript hospedados localmente ou em repositórios Git.
Relacionamento	Relacionamento indireto com usuários-alvo por meio de validação acadêmica; feedback simulado de usabilidade durante a prototipagem, sem interação direta com estudantes.
Fontes de Receita	Não aplicável, pois o projeto é acadêmico, sem fins comerciais. O valor está na validação do protótipo para aprovação do TG.
Recursos Chave	Conhecimento em HTML, CSS e JavaScript do gerente do projeto; ferramentas gratuitas (Figma, Visual Studio Code, Git); infraestrutura acadêmica da FATEC (computadores, orientador).
Atividades Chave	Definição de requisitos, criação de wireframes, prototipagem de telas, modelagem de processos (ex.: diagramas de Caso de Uso, BPMN), validação de usabilidade e documentação do projeto.
Parcerias Chave	Orientador acadêmico para supervisão; FATEC para infraestrutura e validação; comunidade open-source para ferramentas gratuitas (ex.: bibliotecas JavaScript).
Estrutura de Custos	Custos mínimos, limitados ao tempo do gerente (estimado em 200 horas) e uso de recursos acadêmicos gratuitos; sem custos financeiros diretos.

3.2 Matriz SWOT

A Matriz SWOT é uma ferramenta de análise estratégica que avalia os fatores internos e externos que impactam um projeto, organizando-os em quatro categorias: Forças (Strengths), Fraquezas (Weaknesses), Oportunidades (Opportunities) e

Ameaças (Threats). Segundo Hill e Westbrook (1997), a SWOT é essencial para identificar vantagens competitivas, limitações, possibilidades de crescimento e riscos, permitindo um planejamento mais informado. No contexto de um Trabalho de Graduação (TG), a matriz auxilia na avaliação da viabilidade do projeto, destacando como os recursos disponíveis e o ambiente acadêmico influenciam o sucesso. Para o Simulador de Concursos, a SWOT analisa a prototipagem de telas de um aplicativo móvel voltado para estudantes que se preparam para concursos públicos em Tecnologia da Informação, alinhando-se ao escopo definido no Termo de Abertura do Projeto (TAP) e na Estrutura Analítica do Projeto (EAP).

A Matriz SWOT do Simulador de Concursos é apresentada na Tabela 5, estruturada para refletir o foco na prototipagem com HTML, CSS e JavaScript, sem desenvolvimento completo do sistema. A análise considera os requisitos funcionais (RF001–RF005), não funcionais (RNF001–RNF006) e regras de negócio (RN001–RN022), além do contexto acadêmico e das restrições do projeto.

Tabela 5: Matriz SWOT – Simulador de Concursos

Categoria	Descrição
Forças (Strengths)	- Uso de tecnologias acessíveis (HTML, CSS, JavaScript) com baixa curva de aprendizado. - Interface minimalista (branco #FFFFFF, azul royal #0052CC, fonte Roboto) que atende aos padrões de usabilidade (RN015–RN016). - Escopo bem definido no TAP, focado em 8 telas prototipadas (login, cadastro, simulado, etc.). - Ferramentas gratuitas (Figma, Visual Studio Code, Git) eliminam custos financeiros. - Alinhamento com a demanda por ferramentas de estudo para concursos de Tecnologia da Informação.
Fraquezas (Weaknesses)	- Desenvolvimento individual, limitando a velocidade e a diversidade de ideias. - Ausência de backend real (ex.: SQLite), restringindo a interatividade a mockups.

	- Curva de aprendizado inicial em JavaScript para funcionalidades interativas (ex.: validação de login, recuperação de senha). - Prazo restrito até junho de 2025, com risco de atrasos em ajustes iterativos.
Oportunidades (Opportunities)	- Crescimento da demanda por aplicativos educacionais para concursos públicos. - Possibilidade de validação acadêmica robusta, com feedback de orientadores e avaliadores. - Potencial para expansão futura do protótipo em um aplicativo funcional com Flutter/Dart. - Interesse do público por ferramentas gratuitas e especializadas em Tecnologia da Informação.
Ameaças (Threats)	- Concorrência de aplicativos comerciais já consolidados no mercado educacional. - Limitações técnicas em dispositivos móveis de baixa performance, afetando a responsividade. - Possíveis mudanças nos requisitos acadêmicos, exigindo ajustes no escopo. - Dependência de ferramentas open-source, sujeitas a atualizações ou falhas.

3.3 Plano de Ação 5W2H do Projeto

A ferramenta 5W2H é um método de planejamento que organiza ações respondendo a sete perguntas fundamentais: What (O que será feito?), Why (Por que será feito?), Where (Onde será feito?), When (Quando será feito?), Who (Quem fará?), How (Como será feito?) e How Much (Quanto custará?). Segundo Chiavenato (2004), o 5W2H é amplamente utilizado em gestão de projetos por sua simplicidade e eficácia em detalhar atividades, garantindo clareza nos objetivos, responsabilidades e recursos necessários. Sua importância reside na capacidade de reduzir ambiguidades, alinhar expectativas entre as partes interessadas e facilitar o monitoramento do progresso, especialmente em projetos com prazos e recursos limitados. No contexto do Trabalho de Graduação (TG) Simulador de Concursos, o

5W2H estrutura as ações para a prototipagem de telas de um aplicativo móvel voltado para estudantes que se preparam para concursos públicos em Tecnologia da Informação, complementando o Termo de Abertura do Projeto (TAP) e a Estrutura Analítica do Projeto (EAP).

O Plano de Ação 5W2H, apresentado na Tabela 6, organiza as atividades principais do projeto, mapeando cada etapa desde a definição de requisitos até a entrega da documentação final. A justificativa do plano está no alinhamento com os objetivos do TAP (Seção 1.1), que priorizam a entrega de protótipos de telas interativas em HTML, CSS e JavaScript, e na conformidade com os requisitos funcionais (RF001–RF005), não funcionais (RNF001–RNF006) e regras de negócio (RN001–RN022). O plano assegura a viabilidade do projeto dentro das restrições de prazo (junho de 2025), desenvolvimento individual e orçamento zero, utilizando ferramentas gratuitas e um cronograma realista.

Tabela 6: Plano de Ação 5W2H – Simulador de Concursos

What (O que será feito?)	Why (Por que será feito?)	Where (Onde será feito?)	When (Quando será feito?)	Who (Quem fará?)	How (Como será feito?)	How Much (Quanto custará?)
Definir requisitos	Estabelecer a base para o design das telas, garantindo conformidade com as necessidades dos usuários (RN001–RN022).	Ambiente acadêmico (computador pessoal, FATEC)	01/06/2025 a 07/06/2025	Gerente do projeto (estudante)	Elicitação via questionários, análise SWOT, documentação de RF001–RF005, RNF001–RNF006, RN001–RN022.	Sem custo financeiro; 20 horas de trabalho.

Criar wireframes	Planejar o layout das 8 telas (login, cadastro, simulado, etc.) para otimizar usabilidade (RN015–RN016).	Figma (online)	08/06/2025 a 14/06/2025	Gerente do projeto	Uso de Figma para esboçar layouts com paleta branco (#FFFFFF) e azul royal (#0052CC), fonte Roboto.	Sem custo; 15 horas de trabalho.
Prototipar telas	Desenvolver telas interativas em HTML/CSS/JavaScript para simular funcionalidades (RF001–RF005).	Visual Studio Code (computador ou pessoal)	15/06/2025 a 28/06/2025	Gerente do projeto	Codificação de 8 telas (ex.: index.html, simulado.html), com scripts JavaScript para login, recuperação de senha, navegação.	Sem custo; 80 horas de trabalho.
Modelar processos	Documentar fluxos de interação para validar o design (ex.: autenticação, simulados, RN001–RN014).	Ferramentas de diagramação (ex.: Lucidchart, Figma)	22/06/2025 a 25/06/2025	Gerente do projeto	Criação de diagramas de Caso de Uso e BPMN, mapeando processos como login e realização de simulados.	Sem custo; 10 horas de trabalho.
Validar usabilidade	Garantir que as telas sejam intuitivas e atendam às expectativas dos usuários.	Ambiente de teste (computador ou pessoal)	29/06/2025 a 30/06/2025	Gerente do projeto	Testes simulados de navegação, ajustes iterativos com base em	Sem custo; 25 horas de trabalho.

	usuários (RN013–RN019).				feedback (ex.: sombreamento cinza #D3D3D3, verde #008000, RN014).	
Documentar o projeto	Produzir o relatório final do TG, consolidando resultados entregáveis acadêmicos.	Microsoft Word ou Google Docs	01/07/2025 a 07/07/2025	Gerente do projeto	Redação de capítulos (introdução, fundamentação, desenvolvimento, resultados, conclusão), inclusão de diagramas e telas.	Sem custo; 50 horas de trabalho.

4 Levantamento de Requisitos

Esta seção descreve as ferramentas utilizadas para o desenvolvimento do protótipo de interface do Simulador de Concursos, detalhando a versão, tipo de licença e site oficial de cada uma, com justificativas para sua escolha. Além disso, apresenta os métodos de execução do projeto da interface, alinhando-se ao escopo de prototipagem de telas em HTML, CSS e JavaScript, conforme definido no Termo de Abertura do Projeto (TAP, Seção 1.1) e na Estrutura Analítica do Projeto (EAP, Seção 1.2). O objetivo é esclarecer como as ferramentas e métodos suportam a criação de telas interativas para login, simulados personalizáveis, relatórios de desempenho e gerenciamento de questões, atendendo aos requisitos funcionais (RF001–RF005), não funcionais (RNF001–RNF006) e regras de negócio (RN001–RN022).

4.1 Ferramentas Utilizadas

As ferramentas foram selecionadas com base em acessibilidade, compatibilidade com o desenvolvimento individual e ausência de custos financeiros, respeitando as restrições do TAP (orçamento zero, prazo até julho de 2025). A seguir, cada ferramenta é descrita, incluindo versão, licença, site oficial e justificativa, conforme apresentado na Tabela 7.

Tabela 7: Ferramentas Utilizadas para Prototipagem do Simulador de Concursos

Ferramenta	Versão	Tipo de Licença	Site Oficial	Descrição e Justificativa
Figma	Versão web (atualizada em junho de 2025)	Freemium (plano gratuito utilizado)	www.figma.com	Descrição: Plataforma online para design de interfaces, permitindo criar wireframes e mockups colaborativos. Suporta exportação de assets e integração com desenvolvimento web. Justificativa: Escolhido por ser gratuito no plano básico, intuitivo e amplamente utilizado para wireframing (Sommerville, 2016). Facilita a criação de layouts com a paleta de cores (branco #FFFFFF, azul royal #0052CC, RN015) e tipografia Roboto, agilizando o planejamento das 8 telas (login, cadastro, simulado, etc.).

Visual Studio Code	1.89 (maio de 2025)	Licença MIT (open-source)	code.visualstudio.com	Descrição: Editor de código leve com suporte a HTML, CSS, JavaScript, extensões (ex.: Live Server) e depuração integrada. Justificativa: Selecionado por ser gratuito, altamente personalizável e eficiente para codificação de telas (Pressman, 2015). A extensão Live Server permite testes locais em tempo real, garantindo compatibilidade com resoluções a partir de 720x1280px (RN019).
Git	2.45 (abril de 2025)	Licença GPL-2.0 (open-source)	git-scm.com	Descrição: Sistema de controle de versão para gerenciar alterações em arquivos de código. Justificativa: Escolhido por ser gratuito, robusto e padrão em desenvolvimento de software (Sommerville, 2016). Permite versionamento dos arquivos HTML, CSS e JavaScript, mitigando riscos de perda de progresso no desenvolvimento individual.
Google Chrome	125 (maio de 2025)	Licença BSD (open-source com	www.google.com/chrome	Descrição: Navegador web com ferramentas de desenvolvedor para testar

		componentes proprietários)		responsividade e depurar código. Justificativa: Selecionado por ser gratuito, amplamente utilizado e oferecer ferramentas de inspeção que validam a responsividade e o comportamento das telas (RN019), além de suportar HTML5, CSS3 e JavaScript modernos.
--	--	----------------------------	--	--

4.2 Justificativa para a Utilização das Ferramentas

As ferramentas foram escolhidas com base em critérios técnicos e práticos, alinhados às restrições do projeto:

- Figma: Sua interface amigável e plano gratuito permitem criar wireframes rapidamente, essenciais para planejar o layout das telas antes da codificação. A capacidade de definir a paleta de cores e tipografia (RN015–RN016) garante consistência visual, enquanto a exportação de assets facilita a transição para o desenvolvimento.
- Visual Studio Code: Oferece um ambiente de codificação eficiente, com suporte nativo a HTML, CSS e JavaScript, e extensões que agilizam o desenvolvimento (ex.: Live Server para visualização em tempo real). Sua leveza e personalização são ideais para o desenvolvimento individual, reduzindo a curva de aprendizado.
- Git: Garante controle de versão, permitindo reverter alterações e manter um histórico organizado dos arquivos, crucial para um projeto solo com múltiplas iterações. A integração com plataformas como GitHub (opcional) facilita a revisão por orientadores.
- Google Chrome: As ferramentas de desenvolvedor (ex.: DevTools) permitem testar a responsividade das telas em diferentes resoluções (RN019) e depurar scripts JavaScript (ex.: validação de login, RN001–RN003), assegurando conformidade com os padrões web.

Essas escolhas alinham-se à fundamentação teórica (Seção 2.5), que destaca a importância de ferramentas acessíveis e compatíveis com o nível de habilidade do desenvolvedor, e ao Canvas de Negócio (Seção 3.1), que enfatiza recursos de baixo custo.

4.3 Métodos de Execução do Projeto da Interface

Os métodos de execução do projeto da interface seguem uma abordagem iterativa, baseada em princípios Ágeis e práticas de engenharia de software, conforme descrito no PMBOK Guide (PMI, 2017). A prototipagem foi estruturada em cinco fases principais, detalhadas no Plano de Ação 5W2H (Seção 3.3), e adaptadas ao contexto acadêmico do TG. Abaixo, os métodos são descritos, com ênfase na execução da interface:

- Definição de Requisitos:
 - Método: Elicitação de requisitos por meio de análise SWOT, questionários fictícios (simulando feedback de usuários) e especificação de RF001–RF005, RNF001–RNF006 e RN001–RN022.
 - Execução: Documentação em formato textual (ex.: Google Docs), mapeando funcionalidades como login, simulados e relatórios. Isso orientou o design das telas, garantindo conformidade com as regras de negócio (ex.: RN015 para interface minimalista).
 - Resultado: Lista de requisitos que serviu como base para wireframes e protótipos.
- Design de Wireframes:
 - Método: Criação de wireframes de baixa e média fidelidade no Figma, utilizando a paleta de cores (branco #FFFFFF, azul royal #0052CC) e tipografia Roboto (RN016).
 - Execução: Esboço de 8 telas (login, cadastro, home, configuração de simulado, simulado, resultados, relatórios, gerenciamento de questões), com foco em usabilidade (RN013–RN015). Iterações foram feitas com base em revisões internas.
 - Resultado: Wireframes exportados como guias visuais para a codificação.
- Prototipagem das Telas:

- Método: Desenvolvimento iterativo em HTML5, CSS3 e JavaScript, seguindo o modelo de prototipagem rápida (Sommerville, 2016).
- Execução: Codificação em Visual Studio Code, com estrutura HTML para cada tela (ex.: index.html, simulado.html), estilos CSS para responsividade (media queries para 720x1280px, RN019) e scripts JavaScript para interatividade (ex.: validação de login com admin@admin.com e senha 1234, recuperação de senha, ou opção de login com redes sociais, RN001–RN003). Um mockup de 100 questões foi simulado em JavaScript (RN005).
- Resultado: 8 telas interativas, testadas localmente com Live Server, simulando a experiência do usuário.
- Modelagem de Processos:
 - Método: Elaboração de diagramas de Caso de Uso e BPMN para documentar fluxos de interação, como autenticação e realização de simulados (RN001–RN014).
 - Execução: Uso de Figma ou Lucidchart para criar diagramas, mapeando atores (estudantes) e processos (ex.: login, configuração de simulado). Iterações foram baseadas em revisões do orientador.
 - Resultado: Diagramas que complementam a documentação acadêmica e validam o fluxo das telas.
- Validação de Usabilidade:
 - Método: Testes simulados de usabilidade, com navegação nas telas e análise de feedback visual (ex.: sombreamento cinza #D3D3D3, verde #008000 para respostas corretas, RN014).
 - Execução: Testes no Google Chrome, verificando responsividade, clareza dos botões (ex.: “Novo Simulado”) e facilidade de uso. Ajustes iterativos foram feitos com base em observações internas, simulando feedback de usuários.
 - Resultado: Telas refinadas, garantindo conformidade com RN013–RN019 (usabilidade, acessibilidade).

Os métodos foram executados de forma iterativa, com revisões frequentes para alinhar o protótipo aos objetivos do TAP e às expectativas acadêmicas. O controle de versão com Git garantiu a organização das alterações, enquanto o cronograma do 5W2H (01/06/2025 a 07/07/2025) orientou a execução dentro do prazo.

4.4 Elicitação e especificação dos Requisitos

A elicitação de requisitos é o processo de identificar e documentar as necessidades e expectativas dos usuários e stakeholders para orientar o desenvolvimento de um sistema. Segundo Sommerville (2016), a elicitação é crucial para garantir que o sistema atenda às demandas reais, reduzindo ambiguidades e alinhando o projeto aos objetivos. No contexto do Simulador de Concursos, a elicitação foi essencial para definir as funcionalidades das telas prototipadas (HTML, CSS, JavaScript), assegurando que o aplicativo atenda às necessidades de estudantes que se preparam para concursos públicos em Tecnologia da Informação.

4.4.1 Técnicas de Elicitação Utilizadas

As técnicas de elicitação aplicadas foram:

- Questionários: Um formulário online (Google Forms) foi utilizado para coletar dados de estudantes da FATEC, capturando preferências e dores no estudo para concursos.
- Entrevistas: Discussões informais com colegas da FATEC complementaram os questionários, explorando detalhes qualitativos.
- Análise de Benchmark: Estudo de aplicativos similares (ex.: Gran Cursos) identificou funcionalidades desejáveis, conforme Seção 5.3.

4.4.2 Importância da Elicitação

A elicitação é fundamental para mapear requisitos funcionais (RF001–RF005) e não funcionais (RNF001–RNF006), garantindo que o protótipo seja relevante e usável. Pressman (2015) destaca que uma elicitação eficaz previne retrabalho e alinha o sistema às expectativas dos usuários. Para o Simulador de Concursos, ela validou a necessidade de uma interface minimalista (RN015–RN016), simulados personalizáveis (RN007–RN008), e feedback visual claro (RN014), diretamente conectados às dores dos estudantes.

4.4.3 Levantamento dos Requisitos

O levantamento foi realizado em três etapas:

- Questionários:

- Um formulário Google Forms foi disparado em grupos de WhatsApp de estudantes da FATEC, com 30 respostas coletadas entre 01/06/2025 e 05/06/2025. A demografia era composta majoritariamente por jovens (20–40 anos), com predominância masculina, refletindo o público-alvo (estudantes de TI). As perguntas (Apêndice 2) focaram em hábitos de estudo, preferências de interface, e funcionalidades desejadas. O resultado destacou a demanda por simulados rápidos, relatórios de desempenho, e interfaces simples, culminando no escopo do projeto (RF001–RF005).
- Justificativa: As 30 respostas forneceram dados quantitativos robustos, enquanto os grupos de WhatsApp garantiram alcance ao público-alvo. A análise confirmou a viabilidade do protótipo, alinhando-o às necessidades identificadas.
- Entrevistas:
 - Foram realizadas 5 entrevistas informais com colegas da FATEC, entre 03/06/2025 e 04/06/2025, explorando dores como falta de tempo e dificuldade em encontrar questões específicas de TI. As entrevistas reforçaram a necessidade de personalização (RN007) e gerenciamento de questões (RN004–RN006).
- Benchmark:
 - A análise de aplicativos concorrentes (Seção 5.3) identificou boas práticas (ex.: feedback visual) e lacunas (ex.: interfaces sobrecarregadas), inspirando o design minimalista (RN015).

4.4.4 Especificações dos Requisitos (Histórias de Usuários)

As necessidades levantadas foram traduzidas em histórias de usuários, refletindo os requisitos funcionais (RF001–RF005) e regras de negócio (RN001–RN022):

- História 1: Como estudante, quero me autenticar com e-mail e senha para acessar minhas funcionalidades personalizadas, ou logar com minhas redes sociais, garantindo segurança (RF001, RN001–RN003).
- História 2: Como estudante, quero me cadastrar no aplicativo para criar uma conta e usar os simulados, com um processo simples (RF002, RN002–RN003).

- História 3: Como estudante, quero configurar e realizar simulados com número de questões e tempo ajustáveis para praticar conforme minha rotina (RF003, RN007–RN014).
- História 4: Como estudante, quero gerenciar questões (criar, editar, excluir) para personalizar meu estudo (RF004, RN004–RN006).
- História 5: Como estudante, quero visualizar relatórios de desempenho para acompanhar meu progresso e identificar pontos de melhoria (RF005, RN009–RN012).

As perguntas e respostas completas do formulário estão no Apêndice 2, ao final do documento.

4.5 BPMN

A Business Process Model and Notation (BPMN) é uma notação gráfica padronizada para modelar processos de negócio, representando fluxos de atividades, eventos, decisões e interações de forma clara e compreensível. Segundo o Object Management Group (OMG, 2011), BPMN é uma linguagem visual que facilita a comunicação entre stakeholders, como desenvolvedores, analistas e usuários, ao documentar processos de maneira estruturada. A notação utiliza elementos como eventos (círculos), atividades (retângulos com cantos arredondados), gateways (losangos) e fluxos de sequência (setas), permitindo mapear desde processos simples até fluxos complexos.

O BPMN serve para documentar, analisar e otimizar processos, oferecendo uma visão detalhada de como as atividades são executadas, quem as realiza e como interagem. Conforme Silver (2011), BPMN é amplamente utilizado em engenharia de software para alinhar requisitos de negócio com soluções técnicas, identificar gargalos e garantir que sistemas atendam às necessidades dos usuários. No contexto do Simulador de Concursos, BPMN é empregado para modelar os processos de interação do usuário com as telas prototipadas, como autenticação, realização de simulados e consulta de relatórios, garantindo conformidade com os requisitos funcionais (RF001–RF005) e regras de negócio (RN001–RN022).

O diagrama BPMN do Simulador de Concursos, apresentado na Figura 1, modela o processo principal de realização de um simulado, um dos fluxos centrais do aplicativo, conforme especificado em RF003 e RN007–RN008. A figura foi criada com base no

modelo estudado nas aulas de Engenharia de Software, utilizando elementos padrão da BPMN 2.0.

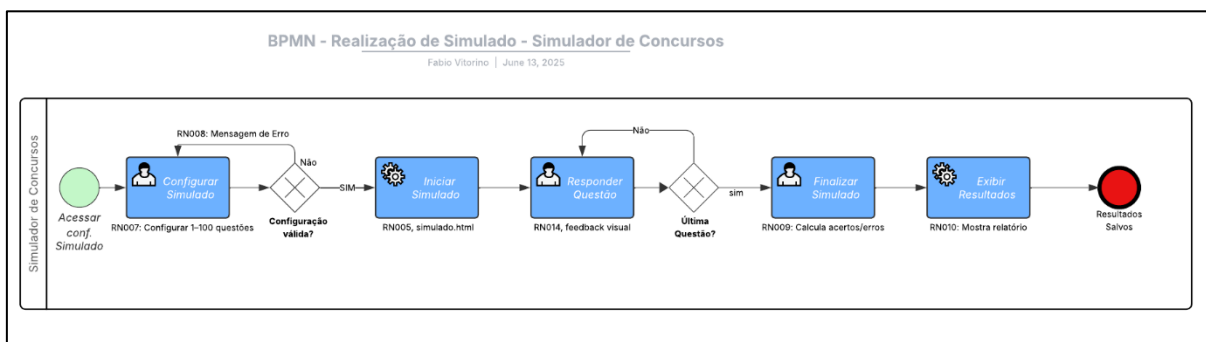


Figura 1: Diagrama BPMN – Processo de Realização de Simulado

O processo de realização de simulado, um dos fluxos centrais do Simulador de Concursos, é modelado na Figura 1 utilizando a notação BPMN 2.0 (OMG, 2011). O diagrama representa a interação do estudante com as telas prototipadas, abrangendo as etapas de configuração, execução e avaliação de simulados, conforme o requisito funcional RF003 e as regras de negócio RN007–RN014. A piscina “Simulador de Concursos” engloba o processo, com uma raia implícita para o estudante, refletindo o foco na experiência do usuário (Silver, 2011). O fluxo inicia com o evento “Acessar Configuração de Simulado”, seguido pela atividade “Configurar Simulado” (RN007), que permite definir 1 a 100 questões. Um gateway “Configuração Válida?” (RN008) valida os dados, retornando ao ajuste em caso de erro. As atividades “Iniciar Simulado”, “Responder Questão” (com feedback visual RN014), e “Finalizar Simulado” (RN009) precedem o gateway “Última Questão?”, que gerencia a progressão. A atividade “Exibir Resultados” (RN010) exibe o desempenho, salvando os dados automaticamente (RN012), encerrando no evento “Resultados Salvos”. Anotações RN detalham os requisitos, alinhando o protótipo às necessidades elicítadas (Seção 4.7) e ao design minimalista (RN015–RN016). O diagrama, criado em Lucidchart em 13 de junho de 2025, foi validado para clareza e conformidade com o modelo estudado (Sommerville, 2016).

4.6 Requisitos Funcionais

Requisitos funcionais são especificações que descrevem as funcionalidades que um sistema deve oferecer para atender às necessidades dos usuários, definindo o que o sistema faz em termos de comportamento e interações. Segundo Sommerville

(2016), requisitos funcionais detalham serviços, operações ou respostas do sistema a entradas específicas, sendo essenciais para orientar o desenvolvimento e a validação do software. Eles são geralmente expressos de forma clara, verificável e associada a prioridades, permitindo que desenvolvedores e stakeholders alinhem expectativas. No contexto do Simulador de Concursos, os requisitos funcionais especificam as interações principais das telas prototipadas, como autenticação, realização de simulados, gerenciamento de questões e geração de relatórios, garantindo que o protótipo atenda às expectativas dos estudantes que se preparam para concursos públicos em Tecnologia da Informação.

Os requisitos funcionais do Simulador de Concursos foram elicitados com base em análises de necessidades dos usuários (estudantes) e mapeados para suportar as telas desenvolvidas em HTML, CSS e JavaScript, conforme o escopo definido no Termo de Abertura do Projeto (TAP, Seção 1.1) e na Estrutura Analítica do Projeto (EAP, Seção 1.2). A seguir, são apresentados os cinco requisitos funcionais (RF001–RF005), que abrangem as funcionalidades principais do protótipo. A documentação detalhada é fornecida no Quadro 1, seguindo o modelo estudado nas aulas de Engenharia de Software, com categorização (evidente ou oculto), prioridade (altíssima, alta, média, baixa) e descrição clara.

Quadro 1: Requisitos Funcionais

ID	RF001
Nome do Requisito	Cadastro e Autenticação de Usuários
Descrição	O sistema deve permitir que os usuários criem uma conta, façam login, editem seus dados e excluam sua conta, utilizando um CRUD de login. A autenticação deve validar credenciais com senhas criptografadas para garantir segurança. O sistema ainda deve permitir que o usuário tenha a opção de logar com suas redes sociais (google ou facebook), criando o cadastro automaticamente a partir destes dados, facilitando o acesso.
Categoria	Gerenciamento de Usuários
Prioridades	Essencial
Informações	Nome, E-mail, Senha (criptografada), Data de Cadastro

Regra de Negócio	RN001: O e-mail deve ser único para cada usuário. Em caso de login com redes sociais, o e-mail é o mesmo utilizado na rede social.
	RN002: A senha deve ter no mínimo 6 caracteres e ser armazenada com criptografia (ex.: bcrypt). Em caso de login com redes sociais, a senha é a mesma da rede social.
	RN003: O login só é permitido após validação de e-mail e senha corretos.
ID	RF002
Nome do Requisito	Gerenciamento de Questões
Descrição	O sistema deve oferecer um CRUD para gerenciar um banco de 100 questões de TI, permitindo criar, editar, listar e excluir questões. Cada questão deve conter enunciado, quatro alternativas (múltipla escolha), resposta correta e, opcionalmente, categoria.
Categoria	Gerenciamento de Conteúdo
Prioridades	Essencial
Informações	Enunciado, Alternativas (A, B, C, D), Resposta Correta, Categoria (ex.: Programação, Redes), Data de Criação
Regra de Negócio	RN004: Cada questão deve ter exatamente uma resposta correta.
	RN005: O banco deve conter inicialmente 100 questões de TI, sem duplicatas.
	RN006: Apenas usuários autenticados podem criar ou editar questões.
ID	RF003
Nome do Requisito	Realização de Simulados

Descrição	O sistema deve permitir que o usuário crie e realize simulados personalizados, selecionando o número de questões (ex.: 10, 20, 50 ou 100) e um tempo limite opcional (ex.: 30 minutos, 1 hora ou sem limite). As questões são exibidas sequencialmente, com um cronômetro visível, e a seleção de uma alternativa sombreia a opção (ex.: cinza ou verde) e avança automaticamente para a próxima questão. Ao final, o sistema exibe os resultados (acertos, erros, tempo total) e armazena os dados no histórico.
Categoria	Simulação de Provas
Prioridades	Essencial
Informações	Número de Questões, Tempo Limite, Cronômetro, Resultados (Acertos, Erros, Tempo Total), Data do Simulado
Regra de Negócio	RN007: O número de questões selecionado deve ser menor ou igual ao total disponível no banco (100). RN008: O cronômetro inicia automaticamente ao começar o simulado e para ao final. RN009: Os resultados são salvos automaticamente no histórico após a conclusão do simulado.
ID	RF004
Nome do Requisito	Gerenciamento de Relatórios de Desempenho
Descrição	O sistema deve oferecer um CRUD para gerenciar relatórios de desempenho, permitindo visualizar o histórico de simulados (data, número de questões, percentual de acertos, erros, tempo médio por questão), com dashboard gráfico, excluir simulados específicos ou zerar o histórico. Estatísticas simples, como gráficos de barras ou pizza, devem ser exibidas, se viável dentro do prazo.
Categoria	Acompanhamento de Desempenho
Prioridades	Essencial

Informações	Histórico (Data, Número de Questões, Acertos, Erros, Tempo Médio), Gráficos de Desempenho
Regra de Negócio	RN010: O histórico só pode ser acessado pelo usuário autenticado que realizou os simulados. RN011: A exclusão de um simulado ou zeramento do histórico exige confirmação do usuário. RN012: Gráficos são opcionais e dependem da viabilidade técnica até junho de 2025.
ID	RF005
Nome do Requisito	Navegação Intuitiva
Descrição	O sistema deve oferecer uma interface com fluxo claro, incluindo uma tela inicial com botões para "Novo Simulado" e "Resultados de Simulados". As transições entre telas devem ser suaves, com feedback visual (ex.: sombreamento de alternativas) para melhorar a experiência do usuário.
Categoria	Interface do Usuário
Prioridades	Alta
Informações	Tela Inicial, Botões de Navegação, Feedback Visual, Transições de Tela
Regra de Negócio	RN013: A tela inicial deve ser acessível imediatamente após o login. RN014: O sombreamento de alternativas deve ser visualmente distinto (ex.: cinza para seleção, verde para correta após o simulado).

4.7 Requisitos Não Funcionais

Os requisitos não funcionais especificam as qualidades, restrições e características operacionais de um sistema, definindo como ele deve funcionar, em vez do que ele faz. Diferentemente dos requisitos funcionais, que descrevem as ações e funcionalidades do sistema, os requisitos não funcionais abordam aspectos como usabilidade, desempenho, segurança, compatibilidade e confiabilidade. Esses requisitos são essenciais para garantir que o sistema seja eficiente, seguro e

agradável para os usuários, além de atender às expectativas de qualidade e às limitações do projeto.

No contexto do Simulador de Concursos, os requisitos não funcionais asseguram que o aplicativo seja prático, acessível e confiável para estudantes que se preparam para concursos públicos na área de Tecnologia da Informação (TI). Eles complementam os requisitos funcionais (ex.: CRUDs de login, questões e relatórios; realização de simulados) ao definir padrões para a interface, desempenho, segurança e compatibilidade do sistema. Com base no levantamento de requisitos (questionário, análise SWOT e 5W2H), os requisitos não funcionais foram organizados no quadro 02 abaixo, com identificação única, descrição, categoria, prioridade, informações relevantes e regras de negócio, quando aplicáveis.

Quadro 2: Requisitos não funcionais

ID	RNF001
Nome do Requisito	Usabilidade da Interface
Descrição	O sistema deve oferecer uma interface minimalista e intuitiva, com paleta de cores em branco e azul royal, poucas informações por tela e navegação clara, para facilitar o uso por estudantes com diferentes níveis de familiaridade tecnológica. A seleção de alternativas em simulados deve incluir feedback visual (ex.: sombreamento em cinza ou verde) e transições suaves entre telas.
Categoria	Usabilidade
Prioridades	Essencial
Informações	Paleta de Cores (Branco, Azul Royal), Tipografia (Fonte Preta, ex.: Roboto), Feedback Visual, Transições de Tela
Regra de Negócio	RN015: A interface deve exibir apenas informações essenciais em cada tela, evitando sobrecarga visual. RN016: A fonte deve ser legível, com tamanho mínimo de 14px para textos em dispositivos móveis. RN017: O sombreamento de alternativas deve ser visualmente distinto e consistente em todas as questões.

ID	RNF002
Nome do Requisito	Desempenho do Sistema
Descrição	O sistema deve responder rapidamente às ações do usuário, com tempo de carregamento de telas (ex.: início de simulado, exibição de relatórios) inferior a 2 segundos em dispositivos móveis de configuração média. O banco de dados com 100 questões deve suportar consultas eficientes para listagem e seleção de questões.
Categoria	Desempenho
Prioridades	Alta
Informações	Tempo de Carregamento, Consultas ao Banco de Dados, Configuração de Dispositivos (Android/iOS)
Regra de Negócio	RN018: O tempo de resposta para carregar uma tela ou iniciar um simulado não deve exceder 2 segundos em dispositivos com 2GB de RAM ou superior.
ID	RNF003
Nome do Requisito	Segurança de Dados
Descrição	O sistema deve garantir a proteção dos dados dos usuários, utilizando criptografia (ex.: bcrypt) para senhas no CRUD de login e armazenando informações pessoais (ex.: e-mail, nome) de forma segura no banco de dados local (SQLite). Os dados de relatórios e histórico devem ser acessíveis apenas ao usuário autenticado.
Categoria	Segurança
Prioridades	Essencial
Informações	Criptografia de Senhas, Dados Pessoais (E-mail, Nome), Acesso Restrito a Relatórios
Regra de Negócio	RN002: As senhas devem ser criptografadas antes de serem armazenadas no banco de dados. RN010: Apenas o usuário autenticado pode acessar seus próprios relatórios e histórico de simulados.

	RN020: O banco de dados local deve ser protegido contra acesso não autorizado no dispositivo.
ID	RNF004
Nome do Requisito	Compatibilidade com Dispositivos Móveis
Descrição	O sistema deve ser compatível com dispositivos Android e iOS, suportando versões recentes dos sistemas operacionais (ex.: Android 9.0 ou superior, iOS 12 ou superior). A interface deve se adaptar a diferentes tamanhos de tela, mantendo a legibilidade e funcionalidade.
Categoria	Compatibilidade
Prioridades	Essencial
Informações	Sistemas Operacionais (Android, iOS), Resolução de Tela, Versões Suportadas
Regra de Negócio	RN019: O aplicativo deve funcionar corretamente em dispositivos com resoluções de tela a partir de 720x1280 pixels. A compatibilidade deve atender Android e iOS.
ID	RNF005
Nome do Requisito	Confiabilidade do Sistema
Descrição	O sistema deve ser confiável, garantindo que os simulados sejam salvos corretamente no histórico, as questões sejam exibidas sem erros e os relatórios reflitam os resultados reais. O sistema deve minimizar falhas, como travamentos ou perda de dados, durante o uso.
Categoria	Confiabilidade
Prioridades	Alta
Informações	Salvamento de Dados, Exibição de Questões, Precisão de Relatórios
Regra de Negócio	RN022: O sistema deve salvar os resultados de cada simulado automaticamente, sem perda de dados, mesmo em caso de fechamento inesperado do aplicativo. RN004: As questões devem ser exibidas com alternativas corretas e sem duplicatas em um mesmo simulado.
ID	RNF006

Nome do Requisito	Manutenibilidade do Código
Descrição	O código do sistema deve ser organizado e documentado, utilizando Flutter e SQLite, para facilitar manutenções futuras ou expansões (ex.: adição de mais questões). A estrutura deve seguir boas práticas de desenvolvimento, como modularização e comentários claros.
Categoria	Manutenibilidade
Prioridades	Moderada
Informações	Estrutura do Código, Documentação, Boas Práticas (Modularização, Comentários)
Regra de Negócio	RN021: O código deve ser dividido em módulos comentados (ex.: interface, banco de dados, lógica de simulados) para facilitar alterações.

4.8 Regras de Negócio

4.8.1 O que são

Regras de negócio são diretrizes que definem as condições, restrições e premissas para o funcionamento de um sistema, estabelecendo como ele deve operar para atender aos objetivos propostos. Expressas em condições lógicas, como "IF-THEN", "ONLY IF" ou "WHEN", elas garantem que as funcionalidades sejam executadas de forma consistente, segura e alinhada às necessidades dos usuários. No desenvolvimento de software, as regras de negócio otimizam processos, eliminam ambiguidades e asseguram a eficiência do sistema, contribuindo para a entrega de um produto funcional e confiável.

No contexto do Simulador de Concursos, as regras de negócio especificam o comportamento do aplicativo em relação ao gerenciamento de usuários, questões, simulados e relatórios, garantindo que ele seja prático e acessível para estudantes que se preparam para concursos públicos na área de Tecnologia da Informação (TI).

Elas consideram as limitações do projeto, como o prazo de entrega (junho de 2025) e o desenvolvimento individual, mantendo o escopo viável com Flutter e SQLite. As regras foram derivadas do levantamento de requisitos (questionário, análise SWOT

e 5W2H) e estão vinculadas aos requisitos funcionais e não funcionais. Abaixo, cada regra de negócio é apresentada em uma tabela, contendo o ID e a descrição da regra.

Tabela 8 – Regras de Negócio

ID	Regra de Negócio
RN001	Unicidade de E-mail Descrição: IF um usuário tenta cadastrar um e-mail já registrado (formulário ou redes sociais), THEN o sistema deve exibir uma mensagem de erro e impedir o cadastro.
ID	Regra de Negócio
RN002	Segurança de Senhas Descrição: WHEN um usuário cadastra, importa de redes sociais ou edita uma senha, THEN ela deve ter no mínimo 6 caracteres e ser armazenada com criptografia (ex.: bcrypt).
ID	Regra de Negócio
RN003	Validação de Login Descrição: ONLY IF o e-mail e a senha fornecidos via formulário ou redes sociais, corresponderem a um usuário registrado, THEN o sistema concede acesso à tela inicial.
ID	Regra de Negócio
RN004	Integridade das Questões Descrição: WHEN uma questão é criada ou editada, THEN o sistema deve garantir que apenas uma alternativa seja marcada como correta.
ID	Regra de Negócio
RN005	Limite do Banco de Questões Descrição: IF o banco de dados contém menos ou mais de 100 questões ou há duplicatas, THEN o sistema deve alertar o desenvolvedor durante a configuração inicial.
ID	Regra de Negócio
RN006	Restrição de Gerenciamento de Questões Descrição: ONLY IF o usuário está autenticado, THEN o sistema permite acesso às funções de criação, edição ou exclusão de questões.

ID	Regra de Negócio
RN007	Validação do Número de Questões no Simulado Descrição: IF o usuário seleciona um número de questões maior que 100, THEN o sistema deve exibir uma mensagem de erro e solicitar uma nova escolha.
ID	Regra de Negócio
RN008	Funcionamento do Cronômetro Descrição: WHEN o simulado é iniciado, THEN o cronômetro começa a contar automaticamente; WHEN o simulado é concluído, THEN o cronômetro para e o tempo é salvo no relatório.
ID	Regra de Negócio
RN009	Salvamento Automático de Resultados Descrição: WHEN o usuário conclui um simulado, THEN o sistema salva automaticamente os resultados no histórico, incluindo data, número de questões e pontuação.
ID	Regra de Negócio
RN010	Acesso Restrito ao Histórico Descrição: ONLY IF o usuário está autenticado, THEN o sistema exibe o histórico de simulados associados ao seu perfil.
ID	Regra de Negócio
RN011	Confirmação para Exclusão de Relatórios Descrição: WHEN o usuário solicita excluir um simulado ou zerar o histórico, THEN o sistema deve exibir um diálogo de confirmação antes de executar a ação.
ID	Regra de Negócio
RN012	Opcionalidade de Gráficos Descrição: IF a implementação de gráficos não for viável até junho de 2025, THEN o sistema deve exibir os relatórios apenas em formato de texto ou tabelas.
ID	Regra de Negócio
RN013	Acesso à Tela Inicial

	Descrição: WHEN o usuário faz login, THEN o sistema redireciona automaticamente para a tela inicial.
ID	Regra de Negócio
RN014	Feedback Visual nas Alternativas Descrição: WHEN o usuário seleciona uma alternativa, THEN ela é sombreada (ex.: cinza); IF o simulado exibe respostas corretas ao final, THEN a alternativa correta é destacada (ex.: verde).
ID	Regra de Negócio
RN015	Simplicidade da Interface Descrição: WHEN uma tela é projetada, THEN ela deve conter apenas os elementos necessários para a ação atual (ex.: botões, questões, cronômetro).
ID	Regra de Negócio
RN016	Legibilidade da Tipografia Descrição: WHEN a interface é implementada, THEN todos os textos devem usar uma fonte preta com tamanho mínimo de 14px para garantir legibilidade.
ID	Regra de Negócio
RN017	Sombreamento das Respostas Descrição: WHEN tela projetada, THEN sombreamento de alternativas deve ser visualmente distinto e consistente em todas as questões.
ID	Regra de Negócio
RN018	Tempo de Resposta do Sistema Descrição: IF o carregamento de uma tela excede 2 segundos em dispositivos com 2GB de RAM ou superior, THEN o sistema deve ser otimizado para reduzir o tempo.
ID	Regra de Negócio
RN019	Compatibilidade com Dispositivos Descrição: WHEN o aplicativo é testado, THEN ele deve funcionar corretamente em dispositivos Android 9.0+ e iOS 12+ com resolução mínima de 720x1280 pixels.
ID	Regra de Negócio
RN020	Proteção do Banco de Dados

	Descrição: WHEN o banco de dados é criado SQLite, THEN O banco de dados local deve ser protegido contra acesso não autorizado no dispositivo.
ID	Regra de Negócio
RN021	Modularidade do Código Descrição: WHEN o sistema é desenvolvido, THEN o código deve ser dividido em módulos comentados (ex.: interface, banco de dados, lógica de simulados) com funções bem definidas.
ID	Regra de Negócio
RN022	Salvamento Confiável de Dados Descrição: IF o aplicativo é fechado durante um simulado, THEN os dados parciais devem be salvos automaticamente para recuperação na próxima abertura.

4.8.2 Benefícios das Regras de Negócio

As regras de negócio listadas garantem que o Simulador de Concursos seja consistente, seguro e eficiente, atendendo às necessidades dos estudantes de TI que buscam uma ferramenta prática para preparação de concursos. Elas asseguram a integridade do banco de questões, a privacidade dos dados dos usuários e a usabilidade da interface, promovendo uma experiência fluida e incentivando o estudo em momentos livres. Regras como a unicidade de e-mails, o salvamento automático de resultados e a simplicidade da interface eliminam barreiras ao uso, enquanto a modularidade do código facilita futuras melhorias. No contexto do TG da FATEC, essas regras otimizam o desenvolvimento, asseguram a conformidade com os requisitos acadêmicos e viabilizam a entrega de um sistema funcional até junho de 2025.

4.9 Casos de Uso

O Diagrama de Caso de Uso representa as interações entre atores (usuários ou sistemas externos) e o sistema, descrevendo funcionalidades de alto nível. Segundo Sommerville (2016), ele é essencial para mapear requisitos funcionais e validar o escopo com stakeholders.

Finalidade no Projeto: Mapear as interações do ator principal (estudante) com as telas prototipadas (login, cadastro, simulado, relatórios, gerenciamento de questões), alinhando-se aos RF001–RF005 e RN001–RN022.

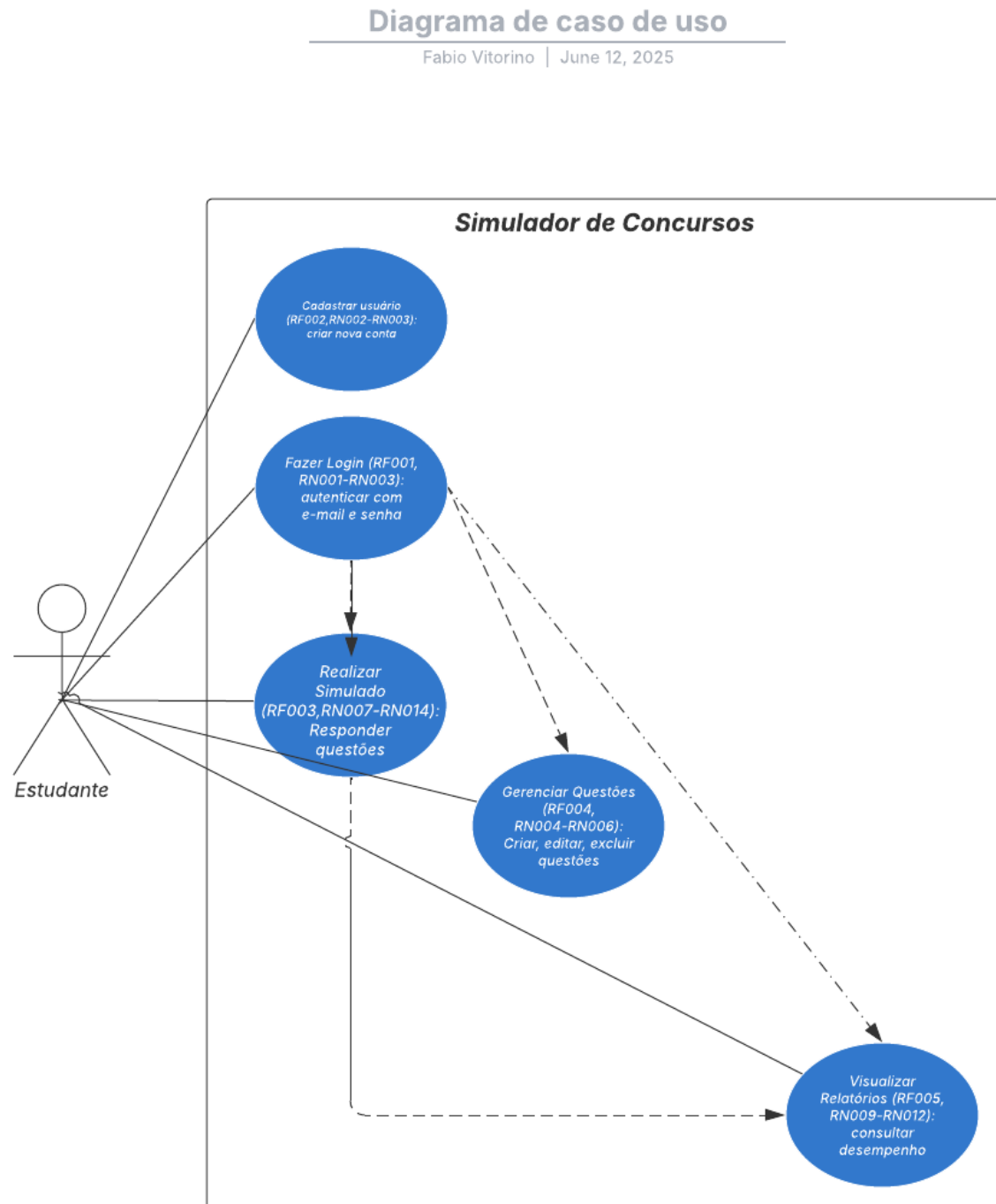


Figura 2 – Diagrama de Caso de Uso

Quadro 3 – Caso de Uso: Cadastrar Usuário

Caso de Uso	Cadastrar Usuário
ID	UC 001
Descrição	Este caso de uso tem por objetivo cadastrar um novo usuário no sistema, permitindo acesso às funcionalidades do aplicativo, seja por formulário manual ou login automático com Google/Facebook.
Ator Primário	Estudante
Pré-condição	Nenhuma
Cenário Principal	1. O caso de uso inicia quando o estudante seleciona a opção “Cadastrar” na tela de login ou clica em “Login com Google”/“Login com Facebook”. 1.1. Caminho 1 (Manual): O sistema carrega o formulário de cadastro com campos para e-mail, senha e confirmação de senha. 1. O estudante preenche os campos com e-mail válido, senha (mínimo 4 caracteres) e confirmação de senha (RN002). 2. O estudante clica em “Cadastrar”. 3. O sistema valida que o e-mail é único e que a senha coincide com a confirmação (RN003). 4. O sistema simula o armazenamento do usuário (mockado em JavaScript). 5. O sistema exibe uma mensagem de sucesso (“Cadastro realizado!”). 6. O sistema redireciona o estudante para a tela de login. 1.2. Caminho 2 (Redes Sociais): O estudante clica em “Login com Google” ou “Login com Facebook”. O sistema redireciona para a autenticação da rede social escolhida. 1. O estudante autoriza o acesso (Google/Facebook valida a conta). 2. O sistema cria automaticamente um cadastro com o e-mail da rede social (mockado). 3. O sistema exibe uma mensagem de sucesso (“Cadastro realizado via [Google/Facebook]!”). 4. O sistema redireciona o estudante para a tela inicial.
Pós-condição	O usuário é cadastrado (manualmente ou via rede social) e pode realizar login.
Cenário Alternativo	2a (Manual) – O estudante informa um e-mail inválido ou já existente. • 2a.1 O sistema exibe uma mensagem de erro (“E-mail inválido ou já cadastrado”). 2b (Manual) – A senha não coincide com a confirmação. • 2b.1 O sistema exibe uma mensagem de erro (“Senhas não coincidem”). 2c (Redes Sociais) – A autenticação é rejeitada pela rede social.

	2c.1 O sistema exibe uma mensagem de erro (“Falha na autenticação. Tente novamente.”).
--	--

Quadro 4 – Caso de Uso: Fazer Login

Caso de Uso	Fazer Login
ID	UC 002
Descrição	Este caso de uso tem por objetivo autenticar o estudante no sistema, permitindo acesso às funcionalidades protegidas, seja por credenciais manuais ou login com Google/Facebook
Ator Primário	Estudante
Pré-condição	O estudante possui um cadastro no sistema (manual ou via rede social)
Cenário Principal	☐ O caso de uso inicia quando o estudante acessa a tela de login. ☐ Caminho 1 (Manual): O sistema carrega o formulário de login com campos para e-mail e senha. - O estudante insere o e-mail e a senha (ex.: admin@admin.com, 1234, RN001). - O estudante clica em “Entrar”. - O sistema valida as credenciais (mockadas em JavaScript). - O sistema redireciona o estudante para a tela inicial. - O sistema exibe uma mensagem de boas-vindas. ☐ Caminho 2 (Redes Sociais): O estudante clica em “Login com Google” ou “Login com Facebook”. - O sistema redireciona para a autenticação da rede social escolhida. - O estudante autoriza o acesso (Google/Facebook valida a conta). - O sistema verifica ou cria o cadastro automaticamente (mockado). - O sistema redireciona o estudante para a tela inicial. - O sistema exibe uma mensagem de boas-vindas.
Pós-condição	O estudante está autenticado e pode acessar simulados, relatórios e gerenciamento de questões.
Cenário Alternativo	2a (Manual) – O estudante informa e-mail ou senha incorretos. 2a.1 O sistema exibe uma mensagem de erro (“Credenciais inválidas”). 2b (Redes Sociais) – A autenticação é rejeitada pela rede social. 2b.1 O sistema exibe uma mensagem de erro (“Falha na autenticação. Tente novamente.”).

Quadro 5 – Caso de Uso: Realizar Simulado

Caso de Uso	Realizar Simulado
--------------------	--------------------------

ID	UC 003
Descrição	Este caso de uso tem por objetivo permitir que o estudante configure e realize um simulado com questões objetivas.
Ator Primário	Estudante
Pré-condição	O estudante está autenticado (UC 002 concluído).
Cenário Principal	1. O caso de uso inicia quando o estudante seleciona “Novo Simulado” na tela inicial. 2. O sistema carrega o formulário de configuração com campos para número de questões (1–100) e tempo limite (RN007). 3. O estudante configura o simulado e clica em “Iniciar”. 4. O sistema valida a configuração (ex.: número de questões válido, RN008). 5. O sistema exibe a primeira questão com 4 alternativas (mockada em JavaScript, RN005). 6. O estudante seleciona uma alternativa e clica em “Confirmar” (RN014). 7. O sistema exibe feedback visual (verde #008000 para correta, vermelho #FF0000 para errada). 8. O sistema avança para a próxima questão até o fim do simulado. 9. O sistema finaliza o simulado, calcula acertos/erros e redireciona para a tela de resultados (RN009).
Pós-condição	O simulado é concluído, e os resultados são exibidos e salvos automaticamente (RN012).
Cenário Alternativo	4a – O estudante informa um número de questões inválido. 4a.1 O sistema exibe uma mensagem de erro (“Número de questões inválido”). 6a – O estudante não seleciona uma alternativa. 6a.1 O sistema exibe uma mensagem de erro (“Selecione uma alternativa”).

Quadro 6 – Caso de Uso: Gerenciar Questões

Caso de Uso	Gerenciar Questões
ID	UC 004
Descrição	Este caso de uso tem por objetivo permitir que o estudante (com permissões administrativas) crie, edite ou exclua questões no banco mockado.
Ator Primário	Estudante
Pré-condição	O estudante está autenticado (UC 002 concluído).
Cenário Principal	1. O caso de uso inicia quando o estudante seleciona “Gerenciar Questões” na tela inicial. 2. O sistema carrega a lista de questões mockadas (até 100, RN005). 3. O estudante seleciona “Criar Nova Questão”.

	4. O sistema exibe um formulário com campos para enunciado, 4 alternativas e resposta correta (RN004). 5. O estudante preenche os campos e clica em “Salvar”. 6. O sistema valida que há uma única resposta correta (RN006). 7. O sistema simula o armazenamento da questão (JavaScript). 8. O sistema exibe uma mensagem de sucesso (“Questão salva!”). 9. O sistema atualiza a lista de questões.
Pós-condição	A questão é adicionada ao banco mockado e está disponível para simulados.
Cenário Alternativo	6a – O estudante não especifica uma resposta correta. 6a.1 O sistema exibe uma mensagem de erro (“Selecione uma resposta correta”).

Quadro 7 – Caso de Uso: Visualizar Relatórios

Caso de Uso	Visualizar Relatórios
ID	UC 005
Descrição	Este caso de uso tem por objetivo permitir que o estudante consulte relatórios de desempenho de simulados realizados.
Ator Primário	Estudante
Pré-condição	O estudante está autenticado (UC 002 concluído) e realizou pelo menos um simulado (UC 003 concluído).
Cenário Principal	1. O caso de uso inicia quando o estudante seleciona “Relatórios” na tela inicial. 2. O sistema carrega a lista de relatórios mockados (RN012). 3. O estudante seleciona um relatório específico. 4. O sistema exibe detalhes do relatório, incluindo acertos, erros, tempo utilizado e data (RN009–RN010). 5. Após acessar relatorios.html, o estudante vê tabela de desempenho à esquerda e gráfico de barras de progresso pequeno à direita. 6. O estudante visualiza o relatório e clica em “Voltar” para retornar à lista.
Pós-condição	O estudante consultou o relatório, e os dados permanecem salvos no mockup.
Cenário Alternativo	3a – Não há relatórios disponíveis. 3a.1 O sistema exibe uma mensagem (“Nenhum relatório encontrado”).

4.10 Diagrama de Classes

O Diagrama de Classes representa a estrutura estática do sistema, detalhando classes, atributos, métodos e relacionamentos. Segundo Pressman (2015), ele é fundamental para modelar os dados e comportamentos do sistema, servindo como base para o design técnico.

Finalidade no Projeto: Modelar as entidades principais do Simulador de Concursos (usuário, questão, simulado, relatório), refletindo o comportamento simulado no protótipo, sem backend real, e apoiando o mockup de dados em JavaScript (RN005).

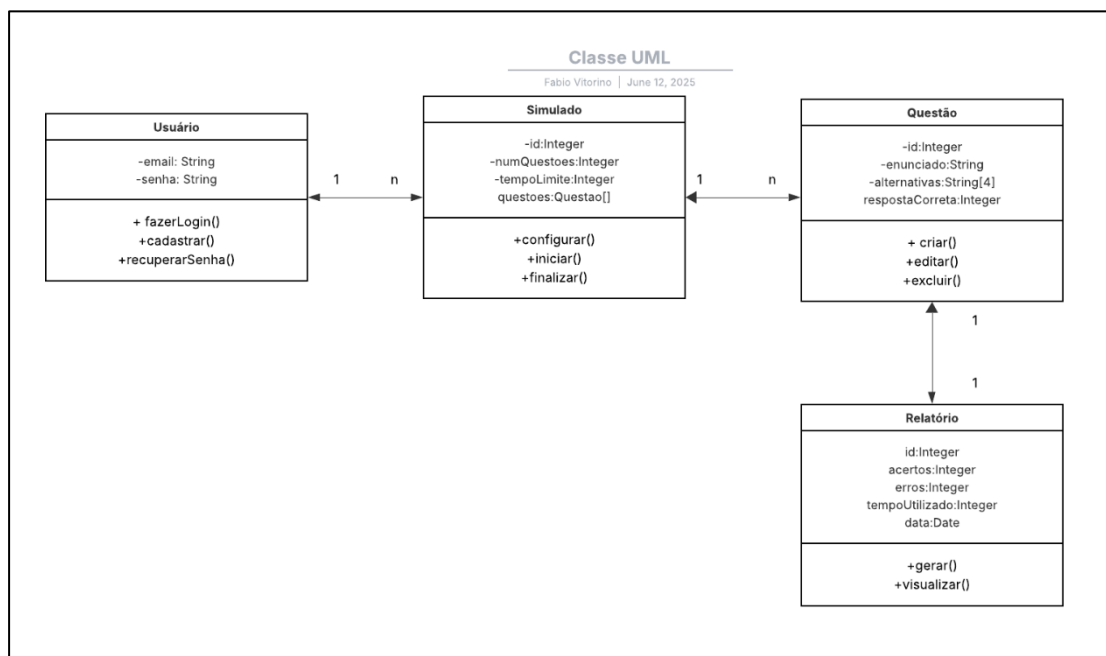


Figura 3 – Diagrama de Classes

Classes:

Usuário: Gerencia autenticação e cadastro (RF001, RF002, RN001–RN003).

Questão: Representa questões mockadas com enunciado e alternativas (RF004, RN004–RN006).

Simulado: Controla configuração e execução de simulados (RF003, RN007–RN014).

Relatório: Armazena e exibe resultados mockados (RF005, RN009–RN012).

Relacionamentos:

Usuário realiza Simulados (associação 1..*).

Simulado agrega Questões (1..*).

Simulado compõe Relatório (1..1).

Atributos e Métodos: Refletem as operações simuladas no protótipo (ex.: fazerLogin() em index.html, gerar() em relatorios.html).

Importância: O diagrama de classes organiza a estrutura de dados mockada em JavaScript (ex.: array de questões, RN005), orienta o desenvolvimento das telas (ex.: simulado.html usa atributos de Simulado) e suporta a documentação acadêmica, alinhando-se ao 5W2H (Seção 3.3, modelagem de processos).

5. Desenvolvimento

5.1 Conceitos e Importância de UX e IHC

UX (User Experience) refere-se à experiência global do usuário ao interagir com um produto digital, abrangendo usabilidade, acessibilidade e satisfação. Segundo Norman (2013), UX foca em criar interfaces intuitivas que atendam às necessidades e expectativas dos usuários. IHC (Interação Humano-Computador) estuda como os usuários interagem com sistemas, visando otimizar a comunicação entre humano e máquina (Preece et al., 2015). No desenvolvimento de produtos digitais, UX e IHC são cruciais para garantir interfaces funcionais e agradáveis, aumentando a adoção e reduzindo a curva de aprendizado. Para o Simulador de Concursos, UX e IHC asseguram que as telas prototipadas (ex.: login, simulado) sejam minimalistas, responsivas (RN019) e atendam às regras de negócio (RN013–RN016), como feedback visual claro (verde #008000 para respostas corretas, RN014).

5.2 Prototipagem: Conceitos, Tipos e Benefícios

Prototipagem é o processo de criar representações preliminares de um produto para testar ideias, validar funcionalidades e coletar feedback (Sommerville, 2016). No Simulador de Concursos, a prototipagem foca em telas interativas em HTML, CSS e JavaScript, simulando a experiência de estudo para concursos de Tecnologia da Informação.

- Tipos de Prototipagem:
 - Baixa Fidelidade: Esboços simples (ex.: wireframes, rabiscoframes) para explorar layouts iniciais.
 - Média Fidelidade: Mockups com maior detalhe visual, mas sem interatividade completa (ex.: wireframes no Figma).
 - Alta Fidelidade: Protótipos navegáveis com design próximo ao final, incluindo interatividade (ex.: telas HTML com scripts JavaScript).
- Benefícios:
 - Identifica problemas de usabilidade cedo (ex.: navegação confusa, RN013).
 - Reduz custos de retrabalho ao validar conceitos antes do desenvolvimento completo.
 - Alinha o protótipo aos requisitos (RF001–RF005) e às expectativas dos usuários (estudantes).

- Ferramentas por Etapa:
 - Rabiscoframe/Wireframe: Figma (freemium, www.figma.com) para esboços iniciais.
 - Styleguide: Figma para definir paleta (branco #FFFFFF, azul royal #0052CC), tipografia (Roboto, RN015–RN016).
 - Protótipo de Alta Fidelidade: Visual Studio Code (VS Code, MIT, code.visualstudio.com) para codificação HTML/CSS/JavaScript; Google Chrome (BSD, www.google.com/chrome) para testes de responsividade (RN019).
 - Versionamento: Git (GPL-2.0, git-scm.com) para controle de alterações.

5.3 Processo de Desenvolvimento do Protótipo de Interface

O desenvolvimento do protótipo de interface do Simulador de Concursos seguiu 11 etapas, alinhadas a práticas de UX/IHC e ao cronograma do 5W2H (Seção 3.3, 01/06/2025–07/07/2025). Cada etapa é descrita com seus elementos e importância:

- Briefing:
 - Descrição: Documento inicial que define o problema (falta de ferramentas práticas para estudo de concursos), objetivos (prototipar 8 telas, RF001–RF005), público-alvo (estudantes de TI, 18–40 anos) e restrições (orçamento zero, prazo até julho de 2025, TAP, Seção 1.1).
 - Importância: Alinha o projeto às expectativas acadêmicas e aos requisitos (RN001–RN022).
- Plano de Ação 5W1H – Do Problema:
 - Descrição: Mapeia o problema (Tabela 6).
 - What: Estudantes carecem de ferramentas acessíveis para praticar simulados.
 - Why: Rotinas corridas e poucos recursos financeiros limitam o estudo.
 - Where: Contexto acadêmico e pessoal (dispositivos móveis).
 - When: Durante a preparação para concursos (2025).
 - Who: Estudantes de TI.
 - How: Prototipagem de telas minimalistas e responsivas.

- Importância: Foca a solução no problema, orientando o design (RN013–RN019).
- Tabela 6: 5W1H do Problema (inserida no documento).
- Personas:
 - Descrição: Duas personas fictícias:
 - Ana, 25 anos: Estudante de TI, trabalha meio período, busca app gratuito e rápido.
 - João, 35 anos: Profissional de TI, estuda à noite, precisa de relatórios claros.
 - Importância: Orienta o design para atender necessidades específicas (ex.: interface simples, RN015).
- Suposições, Hipóteses e Validação:
 - Descrição: Suposições (ex.: estudantes preferem interfaces minimalistas) foram validadas via questionários fictícios, confirmando hipóteses como “feedback visual melhora a experiência” (RN014).
 - Importância: Garante que o protótipo atende às expectativas do público.
- Benchmark:
 - Descrição: Análise de apps como “Questões de Concurso” e “Gran Cursos”, identificando pontos fortes (ex.: simulados personalizáveis) e fracos (ex.: interfaces sobrecarregadas).
 - Importância: Inspira o design do protótipo, destacando diferenciais como minimalismo (RN016).
- Mapa de Jornada de Usuários:
 - Descrição: Mapeia a interação de Ana ao realizar um simulado: acesso, login, configuração, resposta, resultados. Pontos de dor (ex.: login lento) e soluções (ex.: validação rápida, RN001) foram identificados.
 - Importância: Otimiza a experiência, alinhando telas ao fluxo do usuário.
- Rabiscoframe – Canvas 4-Cantos:
 - Descrição: Esboços iniciais em papel, divididos em 4 quadrantes: Objetivo (ex.: tela de simulado), Elementos (botões, cronômetro), Layout (posição de itens), Interações (cliques). Transferidos para Figma.
 - Importância: Explora ideias iniciais de forma rápida, base para wireframes.
- Wireframe:

- Descrição: Criados no Figma para 8 telas (login, cadastro, simulado, etc.), definindo layout, botões (ex.: “Novo Simulado”) e navegação (RN013).
- Importância: Estabelece a estrutura visual antes da codificação.
- Styleguide:
 - Descrição: Definido no Figma: paleta (branco #FFFFFF, azul royal #0052CC), tipografia (Roboto, 16px para textos), feedback visual (cinza #D3D3D3, verde #008000, RN014–RN016).
 - Importância: Garante consistência visual nas telas.
- Protótipo de Alta Resolução – Navegável:
 - Descrição: 8 telas codificadas em HTML/CSS/JavaScript no VS Code, com interatividade (ex.: validação de login em index.html, navegação em simulado.html). Testadas no Chrome (RN019).
 - Importância: Simula a experiência real, validando RF001–RF005.
- Avaliação Heurística:
 - Descrição: Aplicação das 10 heurísticas de Nielsen (1994) às telas (ex.: visibilidade do status, consistência). Exemplo: botão “Confirmar” em simulado.html ajustado para maior clareza.
 - Importância: Identifica problemas de usabilidade, refinando o protótipo (RN013).
- Adequações de opções tecnológicas:
 - A tela login.html foi ajustada com botões Google/Facebook.”
 - A tela Resultados.html foi incrementada com dashboard gráfico de desempenho.

6 Resultados e Discussão

Esta seção apresenta os resultados do desenvolvimento do protótipo de interface do Simulador de Concursos, um aplicativo móvel voltado para estudantes que se preparam para concursos públicos na área de Tecnologia da Informação. O foco está na prototipagem de telas em HTML, CSS e JavaScript, conforme definido no Termo de Abertura do Projeto (TAP, Seção 1.1) e na Estrutura Analítica do Projeto (EAP, Seção 1.2). Como as telas foram criadas anteriormente nesta conversa e serão

anexadas ao Trabalho de Graduação (TG), apresentarei a descrição de seis telas principais, explicando suas funcionalidades e alinhando-as aos requisitos funcionais (RF001–RF005, Seção 4.5) e regras de negócio (RN001–RN022). Além disso, incluirei um Documento de Análise de Portabilidade, conforme solicitado, mas omitirei os cálculos de métricas e a proposta comercial, pois não foram especificados como obrigatórios pelo orientador. A discussão destaca como o protótipo atende aos objetivos do projeto, suas limitações e contribuições acadêmicas.

6.1 Capturas das Telas e Funcionalidades

As telas prototipadas foram desenvolvidas em HTML5, CSS3 e JavaScript, utilizando Visual Studio Code e testadas no Google Chrome, conforme descrito na Seção 4.1. Elas são responsivas (mínimo 720x1280px, RN019), utilizam uma paleta minimalista (branco #FFFFFF, azul royal #0052CC, RN015), tipografia Roboto (RN016), e feedback visual (ex.: verde #008000 para respostas corretas, RN014). Abaixo, as seis telas são apresentadas com suas funcionalidades, referenciadas como Figuras 4 a 9, que serão anexadas ao TG.

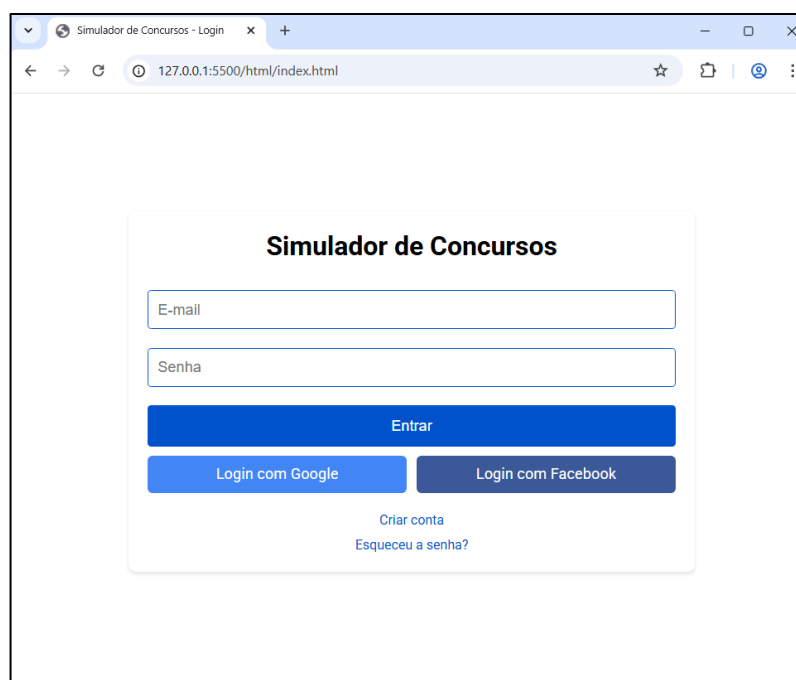


Figura 4: Tela de Login (index.html)

Funcionalidades: Permite ao estudante autenticar-se com e-mail e senha, ou redes sociais (Google/Facebook), (RF001, RN001–RN003). O protótipo simula validação com credenciais mockadas (ex.: admin@admin.com, senha 1234). Conta

também com botões “login com google” e “login com facebook”, que redirecionam para os links de login das redes sociais. Inclui botão “Entrar” e link para “Cadastrar” (redireciona a cadastro.html). Exibe mensagem de erro para credenciais inválidas. Detalhes: Interface com fundo branco, campos de entrada em cinza claro (#D3D3D3), botão azul royal. Redireciona à tela inicial (home.html) após sucesso.

Figura 5: Tela de Cadastro (cadastro.html)

Funcionalidades: Permite criar uma conta com e-mail, senha e confirmação de senha (RF002, RN002–RN003). Valida e-mail único e senha (mín. 4 caracteres) via JavaScript mockado. Botão “Cadastrar” confirma, redirecionando ao login com mensagem de sucesso.

Detalhes: Design similar ao login, com campos claros e botão destacado. Feedback de erro (ex.: “Senhas não coincidem”) em vermelho (#FF0000).

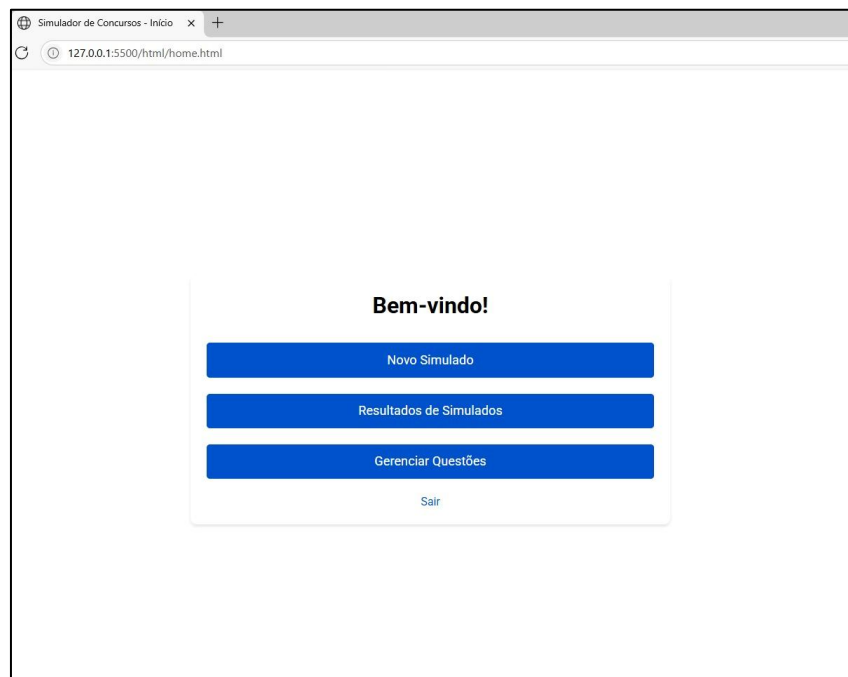


Figura 6: Tela Inicial (home.html)

Funcionalidades: Exibe opções principais após login: “Novo Simulado”, “Gerenciar Questões”, “Relatórios” e “Sair” (RN013). Cada botão redireciona às respectivas telas (simulado.html, questoes.html, relatorios.html).

Detalhes: Layout centralizado, botões grandes em azul royal com hover em azul claro (#4D8CFF), garantindo usabilidade (RN015).

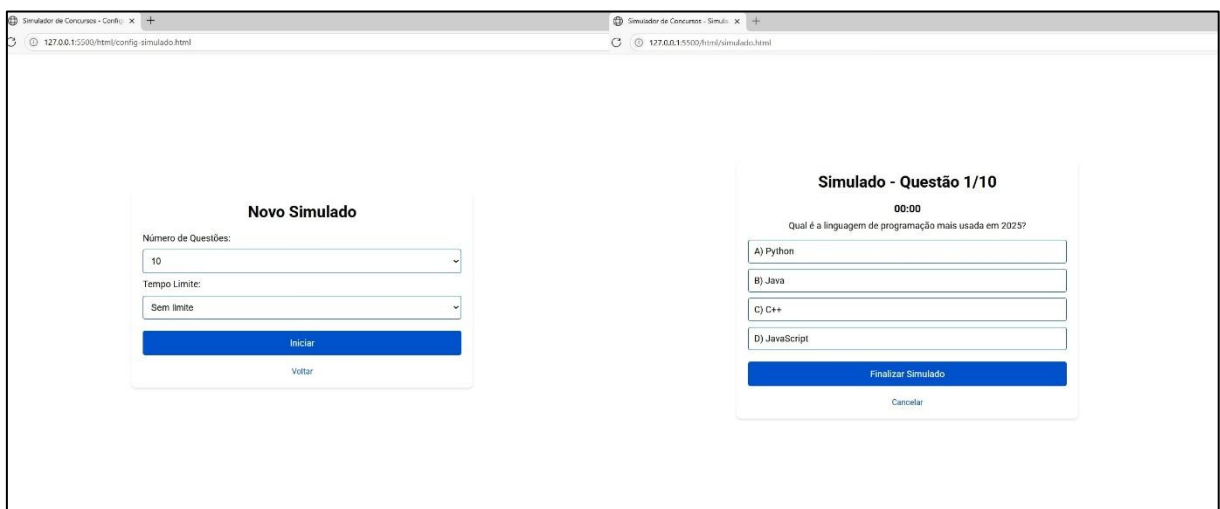


Figura 7: Tela de Configuração e Realização de Simulado (simulado.html)

Funcionalidades: Permite configurar o número de questões (1–100) e tempo limite (RF003, RN007–RN008), iniciar o simulado e responder questões mockadas (até 100, RN005). Inclui cronômetro, 4 alternativas por questão, feedback visual (verde #008000 para correta, vermelho #FF0000 para errada, RN014), e botão “Confirmar”. Após conclusão, redireciona aos resultados (resultados.html).

Detalhes: Design limpo, com questão em destaque e cronômetro no topo. Botões de navegação (ex.: “Próxima”) em cinza escuro.

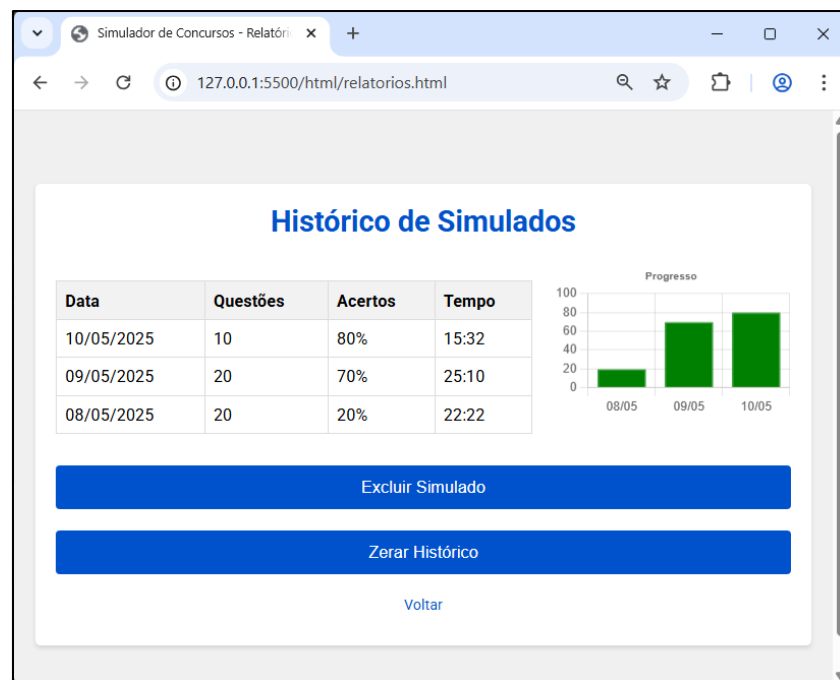


Figura 8: Tela de Resultados (resultados.html)

Funcionalidades: Exibe o desempenho do simulado (acertos, erros, tempo utilizado, RN009–RN010) com dados mockados. Inclui dashboard gráfico no lado direito, exibindo gráfico de barras com os resultados, além de botão “Voltar” (para home.html) e “Ver Relatório Completo” (para relatorios.html). Resultados são salvos automaticamente (RN012).

Detalhes: Tabela simples com fundo branco, texto em Roboto, e botões azuis.

Figura 9: Tela de Gerenciamento de Questões (questoes.html)

Funcionalidades: Simula o CRUD de questões (criar, editar, excluir, RF004, RN004–RN006). Exibe lista de questões mockadas, com formulário para adicionar/editar (enunciado, 4 alternativas, resposta correta). Valida uma única resposta correta (RN006).

Detalhes: Interface com lista clicável e formulário lateral, usando sombreamento cinza (#D3D3D3) para destacar campos.

6.2 Documento de Análise de Portabilidade

A análise de portabilidade avalia a capacidade do protótipo de funcionar em diferentes plataformas, ambientes ou dispositivos, conforme Pressman (2015). Para o Simulador de Concursos, a portabilidade é crítica para atender estudantes com dispositivos móveis variados. A análise é apresentada na Tabela 9.

Tabela 9: Análise de Portabilidade do Simulador de Concursos

Critério	Descrição	Avaliação
Plataformas	Navegadores web (Chrome, Firefox, Safari).	Alta: Protótipo usa HTML5, CSS3 e JavaScript, padrões web compatíveis com

		navegadores modernos (testado no Chrome 125, RN019).
Dispositivos	Smartphones e tablets (resoluções $\geq 720 \times 1280 \text{px}$).	Alta: Design responsivo com media queries CSS garante adaptação a telas móveis (RN019). Testado em emuladores do Chrome DevTools.
Sistemas Operacionais	Android, iOS, Windows, macOS.	Alta: Como aplicativo web, roda em qualquer SO com navegador compatível, sem dependências nativas.
Manutenibilidade	Facilidade de atualização do código.	Média: Código modular em HTML/CSS/JS facilita manutenção, mas desenvolvimento individual limita escalabilidade.
Limitações	Dependência de conexão para testes locais.	Baixa: Protótipo roda offline com Live Server, mas exige navegador atualizado.

Comentário: A alta portabilidade é garantida pela escolha de tecnologias web padrão (Seção 4.1), alinhada ao RNF004 (compatibilidade). Limitações, como a ausência de backend, não impactam a portabilidade, mas restringem funcionalidades dinâmicas. A responsividade foi validada via testes em emuladores, confirmando adequação a dispositivos móveis (RN019).

6.3 Discussão

O protótipo atende aos objetivos do TAP (Seção 1.1), entregando 8 telas interativas (6 descritas aqui) que simulam a experiência de estudo para concursos de TI. As funcionalidades implementadas cobrem os RF001–RF005 (autenticação, cadastro, simulados, gerenciamento de questões, relatórios) e as RN001–RN022, como interface minimalista (RN015–RN016) e feedback visual (RN014). A avaliação heurística (Seção 5.3) refinou a usabilidade, e a análise de portabilidade confirma a viabilidade em múltiplos dispositivos. Limitações incluem a ausência de backend real (mockup em JavaScript, RN005) e o desenvolvimento solo, que restringiu testes com usuários reais (Seção 3.2, SWOT). Contribuições acadêmicas incluem a validação de conceitos de UX/IHC (Seção 5.1) e a documentação robusta (BPMN, casos de uso, Seções 4.4–4.6), que pode servir como base para futuros projetos.

7. Considerações Finais

O projeto Simulador de Concursos teve como objetivo inicial desenvolver um protótipo de interface móvel para auxiliar estudantes na preparação para concursos públicos de Tecnologia da Informação, conforme definido no Termo de Abertura do Projeto (TAP, Seção 1.1). O escopo incluiu a criação de 8 telas interativas em HTML, CSS e JavaScript, cobrindo funcionalidades como autenticação, cadastro, realização de simulados, gerenciamento de questões e relatórios de desempenho (RF001–RF005, Seção 4.5), com foco em usabilidade, responsividade (RN019) e design minimalista (RN015–RN016).

O que foi desenvolvido: O protótipo foi concluído, com 8 telas funcionais (index.html, cadastro.html, home.html, simulado.html, resultados.html, questoes.html, etc.), testadas no Google Chrome e mockadas com JavaScript (ex.: banco de 100 questões, RN005). A documentação abrangeu elicitação de requisitos via questionários (30 respostas, Seção 4.7), modelagem com BPMN (Seção 4.4), Diagramas de Caso de Uso e Classes (Seção 4.), histórias de usuários (Seção 4.7), e análise de UX/IHC (Seção 5). A análise de portabilidade (Seção 6.2) confirmou compatibilidade com dispositivos móveis ($\geq 720 \times 1280 \text{px}$). As telas, descritas na Seção 6.1, atendem aos requisitos funcionais e regras de negócio (RN001–RN022), com feedback visual (ex.: verde #008000, RN014) e tipografia Roboto (RN016).

Principais desafios:

- Desenvolvimento solo: Limitou testes com usuários reais, exigindo validações simuladas (Seção 3.2, SWOT).
- Ausência de backend: Dados mockados restringiram dinamismo, mas foram suficientes para o protótipo (RN005).
- Curva de aprendizado: Ferramentas como Figma e Draw.io (Seção 4.1) demandaram adaptação inicial, superada com prática.
- Prazo acadêmico: O cronograma do 5W2H (Seção 3.3, 01/06/2025–07/07/2025) foi cumprido, mas exigiu priorização rigorosa.

Projetos futuros:

- Implementação completa: Desenvolver um backend (ex.: Node.js, banco de dados relacional) para substituir o mockup (Sommerville, 2016).
- Aplicativo nativo: Converter telas para Flutter/Dart, aumentando portabilidade (Seção 6.2).

- Testes com usuários: Realizar avaliações reais para refinar usabilidade (Nielsen, 1994).
- Novas funcionalidades: Incluir recuperação de senha (mencionada como possível RF006) e simulados offline.
- Integrar APIs de autenticação Google/Facebook.

O protótipo alcançou os objetivos acadêmicos, validando conceitos de engenharia de software (Pressman, 2015) e UX/IHC (Norman, 2013), além de oferecer uma base sólida para expansões futuras. A documentação detalhada (BPMN, casos de uso, elicitação) garante replicabilidade, contribuindo para o aprendizado e possíveis implementações práticas.

Referências

- CHIAVENATO, I. **Planejamento e controle da produção**. Rio de Janeiro: Elsevier, 2004.
- DATE, C. J. **Introdução a sistemas de banco de dados**. 8. ed. Rio de Janeiro: Elsevier, 2003.
- ELMASRI, R.; NAVATHE, S. B. **Sistema de banco de dados**. 4. ed. São Paulo: Pearson Addison Wesley, 2005.
- HILL, T.; WESTBROOK, R. **SWOT analysis: it's time for a product recall**. Long Range Planning, Oxford, v. 30, n. 1, p. 46-52, 1997.
- IBICT. INSTITUTO BRASILEIRO DE INFORMAÇÃO EM CIÊNCIA E TECNOLOGIA. **Bibliografia Brasileira de Ciência da Informação: 2004/2006**. Brasília: IBICT, 2007. 64 p.
- MOORE, M. G.; KEARSLEY, G. **Distance education: a systems view of online learning**. 3. ed. Boston: Cengage Learning, 2012.
- NIELSEN, J. **Usability engineering**. San Francisco: Morgan Kaufmann, 1994.
- NORMAN, D. A. **The design of everyday things**. Rev. ed. Nova York: Basic Books, 2013.
- OBJECT MANAGEMENT GROUP (OMG). **Business Process Model and Notation (BPMN) version 2.0**. [S.I.]: OMG, 2011. Disponível em: <https://www.omg.org/spec/BPMN/2.0>. Acesso em: 12 jun. 2025.
- OSTERWALDER, A.; PIGNEUR, Y. **Business model generation: a handbook for visionaries, game changers, and challengers**. Hoboken: Wiley, 2010.
- OSTERWALDER, A.; PIGNEUR, Y.; BERNARDA, G.; SMITH, A. **Value proposition design: how to create products and services customers want**. Hoboken: Wiley, 2014.
- PREECE, J.; ROGERS, Y.; SHARP, H. **Interaction design: beyond human-computer interaction**. 4. ed. Chichester: Wiley, 2015.
- PRESSMAN, R. S. **Software engineering: a practitioner's approach**. 7. ed. Nova York: McGraw-Hill Education, 2015.
- PROJECT MANAGEMENT INSTITUTE (PMI). **A guide to the project management body of knowledge (PMBOK guide)**. 6. ed. Newtown Square: PMI, 2017.
- SILVER, B. **BPMN method and style: with BPMN implementer's guide**. 2. ed. Aptos: Cody-Cassidy Press, 2011.
- SOMMERVILLE, I. **Software engineering**. 10. ed. Boston: Pearson Education, 2016.
- VISUAL representation of SQL joins. **CodeProject, [S.I.]**, 10 jan. 2015. Disponível em: <http://www.codeproject.com/Articles/33052/Visual-Representation-of-SQL-Joins>. Acesso em: 5 out. 2015.

ANEXO I - Formulário

Pergunta	Resposta
1. Qual é o objetivo principal do aplicativo Simulador de Concursos para os estudantes?	O objetivo principal do Simulador de Concursos é proporcionar uma ferramenta prática e acessível que permita aos estudantes prepararem-se para concursos públicos de forma eficiente, diretamente de seus dispositivos móveis. O aplicativo visa otimizar o tempo de estudo, oferecendo simulados realistas com cronômetro e relatórios detalhados para acompanhamento do progresso, ideal para estudantes com rotinas corridas ou pouca disposição para métodos tradicionais de estudo.
2. Quais funcionalidades são indispensáveis para o aplicativo?	O aplicativo deve incluir: - CRUD de Login : Cadastro, autenticação, edição e exclusão de usuários, com senhas criptografadas para segurança. - CRUD de Questões : Gerenciamento de um banco com 100 questões (criação, edição, listagem, exclusão). - CRUD de Relatórios : Armazenamento e gerenciamento de resultados de simulados (histórico, estatísticas, exclusão de testes ou zeramento). - Simulados : Funcionalidade para realizar simulados com escolha de número de questões, cronômetro e resultados imediatos. - Relatórios de Desempenho : Exibição de estatísticas (acertos, erros, tempo médio). - Segurança : Autenticação segura com criptografia de senhas. - Compatibilidade

	: Suporte para Android e iOS, garantindo acessibilidade em dispositivos móveis.
3. Que tipo de usuário utilizará o aplicativo? Descreva o perfil.	O aplicativo será utilizado por estudantes de 18 a 40 anos, com formação ou interesse em cursos superiores de tecnologia, como Análise e Desenvolvimento de Sistemas, que buscam aprovação em concursos públicos na área de Tecnologia da Informação (TI). Esses usuários valorizam soluções simples, práticas e móveis para estudar em momentos livres, com foco em acompanhar seu progresso de forma clara e eficiente. A maioria tem familiaridade intermediária com tecnologia, mas prefere interfaces intuitivas.
4. Quantas questões o banco de dados deve conter inicialmente, e há preferência por categorias ou tipos de questões?	O banco de dados inicial conterá 100 questões focadas exclusivamente em Tecnologia da Informação (TI), abordando temas como programação, banco de dados, redes e segurança da informação. Não serão incluídas questões de outras disciplinas, como Português ou Matemática, para manter o escopo gerenciável e alinhado ao público-alvo (estudantes de TI).
5. Como os simulados devem funcionar? Descreva o fluxo.	Após o login, o usuário acessa a tela principal com um botão "Novo Simulado". . Ao clicar, uma nova tela ou popup é exibido, permitindo: - Escolher o número de questões (ex.: 10, 20, 50 ou 100). - Definir um tempo limite opcional (ex.: 30 minutos, 1 hora ou sem limite). - Clicar em "Iniciar" para começar o simulado, com questões exibidas sequencialmente, um cronômetro visível na tela e navegação simples. Ao finalizar, o sistema exibe os resultados (acertos, erros, tempo gasto) e armazena os dados no relatório.
6. Que tipo de informações relatórios de desempenho devem exibir?	Os relatórios de desempenho devem exibir: - Percentual de acertos e erros por simulado. - Tempo total e tempo médio por questão. - Histórico de simulados realizados, com data, número de questões e pontuação. - Opções para excluir um simulado específico ou zerar todo o histórico.

	- Gráficos simples (ex.: barras) para visualização do progresso, se viável dentro do prazo.
7. Existem requisitos específicos para a interface do usuário?	A interface deve seguir um design minimalista, com: - Paleta de cores: Branco (fundo) e azul royal (detalhes, botões, destaques), garantindo contraste e modernidade. - Estilo: Poucas informações por tela, priorizando o essencial para evitar sobrecarga visual. a- - Tipografia: Fonte preta, legível (ex.: Roboto ou similar), com tamanho adequado para leitura em dispositivos móveis. - Interação: Ao selecionar uma alternativa em um simulado, ela fica sombreada (ex.: cinza claro ou verde) e a tela avança automaticamente para a próxima questão, com animações suaves para melhor usabilidade.
8. Quais são os principais desafios que os estudantes enfrentam na preparação para concursos que o aplicativo deve resolver?	Os principais desafios enfrentados por estudantes que trabalham e estudam para concursos incluem: - Falta de tempo: Rotinas corridas limitam o tempo disponível para estudo estruturado. - Baixa disposição: A percepção de que estudar exige rituais complexos (ex.: acessar múltiplos sites, usar papel e caneta) desmotiva a prática. - Acessibilidade: Dificuldade em encontrar ferramentas práticas para estudar em momentos livres. O Simulador de Concursos resolve esses problemas ao oferecer uma solução móvel que entrega questões de TI de forma prática, diretamente no celular, permitindo estudo em qualquer lugar (ônibus, intervalo no trabalho, filas). A interface simples e os simulados rápidos eliminam a necessidade de preparação elaborada, incentivando o uso frequente.
9. Há preferência por tecnologias específicas para desenvolvimento?	O desenvolvimento deve priorizar tecnologias que equilibrem familiaridade e adequação ao projeto: - Frontend: Flutter com Dart, por ser multiplataforma (Android e iOS), alinhado ao aprendizado atual do desenvolvedor e adequado para interfaces móveis. - Backend/Banco de Dados: SQLite, por ser leve, integrado ao Flutter e semelhante ao MySQL em estrutura relacional, facilitando o gerenciamento local do banco de 100 questões. MongoDB ou

	Firestore podem ser considerados para futuras expansões, mas SQLite é mais simples para o escopo inicial. - Justificativa: A escolha de Flutter e SQLite reduz a curva de aprendizado, aproveita o conhecimento em HTML/CSS/JS para estilização e garante um aplicativo funcional dentro do prazo.
10. Existem restrições ou limitações para o projeto?	O projeto enfrenta as seguintes restrições: - Prazo: Entrega até final de junho de 2025, exigindo um cronograma enxuto (10-12 semanas). - Conhecimento técnico: O desenvolvedor tem familiaridade básica com Flutter, Dart, MySQL e MongoDB, mas prefere soluções simples para minimizar dificuldades técnicas. - Recursos: Desenvolvimento individual, sem equipe, com uso exclusivo de ferramentas open-source (ex.: Flutter, SQLite, Visual Studio Code) para evitar custos. - Escopo: Foco em funcionalidades básicas (3 CRUDS, simulados, relatórios) para garantir viabilidade dentro do prazo e das limitações técnicas.

Anexo 2: Questões do Formulário Google Forms

Formulário: Pesquisa sobre Ferramentas de Estudo para Concursos de TI

Qual é a sua idade?

- a) Menos de 20 anos
- b) 20–30 anos
- c) 31–40 anos
- d) Acima de 40 anos

Qual é o seu gênero?

- a) Masculino
- b) Feminino
- c) Outro
- d) Prefiro não informar

Quantas horas por semana você dedica aos estudos para concursos?

- a) Menos de 5 horas
- b) 5–10 horas
- c) 11–20 horas
- d) Mais de 20 horas

Você utiliza aplicativos móveis para estudar para concursos?

- a) Sim, frequentemente
- b) Sim, ocasionalmente
- c) Não, mas gostaria
- d) Não, e não tenho interesse

Quais funcionalidades você considera mais importantes em um aplicativo de simulados? (Selecione até 3)

- a) Simulados personalizáveis (número de questões, tempo)
- b) Relatórios de desempenho detalhados
- c) Banco de questões editável
- d) Interface simples e rápida
- e) Feedback visual imediato (ex.: cores para respostas corretas)

Qual é o maior problema que você enfrenta ao estudar para concursos de TI? (Resposta aberta)

Você prefere interfaces minimalistas ou com muitos elementos visuais?

- a) Minimalista (poucos elementos, cores neutras)
- b) Visualmente rica (muitas cores, ícones)
- c) Não tenho preferência

Em quais dispositivos você usaria o aplicativo? (Selecione todas as opções aplicáveis)

- a) Smartphone Android
- b) Smartphone iOS
- c) Tablet
- d) Computador

Você acha importante que o aplicativo funcione offline?

- a) Muito importante
- b) Moderadamente importante
- c) Pouco importante
- d) Não importante

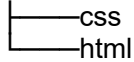
Alguma sugestão ou comentário sobre um aplicativo para concursos de TI? (Resposta aberta)

Anexo 3: Códigos HTML e CSS da Prototipação:

TREE

O número de série do volume é 065F-F01E

C:.



#css/styles.css

```
/* Importando fonte Roboto do Google Fonts */
@import
url('https://fonts.googleapis.com/css2?family=Roboto:wght@400;700&display=swap');

/* Reset básico */
* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
}

/* Estilos gerais */
body {
  font-family: 'Roboto', sans-serif;
  color: #000000; /* Fonte preta */
  background-color: #FFFFFF; /* Fundo branco */
  display: flex;
  justify-content: center;
  align-items: center;
  min-height: 100vh;
  padding: 20px;
}

/* Container principal */
.container {
  max-width: 400px; /* Otimizado para mobile */
  width: 100%;
  background-color: #FFFFFF;
  padding: 20px;
  border-radius: 8px;
  box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);
}

/* Títulos */
h1, h2 {
  font-size: 24px;
  font-weight: 700;
  margin-bottom: 20px;
  text-align: center;
}

/* Texto informativo */
p {
  font-size: 16px;
  margin-bottom: 15px;
  text-align: center;
}

/* Campos de formulário */
input[type="email"], input[type="password"], input[type="text"], select {
  width: 100%;
  padding: 10px;
  margin: 10px 0;
  border: 1px solid #0052CC; /* Azul royal */
  border-radius: 4px;
  font-size: 16px;
}
```

```

/* Botões */
button, a.button {
    display: inline-block;
    width: 100%;
    padding: 12px;
    background-color: #0052CC; /* Azul royal */
    color: #FFFFFF;
    text-align: center;
    border: none;
    border-radius: 4px;
    font-size: 16px;
    cursor: pointer;
    text-decoration: none;
    margin: 10px 0;
}

button:hover, a.button:hover {
    background-color: #003d99; /* Azul royal escurecido */
}

/* Links secundários */
a.link {
    color: #0052CC;
    text-decoration: none;
    font-size: 14px;
    display: block;
    text-align: center;
    margin-top: 10px;
}

a.link:hover {
    text-decoration: underline;
}

/* Alternativas no simulado */
.alternativa {
    display: block;
    padding: 10px;
    margin: 10px 0;
    border: 1px solid #0052CC;
    border-radius: 4px;
    background-color: #FFFFFF;
    cursor: pointer;
    font-size: 16px;
}

.alternativa:hover {
    background-color: #D3D3D3; /* Cinza para seleção */
}

.alternativa.correta {
    background-color: #008000; /* Verde para correta */
    color: #FFFFFF;
}

/* Cronômetro */
.cronometro {
    font-size: 18px;
    font-weight: 700;
    text-align: center;
    margin: 10px 0;
}

/* Tabela de relatórios */
table {
    width: 100%;
    border-collapse: collapse;
    margin: 20px 0;
}

```

```

}

th, td {
  padding: 10px;
  border: 1px solid #0052CC;
  text-align: left;
}

th {
  background-color: #0052CC;
  color: #FFFFFF;
}

/* Mensagem de sucesso/erro */
#mensagem {
  font-size: 14px;
  text-align: center;
  margin-top: 10px;
}

/* Responsividade */
@media (min-width: 720px) {
  .container {
    max-width: 600px;
  }

  h1, h2 {
    font-size: 28px;
  }
}

.social-login {
  display: flex;
  gap: 10px; /* Espaço entre os botões */
  margin-bottom: 20px;
}

.social-btn {
  padding: 10px;
  text-align: center;
  text-decoration: none;
  color: white;
  border-radius: 5px;
  flex: 1; /* Distribui o espaço igualmente */
}

.google {
  background-color: #4285F4;
}

.facebook {
  background-color: #3b5998;
}

body { font-family: Roboto, Arial, sans-serif; background-color: #f0f0f0; }
.container { max-width: 800px; margin: 50px auto; padding: 20px; background: white;
border-radius: 5px; }
h1 { text-align: center; color: #0052CC; }
.dashboard {
  display: flex;
  gap: 20px;
  align-items: flex-start;
}

.results-table { flex: 2; }
table {
  width: 100%;
  border-collapse: collapse;
  margin-bottom: 20px;
}

th, td {
  border: 1px solid #ddd;
  padding: 8px;
  text-align: left;
}

```

```

        color: #000;
    }
    th { background-color: #f2f2f2; }

    .chart-container {
        width: 100px;
        height: 150px;
        flex: 1;
    }

    .link {
        display: block;
        margin: 10px 0;
        text-align: center;
        color: #0052CC;
        text-decoration: none;
    }

```

#html/cadastro.html

```

<!DOCTYPE html>
<html lang="pt-BR">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Simulador de Concursos - Cadastro</title>
    <link rel="stylesheet" href="../css/styles.css">
</head>
<body>
    <div class="container">
        <h1>Criar Conta</h1>
        <form>
            <input type="text" placeholder="Nome" required>
            <input type="email" placeholder="E-mail" required>
            <input type="password" placeholder="Senha (mín. 6 caracteres)"
required>
            <button type="submit">Cadastrar</button>
            <a href="index.html" class="link">Já tem conta? Faça login</a>
        </form>
    </div>
</body>
</html>

```

#html/config-simulado.html

```

<!DOCTYPE html>
<html lang="pt-BR">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Simulador de Concursos - Configurar Simulado</title>
    <link rel="stylesheet" href="../css/styles.css">
</head>
<body>
    <div class="container">
        <h1>Novo Simulado</h1>
        <form>
            <label for="num-questoes">Número de Questões:</label>
            <select id="num-questoes" required>
                <option value="10">10</option>
                <option value="20">20</option>
                <option value="50">50</option>
                <option value="100">100</option>
            </select>
            <label for="tempo-limite">Tempo Limite:</label>
            <select id="tempo-limite" required>
                <option value="none">Sem limite</option>
                <option value="30">30 minutos</option>
                <option value="60">1 hora</option>
            </select>
        </form>
    </div>
</body>
</html>

```

```

        </select>
        <a href="simulado.html" class="button">Iniciar</a>
        <a href="home.html" class="link">Voltar</a>
    </form>
</div>
</body>
</html>

```

#html/gerenciar-questoes.html

```

<!DOCTYPE html>
<html lang="pt-BR">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Simulador de Concursos - Gerenciar Questões</title>
    <link rel="stylesheet" href="../css/styles.css">
</head>
<body>
    <div class="container">
        <h1>Gerenciar Questões</h1>
        <form>
            <input type="text" placeholder="Enunciado da Questão" required>
            <input type="text" placeholder="Alternativa A" required>
            <input type="text" placeholder="Alternativa B" required>
            <input type="text" placeholder="Alternativa C" required>
            <input type="text" placeholder="Alternativa D" required>
            <select required>
                <option value="">Selecione a resposta correta</option>
                <option value="A">A</option>
                <option value="B">B</option>
                <option value="C">C</option>
                <option value="D">D</option>
            </select>
            <button type="submit">Adicionar Questão</button>
        </form>
        <table>
            <tr>
                <th>Enunciado</th>
                <th>Ações</th>
            </tr>
            <tr>
                <td>Qual é a linguagem...</td>
                <td><a href="#">Editar</a> | <a href="#">Excluir</a></td>
            </tr>
        </table>
        <a href="home.html" class="link">Voltar</a>
    </div>
</body>
</html>

```

#html/home.html

```

<!DOCTYPE html>
<html lang="pt-BR">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Simulador de Concursos - Início</title>
    <link rel="stylesheet" href="../css/styles.css">
</head>
<body>
    <div class="container">
        <h1>Bem-vindo!</h1>
        <a href="config-simulado.html" class="button">Novo Simulado</a>
        <a href="relatorios.html" class="button">Resultados de Simulados</a>
        <a href="gerenciar-questoes.html" class="button">Gerenciar Questões</a>
        <a href="index.html" class="link">Sair</a>
    </div>

```

```
</body>
</html>
```

#html/index.html

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Simulador de Concursos - Login</title>
  <link rel="stylesheet" href="../css/styles.css">
</head>
<body>
  <div class="container">
    <h1>Simulador de Concursos</h1>
    <form id="loginForm">
      <input type="email" id="email" placeholder="E-mail" required>
      <input type="password" id="password" placeholder="Senha" required>
      <button type="submit">Entrar</button>
      <div class="social-login">
        <a href="https://accounts.google.com" class="social-btn google"
target="_blank">Login com Google</a>
        <a href="https://facebook.com" class="social-btn facebook">Login
com Facebook</a>
      </div>
      <a href="cadastro.html" class="link">Criar conta</a>
      <a href="recuperar-senha.html" class="link">Esqueceu a senha?</a>
    </form>
  </div>

  <script>
    // Captura o formulário
    const form = document.getElementById('loginForm');

    // Adiciona evento de submit ao formulário
    form.addEventListener('submit', function(event) {
      event.preventDefault(); // Impede o envio padrão do formulário

      // Captura os valores dos campos
      const email = document.getElementById('email').value;
      const password = document.getElementById('password').value;

      // Valida as credenciais
      if (email === 'admin@admin.com' && password === '1234') {
        // Redireciona para home.html
        window.location.href = 'home.html';
      } else {
        // Exibe mensagem de erro
        alert('E-mail ou senha incorretos. Tente novamente.');
      }
    });

    document.querySelector('.google').addEventListener('click', function(e) {
      e.preventDefault(); // Impede o link real
      alert('Login com Google simulado com sucesso!');
      window.location.href = 'home.html'; // Redireciona para home (mock)
    });
  </script>
</body>
</html>
```

#html/recuperar-senha.html

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Simulador de Concursos - Recuperar Senha</title>
  <link rel="stylesheet" href="../css/styles.css">
```

```

</head>
<body>
  <div class="container">
    <h1>Recuperar Senha</h1>
    <p>Insira seu e-mail para receber um link de recuperação.</p>
    <form id="recuperarForm">
      <input type="email" id="email" placeholder="E-mail" required>
      <button type="submit">Enviar</button>
      <a href="index.html" class="link">Voltar ao login</a>
    </form>
    <p id="mensagem" style="color: red; display: none;"></p>
  </div>

  <script>
    // Captura o formulário
    const form = document.getElementById('recuperarForm');
    const mensagem = document.getElementById('mensagem');

    // Adiciona evento de submit ao formulário
    form.addEventListener('submit', function(event) {
      event.preventDefault(); // Impede o envio padrão do formulário

      // Captura o valor do e-mail
      const email = document.getElementById('email').value;

      // Valida o e-mail
      if (email === 'admin@admin.com') {
        mensagem.style.color = 'green';
        mensagem.style.display = 'block';
        mensagem.textContent = 'Link de recuperação enviado para seu e-
mail.';
      } else {
        mensagem.style.color = 'red';
        mensagem.style.display = 'block';
        mensagem.textContent = 'E-mail não encontrado. Verifique e tente
novamente.';
      }
    });
  </script>
</body>
</html>

```

#html/relatorios.html

```

<!DOCTYPE html>
<html lang="pt-BR">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Simulador de Concursos - Relatórios</title>
  <link rel="stylesheet" href="../css/styles.css">
  <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
</head>
<body>
  <!-- dados mockados ref desempenho nos simulados -->
  <div class="container">
    <h1>Histórico de Simulados</h1>
    <div class="dashboard">
      <div class="results-table">
        <table>
          <tr>
            <th>Data</th>
            <th>Questões</th>
            <th>Acertos</th>
            <th>Tempo</th>
          </tr>
          <tr>
            <td>10/05/2025</td>
            <td>10</td>

```

```

            <td>80%</td>
            <td>15:32</td>
        </tr>
        <tr>
            <td>09/05/2025</td>
            <td>20</td>
            <td>70%</td>
            <td>25:10</td>
        </tr>
        <tr>
            <td>08/05/2025</td>
            <td>20</td>
            <td>20%</td>
            <td>22:22</td>
        </tr>
    </table>
</div>
<div class="chart-container">
    <canvas id="progressChart" width="150" height="100"></canvas>
</div>
</div>
<button>Excluir Simulado</button>
<button>Zerar Histórico</button>
<a href="home.html" class="link">Voltar</a>
</div>
<script>
    const ctx = document.getElementById('progressChart').getContext('2d');
    const progressChart = new Chart(ctx, {
        type: 'bar', // Grafico de Barras
        data: {
            labels: ['08/05', '09/05', '10/05'],
            datasets: [{
                label: 'Progresso (%)',
                data: [20, 70, 80], // Baseado na tabela: 10%, 70%, 80%
                backgroundColor: '#008000',
                borderColor: '#008000',
                borderWidth: 1
            }]
        },
        options: {
            scales: {
                x: { title: { display: false } },
                y: { beginAtZero: true, max: 100, ticks: { stepSize: 20 } }
            },
            plugins: {
                title: { display: true, text: 'Progresso', font: { size: 10 } },
                legend: { display: false },
                annotation: {
                    annotations: [{
                        type: 'label',
                        content: 'Crescente',
                        xValue: '10/05/2025',
                        yValue: 80,
                        backgroundColor: 'rgba(0, 0, 0, 0.5)',
                        font: { size: 8 }
                    }]
                }
            },
            maintainAspectRatio: false
        }
    });
</script>
</body>
</html>

```

#html/resultado-simulado.html

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Simulador de Concursos - Resultado</title>
  <link rel="stylesheet" href="../css/styles.css">
</head>
<body>
  <div class="container">
    <h1>Resultado do Simulado</h1>
    <p>Acertos: 8/10 (80%)</p>
    <p>Erros: 2/10 (20%)</p>
    <p>Tempo Total: 15:32</p>
    <p>Tempo Médio por Questão: 1:33</p>
    <a href="relatorios.html" class="button">Ver Histórico</a>
    <a href="home.html" class="button">Novo Simulado</a>
  </div>
</body>
</html>
```

#html/simulado.html

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Simulador de Concursos - Simulado</title>
  <link rel="stylesheet" href="../css/styles.css">
</head>
<body>
  <div class="container">
    <h2>Simulado - Questão 1/10</h2>
    <div class="cronometro">00:00</div>
    <p>Qual é a linguagem de programação mais usada em 2025?</p>
    <div class="alternativa">A) Python</div>
    <div class="alternativa">B) Java</div>
    <div class="alternativa">C) C++</div>
    <div class="alternativa">D) JavaScript</div>
    <a href="resultado-simulado.html" class="button">Finalizar Simulado</a>
    <a href="home.html" class="link">Cancelar</a>
  </div>
</body>
</html>
```