



INSTITUTO FEDERAL GOIANO
CAMPUS TRINDADE
ENGENHARIA DA COMPUTAÇÃO

TATIANA YUKARI SEKIYA IABAGATA

**APLICAÇÃO DE FERRAMENTAS BASEADAS EM INTELIGÊNCIA ARTIFICIAL
PARA MINIMIZAÇÃO DE DESAFIOS E OTIMIZAÇÃO DE PROCESSOS NA
ENGENHARIA DE SOFTWARE**

Trindade

2025

TATIANA YUKARI SEKIYA IABAGATA

**APLICAÇÃO DE FERRAMENTAS BASEADAS EM INTELIGÊNCIA ARTIFICIAL
PARA MINIMIZAÇÃO DE DESAFIOS E OTIMIZAÇÃO DE PROCESSOS NA
ENGENHARIA DE SOFTWARE**

Trabalho de Conclusão de Curso apresentado
ao curso de Engenharia da Computação do
Instituto Federal Goiano Campus Trindade,
como parte da exigência para obtenção do título
de bacharel em Engenharia da Computação.

Orientador(a): Dra. Jaqueline Alves Ribeiro

Trindade

2025

**Ficha de identificação da obra elaborada pelo autor, através do
Programa de Geração Automática do Sistema Integrado de Bibliotecas do IF Goiano - SIBi**

I11a Iabagata, Tatiana Yukari Sekiya
APLICAÇÃO DE FERRAMENTAS BASEADAS EM
INTELIGÊNCIA ARTIFICIAL PARA MINIMIZAÇÃO DE
DESAFIOS E OTIMIZAÇÃO DE PROCESSOS NA
ENGENHARIA DE SOFTWARE / Tatiana Yukari Sekiya
Iabagata. Trindade 2025.

41f. il.

Orientadora: Prof^a. Dra. Jaqueline Alves Ribeiro.
Tcc (Bacharel) - Instituto Federal Goiano, curso de 0820303 -
Bacharelado em Engenharia de Computação - Trindade (Campus
Trindade).

1. Engenharia de software. 2. Inteligência artificial. 3.
Engenharia de requisitos. 4. Prototipagem. 5. Engenharia de
prompt. I. Título.



SERVIÇO PÚBLICO FEDERAL
MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÉNCIA E TECNOLOGIA GOIANO

Documentos 3/2026 - CE-TRI/GE-TRI/CMPTRI/IFGOIANO

Repositório Institucional do IF Goiano - RIIF Goiano

Sistema Integrado de Bibliotecas

**TERMO DE CIÊNCIA E DE AUTORIZAÇÃO PARA DISPONIBILIZAR PRODUÇÕES TÉCNICO-CIENTÍFICAS NO
REPOSITÓRIO INSTITUCIONAL DO IF GOIANO**

Com base no disposto na Lei Federal nº 9.610/98, AUTORIZO o Instituto Federal de Educação, Ciência e Tecnologia Goiano, a disponibilizar gratuitamente o documento no Repositório Institucional do IF Goiano (RIIF Goiano), sem ressarcimento de direitos autorais, conforme permissão assinada abaixo, em formato digital para fins de leitura, download e impressão, a título de divulgação da produção técnico-científica no IF Goiano.

Identificação da Produção Técnico-Científica

- | | |
|--|---|
| <input type="checkbox"/> Tese | <input type="checkbox"/> Artigo Científico |
| <input type="checkbox"/> Dissertação | <input type="checkbox"/> Capítulo de Livro |
| <input type="checkbox"/> Monografia – Especialização | <input type="checkbox"/> Livro |
| <input checked="" type="checkbox"/> TCC - Graduação | <input type="checkbox"/> Trabalho Apresentado em Evento |
| <input type="checkbox"/> Produto Técnico e Educacional - Tipo: _____ | |

Nome Completo do Autor: Tatiana Yukari Sekiya Iabagata

Matrícula: 2020108203030261

Título do Trabalho: Aplicação de Ferramentas Baseadas em Inteligência Artificial para Minimização de Desafios e Otimização de Processos na Engenharia de Software

Restrições de Acesso ao Documento

Documento confidencial: Não Sim, justifique: _____

Informe a data que poderá ser disponibilizado no RIIF Goiano: //

O documento está sujeito a registro de patente? Sim Não

O documento pode vir a ser publicado como livro? Sim Não

O/A referido/a autor/a declara que:

1. o documento é seu trabalho original, detém os direitos autorais da produção técnico-científica e não infringe os direitos de qualquer outra pessoa ou entidade;
2. obteve autorização de quaisquer materiais inclusos no documento do qual não detém os direitos de autor/a, para conceder ao Instituto Federal de Educação, Ciência e Tecnologia Goiano os direitos requeridos e que este material cujos direitos autorais são de terceiros, estão claramente identificados e reconhecidos no texto ou conteúdo do documento entregue;
3. cumpriu quaisquer obrigações exigidas por contrato ou acordo, caso o documento entregue seja baseado em trabalho financiado ou apoiado por outra instituição que não o Instituto Federal de Educação, Ciência e Tecnologia Goiano.

Trindade, 27/01/2026.

Tatiana Yukari Sekiya labagata

Assinatura do Autor e/ou Detentor dos Direitos Autorais

Ciente e de acordo:

Jaqueleine Alves Ribeiro

Assinatura do(a) orientador(a)

Documento assinado eletronicamente por:

- **Jaqueleine Alves Ribeiro, PROFESSOR ENS BASICO TECN TECNOLOGICO**, em 27/01/2026 14:30:34.
- **Tatiana Yukari Sekiya labagata, 2020108203030261 - Discente**, em 27/01/2026 14:32:36.

Este documento foi emitido pelo SUAP em 27/01/2026. Para comprovar sua autenticidade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifgoiano.edu.br/autenticar-documento/> e forneça os dados abaixo:

Código Verificador: 782331

Código de Autenticação: d6d0d392ab



INSTITUTO FEDERAL GOIANO
Campus Trindade
Av. Wilton Monteiro da Rocha, S/N, Setor Cristina II, TRINDADE / GO, CEP 75389-269
(62) 3506-8000



SERVIÇO PÚBLICO FEDERAL
MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÉNCIA E TECNOLOGIA GOIANO

Ata nº 50/2025 - CE-TRI/GE-TRI/CMPTRI/IFGOIANO

ATA DE DEFESA DE TRABALHO DE CURSO

Aos 10 dias do mês de dezembro de 2025, às 20 horas e 45 minutos, reuniu-se a banca examinadora composta pelos docentes: Jaqueline Alves Ribeiro (orientadora), Adson Silva Rocha (membro), Renato de Sousa Gomide (membro), para examinar o Trabalho de Curso intitulado **“Aplicação de ferramentas baseadas em Inteligência Artificial para minimização de desafios e otimização de processos na Engenharia de Software”** da estudante **Tatiana Yukari Sekiya labagata**, Matrícula nº **2020108203030261** do Curso de Bacharelado em Engenharia de Computação do IF Goiano – Campus Trindade. A palavra foi concedida ao estudante para a apresentação oral do TC, houve arguição do candidato pelos membros da banca examinadora. Após tal etapa, a banca examinadora decidiu pela **APROVAÇÃO COM RESSALVAS** da estudante com nota 7,5. Ao final da sessão pública de defesa foi lavrada a presente ata que segue assinada pelos membros da Banca Examinadora.

(Assinado Eletronicamente)

Jaqueline Alves Ribeiro

Orientadora

(Assinado Eletronicamente)

Adson Silva Rocha

Membro

(Assinado Eletronicamente)

Renato de Sousa Gomide

Membro

Documento assinado eletronicamente por:

- **Jaqueline Alves Ribeiro, PROFESSOR ENS BASICO TECN TECNOLOGICO**, em 10/12/2025 21:59:55.
- **Adson Silva Rocha, PROFESSOR ENS BASICO TECN TECNOLOGICO**, em 10/12/2025 22:00:58.
- **Renato de Sousa Gomide, PROFESSOR ENS BASICO TECN TECNOLOGICO**, em 10/12/2025 22:01:04.

Este documento foi emitido pelo SUAP em 10/12/2025. Para comprovar sua autenticidade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifgoiano.edu.br/autenticar-documento/> e forneça os dados abaixo:

Código Verificador: 773328

Código de Autenticação: 1c009552c5



INSTITUTO FEDERAL GOIANO

Campus Trindade

Av. Wilton Monteiro da Rocha, S/N, Setor Cristina II, TRINDADE / GO, CEP 75389-269

(62) 3506-8000

Dedico este trabalho à minha família, aos meus amigos e ao meu companheiro de vida. Sem vocês, nada disso seria possível.

AGRADECIMENTOS

Aos meus pais, pelo apoio incondicional e por todo o esforço que dedicaram ao longo dessa jornada para que eu pudesse chegar até aqui.

À minha irmã, Juliana, que sempre apoiou minhas escolhas e esteve ao meu lado em qualquer circunstância.

Ao meu grande companheiro de vida, Matheus, que sempre buscou o melhor para me apoiar e acreditou em mim, mesmo quando eu mesma duvidava.

Aos meus amigos, que compreenderam minhas ausências e cansaço, e me apoiaram em cada etapa desse caminho.

À minha orientadora, Jaqueline, que acreditou no projeto desde o início, ofereceu força e incentivo para sua concretização e sempre se mostrou aberta a ouvir novas ideias e orientar com dedicação.

RESUMO

A Engenharia de Software enfrenta desafios significativos nas fases iniciais do desenvolvimento, especialmente na definição de requisitos e na criação de protótipos, o que frequentemente resulta em retrabalho, inconsistências e aumento de custos. Diante desse cenário, este trabalho tem como objetivo analisar as dificuldades presentes nessas etapas e propor soluções baseadas em ferramentas de Inteligência Artificial, com destaque para o NotebookLM e o Gemini Canvas. A pesquisa adotou uma abordagem aplicada, iniciando com uma investigação teórica sobre requisitos, prototipagem e IA, seguida da análise das funcionalidades das ferramentas e, por fim, da simulação de um cenário fictício de um Sistema de Gestão de Condomínios. Foram elaborados arquivos estruturados que serviram como base de conhecimento, possibilitando a aplicação de técnicas de engenharia de prompt para validação, organização e geração de requisitos, além da criação de protótipos funcionais. Os resultados demonstraram que o NotebookLM contribui para reduzir ambiguidades e agilizar a análise de requisitos, enquanto o Gemini Canvas facilita a prototipação do sistema. Conclui-se que o uso dessas ferramentas pode minimizar falhas comuns nas etapas iniciais do desenvolvimento de software, tornando o processo mais eficiente, preciso e alinhado às necessidades do projeto.

Palavras-chave: engenharia de software; inteligência artificial; engenharia de requisitos; prototipagem; engenharia de prompt.

ABSTRACT

Software Engineering faces significant challenges in the initial development phases, especially in requirements definition and prototyping, which frequently results in rework, inconsistencies, and increased costs. Given this scenario, this work aims to analyze the difficulties present in these stages and propose solutions based on Artificial Intelligence tools, highlighting NotebookLM and Gemini Canvas. The research adopted an applied approach, beginning with a theoretical investigation into requirements, prototyping, and AI, followed by an analysis of the tools' functionalities, and finally, the simulation of a fictitious scenario of a Condominium Management System. Structured files were created to serve as a knowledge base, enabling the application of prompt engineering techniques for the validation, organization, and generation of requirements, as well as the creation of functional prototypes. The results demonstrated that NotebookLM contributes to reducing ambiguities and streamlining requirements analysis, while Gemini Canvas facilitates system prototyping. The study concludes that the use of these tools can minimize common failures in the initial stages of software development, making the process more efficient, precise, and aligned with the project's needs.

Keywords: software engineering; artificial intelligence; requirements engineering; prototyping; prompt engineering.

SUMÁRIO

1. INTRODUÇÃO	8
2. PROBLEMATIZAÇÃO	9
3. OBJETIVOS	10
3.1 Objetivo geral.....	10
3.2 Objetivos específicos.....	10
4. JUSTIFICATIVA	11
5. REVISÃO DE LITERATURA	12
5.1 Engenharia de Software.....	12
5.2 Requisitos	12
5.2.1 Classificação dos requisitos.....	12
5.3 Engenharia de requisitos.....	13
5.3.1 Processo da Engenharia de Requisitos	13
5.4 Definição de Protótipo.....	14
5.4.1 Processo de desenvolvimento de protótipo	14
5.5 Inteligência Artificial.....	15
5.5.1 Inteligência Artificial na Engenharia de Software.....	16
5.6 Engenharia de Prompt	17
6. METODOLOGIA.....	18
6.1 Organização das Etapas da Pesquisa.....	18
6.1.1 Etapa 1 - Investigação Teórica e Identificação das Dificuldades	18
6.1.2 Etapa 2 - Análise das Ferramentas NotebookLM e Gemini Canvas	18
6.1.3 Etapa 3 - Demonstração da Aplicabilidade em Cenários Simulados	20
7. RESULTADOS E DISCUSSÃO.....	22
7.1 Visão Geral dos Cenários Simulados	22
7.2 Criação dos Prompts	23
7.2.1 Prompts criados para o cenário de Requisitos	23
7.2.1.1 Validação de Existência de Requisito Funcional.....	23
7.2.1.2 Validação de Não Existência de Requisito Funcional	23
7.2.1.3 Avaliação da Melhoria "Relatório de Performance do Porteiro"	24
7.2.1.4 Documentar um Requisito Funcional (RF).....	24

7.2.1.5 Pesquisa de Detalhes Específicos de um Requisito	25
7.2.2 Prompts criado para o cenário de prototipagem	25
7.2.2.1 Protótipo de tela de aterrissagem (login)	25
7.2.2.2 Protótipo do painel de controle principal (dashboard) que o porteiro visualizará após realizar o login	26
7.3 Resultados dos prompts	27
7.3.1 Resultados do Prompts criados para o cenário de Requisitos.....	28
7.3.1.1 Validação de Existência de Requisito Funcional.....	28
7.3.1.2 Validação de Não Existência de Requisito Funcional	28
7.3.1.3 Avaliação da Melhoria "Relatório de Performance do Porteiro"	28
7.3.1.4 Documentar um Requisito Funcional (RF).....	29
7.3.1.5 Pesquisa de Detalhes Específicos de um Requisito	29
7.3.2 Resultados do Prompts criados para o cenário de Protótipos	29
8. CONCLUSÃO.....	31
REFERÊNCIAS	33
APÊNDICE A – DOCUMENTOS DESENVOLVIDOS.....	35

1. INTRODUÇÃO

A engenharia de software reúne processos, métodos e ferramentas que tornam viável o desenvolvimento de sistemas computacionais complexos, assegurando que sejam entregues com qualidade e dentro dos prazos estabelecidos (Pressman; Maxim, 2016). No entanto, a prática da Engenharia de Software ainda enfrenta desafios, sendo a definição de requisitos e elaboração de protótipos, que são fundamentais para garantir que o produto não apenas funcione, mas atenda às expectativas dos usuários.

A prototipagem, de acordo com Sommerville (2011), constitui uma etapa fundamental nas fases iniciais do desenvolvimento de software, contribuindo para a validação dos requisitos e para a compreensão das funcionalidades do sistema.

A fase de levantamento de requisitos é uma das principais causas de falhas em projetos de software, que podem resultar em sistemas entregues com atraso, com custos acima do previsto ou que não atendem adequadamente às necessidades dos usuários. Além disso, alterações nos requisitos após sua aprovação costumam ser dispendiosas, e mal-entendidos entre clientes, analistas e desenvolvedores contribuem ainda mais para os problemas (Sommerville; Sawyer, 1997).

Diante desse cenário, fica evidente que esses desafios afetam a qualidade final do software. A aplicação da Inteligência Artificial na engenharia de software mostra-se promissora para otimizar processos e minimizar tais desafios. Pois, permite aprimorar processos criativos por meio do uso de algoritmos com capacidade de autoajuste, capazes de assumir tarefas relacionadas à organização e otimização. Ao proporcionar que máquinas percebam, raciocinem e interpretem informações, a IA amplia as estratégias adotadas nessa área, contribuindo diretamente para diferentes etapas da engenharia de software, como análise, projeto, implementação, testes, manutenção (Barenkamp; Rebstadt; Thomas, 2020).

Este trabalho tem como objetivo geral analisar os desafios enfrentados na Engenharia de Software, com ênfase na definição de requisitos e na criação de protótipos, propondo soluções por meio do uso de ferramentas baseadas em inteligência artificial, como o NotebookLM e o Gemini Canvas.

Para que essas ferramentas sejam eficazes para minimizar dificuldades e otimizar o processo de desenvolvimento, torna-se essencial o uso estratégico da engenharia de prompt, segundo Nascimento (2024), corresponde à arte e ciência de elaborar instruções precisas, estruturadas e orientadas ao contexto, capazes de guiar sistemas de IA generativa a produzir respostas mais confiáveis, completas e alinhadas às necessidades do usuário.

2. PROBLEMATIZAÇÃO

Apesar dos avanços tecnológicos, a Engenharia de Software ainda enfrenta obstáculos na fase de levantamento de requisitos, identificada como uma das principais causas de insucesso em projetos. Essa etapa, quando mal executada, resulta em sistemas entregues com atraso, custos que excedem o planejamento original ou que não atendem adequadamente às necessidades dos usuários. Alterações tardias nos requisitos costumam ser dispendiosas, e falhas de comunicação entre clientes, analistas e desenvolvedores agravam o cenário. A correção de problemas identificados em fases posteriores implica custos consideravelmente mais elevados (Sommerville; Sawyer, 1997).

Essas dificuldades estendem-se à prototipagem, técnica essencial para validar e refinar as necessidades do sistema. A problemática reside na ausência de objetivos claramente definidos, o que gera o risco de gestores e usuários finais não compreenderem a finalidade do protótipo, frustrando as expectativas e limitando os benefícios dessa ferramenta, conforme Sommerville (2011). Combinados, os problemas na especificação de requisitos e na prototipagem elevam o retrabalho e comprometem diretamente a qualidade final e os prazos de entrega do software.

3. OBJETIVOS

3.1 Objetivo geral

Analisar os desafios da Engenharia de Software nas fases de definição de requisitos e de criação de protótipos, propondo soluções com ferramentas de inteligência artificial que minimizem dificuldades e otimizem o processo.

3.2 Objetivos específicos

- Identificar as principais dificuldades na definição de requisitos e criação de protótipo.
- Explorar as ferramentas e as suas funcionalidades.
- Demonstrar como essas ferramentas baseadas em Inteligência Artificial podem ser aplicadas em cenários simulados da Engenharia de Software.

4. JUSTIFICATIVA

A realização deste estudo justifica-se pelo impacto que decisões tomadas nas fases iniciais do desenvolvimento exercem sobre a qualidade final do software. Investigar abordagens que contribuam para a melhoria dessas etapas torna-se relevante, especialmente diante da recorrência de falhas relacionadas à definição de requisitos e à prototipagem em projetos de Engenharia de Software.

Diante disso, propõe-se a exploração de ferramentas baseadas em Inteligência Artificial, como o NotebookLM e o Gemini Canvas. O objetivo é demonstrar a aplicabilidade dessas tecnologias no apoio à especificação de requisitos e à elaboração de protótipos.

Os resultados esperados têm potencial para orientar a reformulação de práticas convencionais, apresentando novas possibilidades de abordagem que tornem o processo mais ágil, preciso e alinhado às exigências atuais do setor de tecnologia. Visto que essas tecnologias oferecem potencial para reduzir ambiguidades, automatizar tarefas repetitivas e aprimorar a análise inicial dos projetos, contribuindo diretamente para a estruturação mais sólida e eficiente do processo de desenvolvimento.

5. REVISÃO DE LITERATURA

Esta seção apresenta os conceitos que sustentam o estudo sobre a aplicação da Inteligência Artificial na Engenharia de Software, abordando os conceitos da própria engenharia de software, da engenharia de requisitos e da Inteligência Artificial.

5.1 Engenharia de Software

A engenharia de software busca oferecer ao desenvolvimento profissional de softwares a aplicação de técnicas voltadas para a especificação, o projeto e a evolução de programas. Ademais, busca adotar uma abordagem sistemática para o desenvolvimento do software, visto que considera analisar os custos, os prazos e a confiabilidade. Ao mesmo tempo procura atender às necessidades dos clientes e dos profissionais envolvidos no processo de desenvolvimento (Sommerville, 2011).

5.2 Requisitos

Os requisitos para Sommerville e Sawyer (1997) são definidos nas etapas iniciais do desenvolvimento do software como uma especificação do que é preciso implementar, incluindo tanto o seu comportamento quanto suas propriedades. Além de estabelecer possíveis restrições ao processo de desenvolvimento da solução.

5.2.1 Classificação dos requisitos

Os requisitos de software devem ser organizados em duas categorias: funcionais e não funcionais, conforme Turine e Masiero (1996). Os funcionais definem as necessidades que o software precisa executar, enquanto os não funcionais impõem condições que orientam a maneira como essas necessidades devem ser implementadas, segundo Sommerville e Sawyer (1997).

A partir dessa distinção, comprehende-se com mais clareza que os requisitos funcionais descrevem os comportamentos que o sistema deve realizar, podendo variar conforme o tipo de software, o perfil dos usuários e a abordagem adotada na definição das necessidades. Já os não funcionais referem-se a características que não estão diretamente ligadas aos serviços prestados pelo sistema, mas sim a propriedades globais, como desempenho, confiabilidade e consumo de recursos (Sommerville, 2011).

5.3 Engenharia de requisitos

Segundo Menezes et al (2015, p.45):

O levantamento ideal das necessidades e operacionalidades de um software passa a ser um elo entre o projeto e a construção do software. Assim, surge o conceito de Engenharia de Requisitos (ER) que representa um conjunto de ferramentas e procedimentos, que visam identificar as reais necessidades de um projeto, estabelecendo as características que o software deve apresentar para a aceitação por parte do cliente.

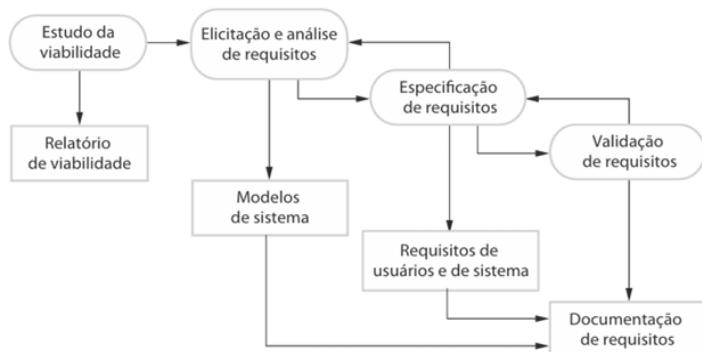
Chaves e Aguiar (2016) afirmam que a Engenharia de Requisitos faz parte do campo da Engenharia de Software e tem como propósito principal gerar um conjunto de requisitos que represente, de forma coerente, completa e relevante, aquilo que realmente é esperado pelo cliente.

A Engenharia de Requisitos é considerada uma das áreas mais desafiadoras do desenvolvimento de sistemas e uma das mais importantes dentro da Engenharia de Software (Hasan; Loucopoulos; Nikolaïdou, 2014 apud Chaves; Aguiar, 2016).

5.3.1 Processo da Engenharia de Requisitos

O processo de engenharia de requisitos tem como finalidade a produção de um documento que descreva um sistema capaz de atender às necessidades identificadas. Esse processo é estruturado em quatro atividades centrais, conforme descrito por Sommerville (2011) e representado na Figura 1. As etapas ilustradas na figura são detalhadas a seguir, com base na explicação do autor.

Figura 1 - Processo de engenharia de requisitos.



Fonte: Sommerville (2011)

O estudo de viabilidade tem como intuito avaliar se as necessidades do usuário podem ser atendidas por meio das tecnologias de software e hardware disponíveis. Essa análise também leva em conta as limitações orçamentárias do projeto, verificando se o desenvolvimento é viável dentro do contexto proposto. Ao final, espera-se um resultado que indique, com base em dados concretos, se o prosseguimento do desenvolvimento é recomendável Sommerville (2011).

A etapa de elicitação e análise de requisitos envolve a obtenção das informações necessárias ao sistema por meio da observação de soluções já existentes, da coleta de dados junto aos usuários e os responsáveis pela aquisição, além da avaliação das tarefas envolvidas Sommerville (2011). Essa fase pode incluir a criação de protótipos e modelos que auxiliem na visualização e compreensão do que será desenvolvido.

A especificação de requisitos consiste em transformar as informações coletadas na fase de análise em um documento formal que descreve o conjunto de necessidades do sistema. Nesse documento, podem ser incluídos dois tipos de requisitos: os do usuário, que apresentam uma visão mais geral voltada ao cliente e ao usuário final, e os requisitos do sistema, que detalham com mais precisão as funcionalidades que deverão ser implementadas Sommerville (2011).

A etapa de validação tem como objetivo analisar se os requisitos definidos são realistas, coerentes e suficientemente completos. Durante essa verificação, é comum que inconsistências ou falhas sejam identificadas no documento, o que exige revisões para garantir que ele esteja adequado às necessidades do projeto Sommerville (2011).

No processo de engenharia de requisitos, as atividades não ocorrem de forma estritamente sequencial. A análise se estende durante as etapas de definição e especificação, e é comum que novos requisitos surjam ao longo do desenvolvimento. Por isso, essas atividades costumam ser realizadas de maneira intercalada.

5.4 Definição de Protótipo

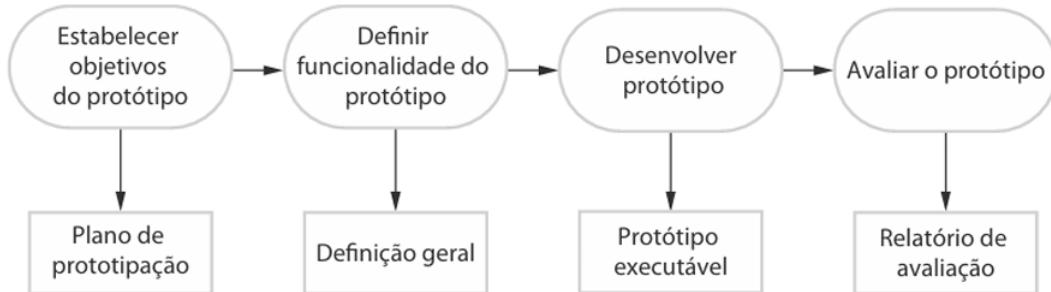
Sommerville (2011) explica que o protótipo tem como objetivo apoiar a elicitação e a validação de requisitos, permitir a visualização antecipada do sistema e reduzir ambiguidades entre desenvolvedores e usuários. Além disso, possibilita a identificação precoce de inconsistências, contribuindo para a diminuição de retrabalho e de custos nas etapas posteriores do desenvolvimento.

5.4.1 Processo de desenvolvimento de protótipo

Ao desenvolver um protótipo, é fundamental que seus objetivos estejam claros desde o início do processo. Pois dependendo da finalidade o protótipo deve ser planejado de forma

específica. Um único modelo não é capaz de atender a todos os objetivos simultaneamente. Caso esses propósitos não sejam definidos previamente, há o risco de gestores e usuários não compreenderem sua real utilidade, o que pode gerar expectativas equivocadas quanto aos resultados obtidos (Sommerville, 2011). O processo é apresentado na Figura 2.

Figura 2 - Processo de desenvolvimento de protótipo.



Fonte: Sommerville (2011)

Os objetivos da prototipação precisam estar claramente definidos desde o início do processo, pois um único protótipo não é capaz de atender a todas as finalidades ao mesmo tempo. Quando esses objetivos não são estabelecidos, há o risco de que a gerência ou os usuários finais não compreendam corretamente a função do protótipo, o que pode resultar na frustração de expectativas e na limitação dos benefícios esperados com seu desenvolvimento.

A próxima etapa do processo consiste em definir quais funcionalidades devem ser incluídas no sistema de protótipo e, mais importante ainda, quais podem ser deixadas de fora. A exclusão de determinadas funções pode ser uma estratégia eficaz para reduzir os custos envolvidos e agilizar o prazo de entrega.

A etapa final do processo corresponde à avaliação do protótipo. Nessa fase, é necessário prever o treinamento dos usuários e utilizar os objetivos definidos anteriormente como base para a elaboração de um plano de avaliação adequado.

5.5 Inteligência Artificial

A máquina é capaz de simular comportamentos inteligentes sempre que recebe comandos claros e bem definidos, atuando conforme as instruções programadas para reproduzir a lógica esperada (McCarthy, 1955). Nesse contexto, Barbosa e Portes (2019) destacam que a Inteligência Artificial pode ser compreendida como a habilidade de sistemas eletrônicos em simular o pensamento humano, atuando de forma autônoma na percepção de variáveis, tomada de decisões e resolução de problemas. Em essência, trata-se do desenvolvimento de tecnologias

capazes de reproduzir, de maneira lógica, a capacidade humana de raciocinar e agir de forma inteligente.

5.5.1 Inteligência Artificial na Engenharia de Software

Barenkamp, Rebstadt e Thomas (2020), observam que a aplicação da Inteligência Artificial (IA) na engenharia de software em um primeiro momento, pode parecer contraditória. Visto que a IA trata de automatizar as atividades por meio de máquinas inteligentes, enquanto a engenharia de software é uma atividade criativa e geralmente, baseada no conhecimento humano. Portanto, uma análise mais aprofundada revela que essas tecnologias podem atuar de forma complementar.

Os processos de engenharia de software podem ser aprimorados pelo suporte de algoritmos autoajustáveis, responsáveis por tarefas de organização e otimização. Nesse sentido, a Inteligência Artificial – compreendida como a capacidade das máquinas de perceber, raciocinar e interpretar informações – expande as abordagens e estratégias aplicadas a esta área, contribuindo de maneira eficaz (Barenkamp; Rebstadt; Thomas, 2020).

Nesse cenário, destacam-se duas ferramentas baseadas em Inteligência Artificial que têm como objetivo oferecer suporte às atividades da engenharia de software, atuando especialmente na organização, análise e tratamento de grandes volumes de informação.

Uma dessas soluções é NotebookLM, um assistente baseado em Inteligência Artificial que aprende com as fontes inseridas, contribuindo para a memorização de fatos, a conexão entre ideias e a síntese de informações, tornando-se um recurso eficaz para lidar com grandes volumes de dados. Pois é capaz de extrair informações relevantes, identificar temas principais e gerar guias a partir de conteúdos fornecidos. Ao receber os arquivos, a ferramenta cria automaticamente um guia de fontes com resumos, destaques dos tópicos centrais e sugestões de perguntas para aprofundamento (Suarez; Brown, 2025).

Conforme Maulana et al. (2025), o Gemini Canvas demonstra elevada eficácia ao apoiar a organização de ideias, a formulação de conceitos e a estruturação de conteúdos de maneira mais aprofundada, graças à sua interação colaborativa com a inteligência artificial. Ao reduzir obstáculos iniciais no processo de criação e desenvolvimento, a ferramenta possibilita análises mais consistentes e refinadas, evidenciando seu potencial para elevar a qualidade dos resultados produzidos.

As ferramentas baseadas em Inteligência Artificial analisadas neste estudo, como o NotebookLM e o Gemini Canvas, são tecnologias recentes e ainda em processo de consolidação acadêmica, o que resulta, até o momento, na escassez de estudos científicos. Diante desse

contexto, parte significativa das informações disponíveis encontra-se em documentos técnicos, blogs e materiais oficiais disponibilizados pelos próprios desenvolvedores, utilizados de forma complementar à literatura científica consolidada que discute, de maneira mais ampla, a aplicação da Inteligência Artificial na Engenharia de Software.

5.6 Engenharia de Prompt

O crescimento das ferramentas baseadas em inteligência artificial ampliou seu uso na criação de textos, imagens e outros materiais de maneira prática e criativa. Entretanto, solicitações simples nem sempre resultam em respostas satisfatórias, o que torna essencial formular instruções mais claras e detalhadas (Nascimento, 2024).

Nesse cenário, surge a engenharia de prompt, que se dedica a desenvolver comandos capazes de orientar essas ferramentas a produzirem resultados mais precisos e alinhados aos objetivos do usuário, compreendendo um conjunto de técnicas que direciona a geração de conteúdo de forma eficiente e específica (Nascimento, 2024).

A prática com a engenharia de prompt significa organizar o pedido de maneira estratégica, escolhendo bem as expressões e moldando a pergunta para que ela se alinhe ao propósito e ao cenário em que será usada (Santos; Martins; Evangelista, 2024).

6. METODOLOGIA

Este projeto adota uma abordagem de estudo exploratório com o objetivo de compreender os desafios recorrentes nas fases iniciais do desenvolvimento de software, especialmente na definição de requisitos e na criação de protótipos, bem como demonstrar a aplicação de ferramentas baseadas em Inteligência Artificial na mitigação desses desafios.

Nesse contexto, o estudo considera a engenharia de prompt como um elemento essencial para elevar a precisão das interações com as ferramentas baseadas em IA, dado que, conforme Nascimento (2024), instruções bem elaboradas permitem direcionar modelos de linguagem a gerar respostas mais adequadas ao contexto e ao objetivo do usuário.

6.1 Organização das Etapas da Pesquisa

A pesquisa foi estruturada em três etapas, conforme os objetivos específicos.

6.1.1 Etapa 1 - Investigaçāo Teórica e Identificação das Dificuldades

A primeira etapa consistiu em uma investigação teórica e exploratória, realizada por meio da análise de livros e artigos científicos. O objetivo foi identificar dificuldades recorrentes nos processos de definição de requisitos e prototipagem, tais como falhas de comunicação, ambiguidades durante a elicitação e limitações na construção de modelos iniciais do sistema. Essa revisão permitiu compreender o contexto em que as ferramentas de IA seriam aplicadas e forneceu a base necessária para as análises posteriores.

6.1.2 Etapa 2 - Análise das Ferramentas NotebookLM e Gemini Canvas

Na segunda etapa, realizou-se a análise das ferramentas com a finalidade de compreender suas funcionalidades, capacidades e se atendem aos critérios estabelecidos para superar os desafios identificados na etapa anterior.

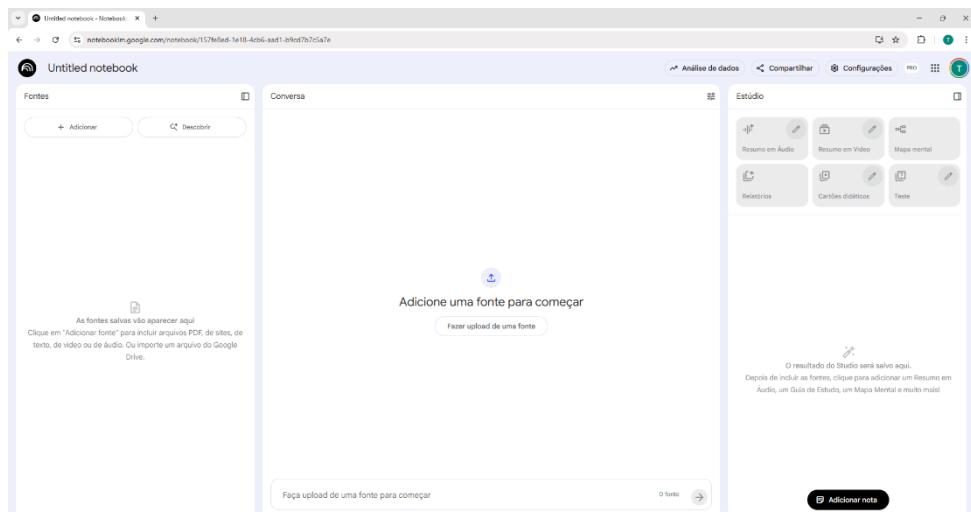
Para selecionar a ferramenta mais adequada para atuar nos desafios relacionados à definição de requisitos foram estabelecidos os seguintes critérios:

- Organização das informações;
- Confiabilidade dos resultados gerados;
- Interação com os conteúdos fornecidos pelo usuário;
- Facilidade operacional para o usuário.

Por meio desses critérios, o NotebookLM foi escolhido devido à sua capacidade de síntese e centralização de informações, personalização e suporte colaborativo. Possibilita ao usuário interagir com seus materiais através de prompts e trabalhar exclusivamente com as fontes fornecidas pelo próprio usuário. Com isso, a ferramenta reduz o risco de imprecisões e promove maior confiabilidade nas informações geradas (Reyna, 2024).

Além de possuir recurso de ajuste do estilo das respostas permitindo alinhar a apresentação das informações às necessidades da equipe, desde análises mais detalhadas até sínteses rápidas de pesquisas complexas. A possibilidade de criação de cadernos compartilhados também se destaca, pois facilita o armazenamento organizado de materiais, agiliza processos como o onboarding e contribui para manter a equipe alinhada às diretrizes do projeto (Schaefer, 2025). A ferramenta é apresentada na Figura 3.

Figura 3 – Ferramenta NotebookLm



Fonte: Autoria própria.

Visando à escolha de uma ferramenta capaz de apoiar a criação de protótipos, foram definidos os seguintes critérios:

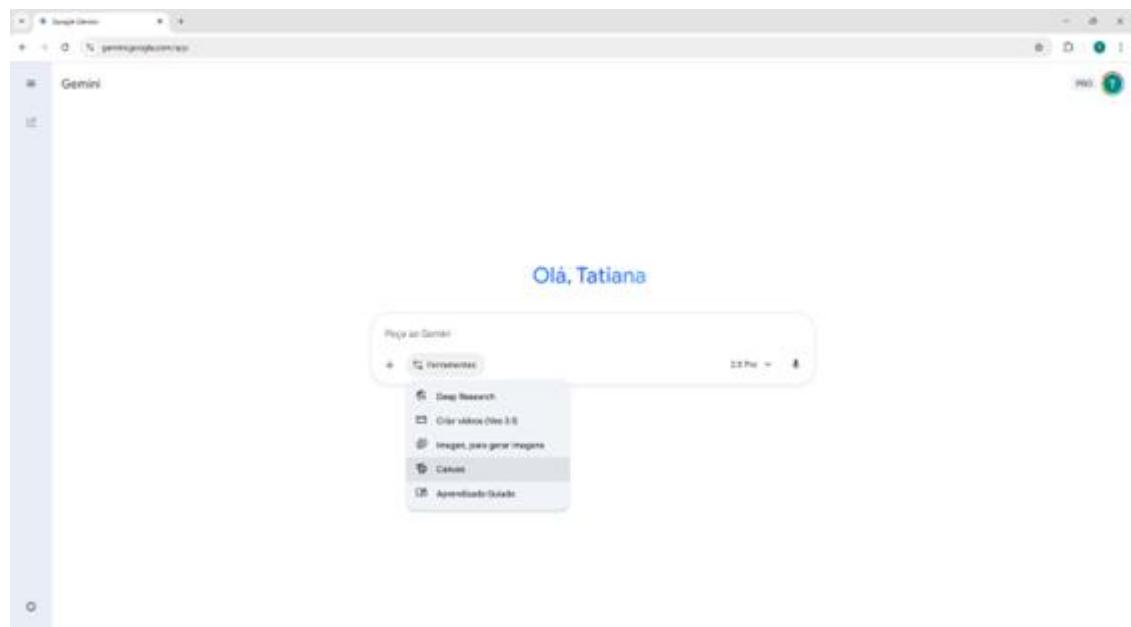
- Agilidade na transformação de ideias em protótipos funcionais;
- Edição e visualização do protótipo em tempo real;
- Possibilitasse a geração de protótipos a partir de prompts;
- Facilidade operacional para o usuário.

Considerando esses critérios, optou-se pelo Gemini Canvas. Visto que é um ambiente interativo integrado ao Gemini que agiliza o processo de transformar ideias de codificação em protótipos funcionais, permitindo criar, editar e visualizar código em tempo real. Ele oferece

suporte ao desenvolvimento de aplicações web e possibilita a exportação do conteúdo para o Google Docs quando necessário, favorecendo um fluxo contínuo de aprimoramento do código. O recurso está sendo disponibilizado globalmente para assinantes do Gemini e Gemini Advanced em todos os idiomas em que os aplicativos do Gemini estão disponíveis (Citron, 2025).

Encontra-se disponível globalmente para assinantes do Gemini e do Gemini Advanced, em todos os idiomas suportados pelas plataformas. Diante dessa ampla acessibilidade e das funcionalidades que oferece, a ferramenta mostra-se adequada ao desenvolvimento das etapas previstas neste projeto, especialmente por atender às demandas relacionadas ao uso de IA no apoio à construção e manipulação de protótipos. A ferramenta é apresentada na Figura 4.

Figura 4 – Ferramenta Gemini Canvas



Fonte: Autoria própria.

As ferramentas selecionadas, além de atenderem os critérios estabelecidos apresentam outros benefícios e usabilidades que otimizam e minimizam os desafios nos processos da engenharia de software.

6.1.3 Etapa 3 - Demonstração da Aplicabilidade em Cenários Simulados

Na terceira etapa, elaborou-se um cenário fictício detalhado referente a uma empresa desenvolvedora de um sistema de gestão de condomínios. Para viabilizar a simulação, foram criados arquivos (simulando a documentação de um projeto real) que seriam utilizados como base de conhecimento para as ferramentas de IA.

A elaboração dos documentos ocorreu de forma sequencial e estruturada, simulando as etapas iniciais dos processos da Engenharia de Software. Inicialmente, definiu-se o um documento denominado Empresa com o objetivo de contextualizar o domínio do problema, caracterizar o produto e estabelecer o cenário organizacional. Em seguida, foi construído o documento Escopo, delimitando as funcionalidades, plataformas envolvidas e fronteiras do sistema, garantindo alinhamento entre as expectativas do projeto e os objetivos do produto.

A partir desse contexto, foram elaborados os documentos de Regras de Negócio, Requisitos Funcionais e Requisitos Não Funcionais, de modo a detalhar o comportamento esperado do sistema e suas restrições. Posteriormente, os Casos de Uso foram definidos com base nos requisitos levantados, descrevendo as interações entre atores e sistema de forma passo a passo. Ademais, foram criados arquivos de apoio, a fim de orientar a atuação das ferramentas baseadas em Inteligência Artificial.

Os arquivos do cenário foram carregados NotebookLM, foi incluído também um arquivo de contexto específico, instruindo a ferramenta a comportar-se como um Analista de Requisitos. Com isso, o NotebookLM foi configurado como um repositório centralizado e um assistente de IA focado na minimização e otimização de problemas relacionados a requisitos. O usuário pode, então, utilizar a Engenharia de Prompt para organizar, sintetizar e estruturar as informações (incluindo requisitos funcionais, não funcionais e regras de negócio), bem como validar a existência de requisitos, avaliar melhorias, classificar comportamentos não documentados e definir novas funcionalidades.

O Gemini Canvas foi preparado especificamente para a etapa de prototipagem. A ferramenta foi alimentada com as mesmas fontes do cenário fictício através da funcionalidade "Gems", que atua como um especialista de IA personalizado. Com o "Gem" já ciente do contexto do sistema, o usuário pôde focar em prompts para a geração do protótipo, detalhando o objetivo do sistema, a visão de cada tipo de usuário e os elementos de tela esperados. Essa abordagem permitiu representar visualmente as funcionalidades e os fluxos do sistema, contribuindo para uma compreensão mais clara das interações previstas.

7. RESULTADOS E DISCUSSÃO

Esta seção apresenta os resultados obtidos na aplicação das ferramentas NotebookLM, Gemini Canvas e das técnicas de engenharia de prompt em cenários simulados de uma empresa de gestão condominial.

7.1 Visão Geral dos Cenários Simulados

O cenário fictício desenvolvido para a pesquisa foi o de um Sistema de Gestão de Condomínios (SGC). Este sistema é concebido como uma plataforma web e mobile, focada especificamente no controle de acesso de moradores, visitantes e portaria.

O cenário fictício desenvolvido para a pesquisa refere-se a um Sistema de Gestão de Condomínios (SGC). Este sistema é concebido como uma plataforma web e mobile, focada especificamente no controle de acesso de moradores, visitantes e da portaria.

Para simular a documentação do projeto e estruturar a base de conhecimento, foi elaborado um conjunto de documentos, cuja as descrições são apresentadas a seguir. Sendo o detalhamento completo disponibilizado no (Apêndice A).

- Empresa: Apresenta o cenário fictício da empresa e seu produto, o Sistema de Gestão de Condomínios (SGC);
- Escopo: Detalha a abrangência do SGC, definindo-o como uma plataforma web e mobile para controle de acesso;
- Regras de Negócio (RN): Estabelece as políticas, lógicas e restrições operacionais específicas do sistema;
- Requisitos Funcionais (RF): Descreve as funcionalidades e comportamentos que o sistema deve executar;
- Requisitos Não Funcionais (RNF): Define os atributos de qualidade e restrições de implementação do sistema;
- Casos de Uso (UC): Descreve as interações passo a passo entre os atores (usuários, como morador ou porteiro) e o sistema (SGC) para alcançar um objetivo específico, detalhando o fluxo principal e os fluxos alternativos de eventos;
- Analista de Requisitos: Arquivo de contexto que define o papel e as responsabilidades de um analista, utilizado para instruir a IA;
- Definições: Fornece a base teórica sobre Engenharia de Requisitos, detalhando o que são requisitos e a distinção entre funcionais e não funcionais.

7.2 Criação dos Prompts

Dado que as ferramentas empregadas utilizam a Inteligência Artificial Gemini, a elaboração dos comandos seguiu as estratégias recomendadas pela documentação da API Gemini, focando na criação de prompts em linguagem natural para extrair respostas precisas e de alta qualidade.

Para isso, foram aplicadas diversas técnicas, como o fornecimento de instruções claras e específicas, a adição de contexto relevante para que o modelo tivesse as informações necessárias, a especificação de restrições e a definição do formato esperado da resposta, incluindo o uso de prefixos de saída, conforme orientado na documentação (Google, 2025). Foram criados os seguintes prompts:

7.2.1 Prompts criados para o cenário de Requisitos

7.2.1.1 Validação de Existência de Requisito Funcional

Verifique se existe um requisito funcional que obriga o sistema a validar tanto o reconhecimento facial quanto os dados pessoais do visitante antes de liberar o acesso.

Restrições: Baseie sua resposta unicamente nos Requisitos Funcionais (RF) documentados.

Formato da Resposta: [Sim/Não]

Justificativa: [Citação do RF correspondente]

7.2.1.2 Validação de Não Existência de Requisito Funcional

Determine se o sistema possui um requisito funcional que permita aos moradores agendarem visitas recorrentes de forma automática (por exemplo, toda sexta-feira) para um mesmo visitante.

Restrições: Sua resposta deve se basear estritamente no que está descrito nos documentos. A pré-aprovação de visitantes frequentes não é o mesmo que o agendamento automático.

Formato da Resposta: [Sim/Não]

Justificativa: [explique a ausência de um RF específico para esta funcionalidade, citando o requisito mais próximo, se houver, e detalhando a diferença.]

7.2.1.3 Avaliação da Melhoria "Relatório de Performance do Porteiro"

Análise a seguinte solicitação de melhoria: "Criar um relatório gerencial que mede a performance dos porteiros, incluindo métricas como o tempo médio para realizar um cadastro manual e o tempo médio para liberar um visitante pré-cadastrado. Avalie se este tipo de relatório é consistente com as capacidades de auditoria e relatórios já previstas.

Restrições: Fundamente sua análise apenas nos requisitos funcionais e escopo de relatórios e auditoria já documentados.

Formato da Resposta: Adequação: [Sim/Não]

Justificativa: [explique se a melhoria é uma extensão natural dos relatórios existentes, citando os requisitos relevantes que já preveem a coleta de dados necessários.]

Sugestão de Inclusão (se aplicável): [indique o arquivo e o requisito onde esta nova métrica de relatório poderia ser detalhada.]

7.2.1.4 Documentar um Requisito Funcional (RF)

Comportamento não documentado: "Após 15 minutos de inatividade na tela da portaria, o sistema automaticamente encerra a sessão do porteiro e exige uma nova autenticação com MFA para reativá-la."

Classifique este comportamento e escreva um novo Requisito Funcional (RF) para descrevê-lo. O novo requisito deve seguir o padrão e a numeração sequencial dos requisitos existentes.

Restrições: A descrição deve ser clara, concisa e focar no que o sistema *faz*. A numeração deve continuar a partir do último RF documentado.

Formato da Resposta:

Classificação: [Requisito Funcional]

Justificativa: [explique por que é um RF (descreve uma ação/função específica do sistema).]

Requisito Proposto: [escreva o novo RF no formato: "RF-XX (novo): [Descrição do requisito]"]

Justificativa: [Explique por que é um RNF (define um critério de qualidade/performance, o "como" o sistema opera).]

Requisito Proposto: [escreva o novo RNF no formato: "RNF-XX (novo): [Descrição do requisito]"]

7.2.1.5 Pesquisa de Detalhes Específicos de um Requisito

Localize e liste todos os requisitos funcionais (RF) que descrevem o processo de verificação do visitante na portaria, incluindo reconhecimento facial, validação de dados e o que acontece se o visitante não tiver um pré-cadastro.

Restrições: Sua resposta deve citar apenas os documentos de Requisitos Funcionais.

Formato da Resposta:

Requisitos Funcionais Encontrados: [liste os RFs com a descrição completa.]

7.2.2 Prompts criado para o cenário de prototipagem

7.2.2.1 Protótipo de tela de aterrissagem (login)

Crie um protótipo de tela de aterrissagem (login), contido em um único arquivo HTML. Esta tela é destinada exclusivamente ao acesso do porteiro de um condomínio a um sistema de software.

Contexto Essencial: O software chama-se "SGC - Sistema de Gestão de Condomínios". Conforme os requisitos, a visão do porteiro é acessada exclusivamente via plataforma desktop (web). O acesso é restrito a funcionários previamente cadastrados e requer autenticação multifator (MFA) após a inserção das credenciais iniciais. O objetivo é apresentar uma tela profissional, que reforce a segurança e a seriedade do sistema.

Requisitos Chave da Interface

Layout Geral: A tela deve ter um layout moderno de duas colunas, otimizado para visualização em desktop.

Coluna Esquerda (Branding):

Deve conter o nome do sistema ("SGC") e um slogan ou descrição curta, como "Plataforma de gestão de acesso para condomínios".

Inclua um elemento visual, como um ícone de escudo ou um placeholder de imagem que remeta à segurança.

Use um fundo com cor escura para criar contraste.

Coluna Direita (Formulário):

Deve ter um título claro, como "Acesso Restrito à Portaria".

O formulário de login deve conter os campos: "Usuário" (ou "ID de funcionário") e "Senha".

Inclua ícones representativos dentro de cada campo de input.

Adicione uma opção "Lembrar-me" (checkbox) e um link para "Esqueceu a senha?".

O botão de ação principal deve ser "Entrar".

Importante: Abaixo do botão "Entrar", adicione uma nota de texto pequena e clara informando ao usuário que um passo de autenticação multifator (MFA) será exigido após o login.

Estilo e Aparência:

Utilize a família de fontes 'Inter'.

O design deve ser limpo, com bom espaçamento e cantos arredondados.

A paleta de cores deve ser profissional, sugerindo tons de cinza, azul ou verde para os elementos de destaque

7.2.2.2 Protótipo do painel de controle principal (dashboard) que o porteiro visualizará após realizar o login

Crie um protótipo do painel de controle principal (dashboard) que o porteiro visualizará após realizar o login. O protótipo deve ser um único arquivo HTML e projetado para ser usado em um monitor de desktop na portaria.

Contexto Essencial: Esta é a tela principal e a ferramenta de trabalho diária do porteiro. Ela deve fornecer uma visão geral clara e imediata de todas as visitas agendadas para o dia, além de permitir ações rápidas como registrar visitantes não esperados e localizar cadastros existentes. O foco é a eficiência operacional e a clareza da informação para agilizar o processo de check-in e garantir a segurança, conforme os requisitos RF-09, RF-10 e o caso de uso UC-04.

Requisitos Chave da Interface

Layout Geral: Um layout de dashboard clássico para desktop, com um cabeçalho fixo, uma barra de navegação lateral e uma área de conteúdo principal.

Cabeçalho (Header):

Deve exibir o logo/nome do sistema "SGC".

No lado direito, deve mostrar o nome do usuário logado (ex: "Porteiro Carlos Silva"), a data e a hora atuais, e um botão de "Sair".

Navegação Lateral (Sidebar):

Deve ter links de navegação claros com ícones. O item ativo deve ser "Painel de Controle". Outros links podem ser incluídos como placeholders (ex: "Histórico de Acessos", "Relatórios").

Área de Conteúdo Principal:

Título: Um título grande e claro, como "Controle de Acessos | [Data Atual]".

Ações Rápidas: Uma seção no topo com:

Um botão proeminente: "+ Registrar Visita Manual" (RF-10, UC-05).

Uma barra de busca funcional para "Localizar visitante por Nome, CPF ou Placa" (UC-04).

Fila de Visitas do Dia (RF-09): O componente mais importante da tela.

Deve ser uma tabela clara e legível com as visitas do dia.

Colunas da Tabela: Status, Nome do Visitante, Destino (Torre/Apto), Horário Previsto, Placa (se houver), e Ações.

Status: Use tags coloridas para identificar visualmente o status de cada visita (ex: Aguardando Chegada em verde, Em Visita em azul, finalizada em cinza, Pendente em amarelo).

Ações: Cada linha deve ter um botão de ação principal, como "Verificar Entrada", que levaria à tela de validação facial e de dados.

Componente de Alertas (RF-21): Uma pequena seção, talvez acima da tabela, para exibir notificações importantes (ex: "Alerta: Foto de baixa qualidade para visitante X", "Inconsistência de documento para visitante Y").

Estilo e Aparência:

Utilize a família de fontes 'Inter'.

O design deve ser profissional, limpo e com alta legibilidade, apropriado para um ambiente de trabalho.

Use uma paleta de cores sóbria (cinzas, azuis) com cores de destaque para status e alertas

7.3 Resultados dos prompts

A aplicação dos prompts no cenário do Sistema de Gestão de Condomínios (SGC) permitiu testar, na prática, a capacidade das ferramentas NotebookLM e Gemini Canvas em lidar com tarefas da Engenharia de Software. Os resultados mostraram que a engenharia de prompt, usada corretamente, aumentou a precisão e a organização nas etapas de definição de requisitos e prototipagem.

No NotebookLM, os prompts de validação de requisitos tiveram um desempenho consistente. A ferramenta consultou exclusivamente os documentos do projeto para confirmar a existência (ou ausência) de requisitos funcionais e não funcionais, justificando as respostas com base na própria documentação. Isso reforça o NotebookLM como um repositório confiável para a análise de requisitos, reduzindo ambiguidades e ajudando a identificar falhas no escopo.

A ferramenta também conseguiu classificar comportamentos não documentados e avaliar melhorias, agindo de acordo com o papel de "Analista de Requisitos" que lhe foi

atribuído pelo arquivo de contexto. A engenharia de prompt foi essencial para guiar a IA e garantir que seguisse as restrições e justificasse suas respostas.

No Gemini Canvas, os prompts de prototipagem ajudaram a ferramenta a estruturar telas e fluxos de acordo com os perfis de usuário do SGC. Os protótipos gerados foram coerentes com o cenário, pois a ferramenta foi previamente contextualizada, resultando em representações visuais alinhadas aos objetivos do projeto.

7.3.1 Resultados do Prompts criados para o cenário de Requisitos

7.3.1.1 Validação de Existência de Requisito Funcional

Resposta: Sim

Justificativa: O requisito funcional RF-11 exige que, no momento do acesso, o sistema verifique o reconhecimento facial contra o template cadastrado e valide os dados pessoais. O acesso é liberado somente se ambos existirem e estiverem válidos.

Localização: Requisitos Funcionais (RF), item RF-11

Fonte: "Requisito Funcional.txt"

7.3.1.2 Validação de Não Existência de Requisito Funcional

Resposta: Não

Justificativa: Não existe um Requisito Funcional que define o agendamento automático de visitas recorrentes em um padrão temporal (e.g., toda sexta-feira).

O requisito mais próximo é o RF-16, que permite configurar políticas de aprovação automática para visitas recorrentes ou frequentes.

Esta política trata da condição de liberação/aprovação, e não da funcionalidade de o morador criar uma agenda recorrente de forma automática no sistema.

Localização: Requisitos Funcionais (RF)

7.3.1.3 Avaliação da Melhoria "Relatório de Performance do Porteiro"

Adequação: Sim

Justificativa: A melhoria é uma extensão natural dos relatórios e auditoria existentes. O sistema já prevê a coleta de dados, fornecendo a base para calcular a performance do porteiro.

Sugestão de Inclusão (se aplicável): Esta nova métrica de relatório poderia ser detalhada como uma especificação adicional ou subitem dentro do Requisito Funcional RF-22, no arquivo "Requisito Funcional.txt".

Fontes: Requisito Funcional.txt

7.3.1.4 Documentar um Requisito Funcional (RF)

Classificação: Requisito Funcional

Justificativa: Descreve uma ação (função) específica do sistema relacionada à segurança e à gestão de sessão (encerramento automático), que define o que o sistema deve fazer.

Requisito Proposto: RF-31 (novo): O sistema deve encerrar automaticamente a sessão da portaria após 15 minutos de inatividade, exigindo nova autenticação multifator (MFA) para reativá-la.

Localização Sugerida: "Requisito Funcional.txt", sequência dos itens RF-27 a RF-30 (novos).

7.3.1.5 Pesquisa de Detalhes Específicos de um Requisito

Requisitos Funcionais Encontrados: RF-10: Quando o visitante chegar sem pré-cadastro, o porteiros deve cadastrar manualmente dados pessoais e destino; o sistema contata o morador para liberação (impedimento original).

RF-11: No momento do acesso, o sistema deve verificar reconhecimento facial contra o template cadastrado e validar dados pessoais; acesso só é liberado se ambos existirem e estiverem válidos.

RF-12: Se houver placa cadastrada e LPR disponível, validar a placa nas cancelas de visitantes e abrir automaticamente quando permitido.

RF-13: Permitir a emissão de QR Code temporário como alternativa de autenticação (ex.: leitor no totem).

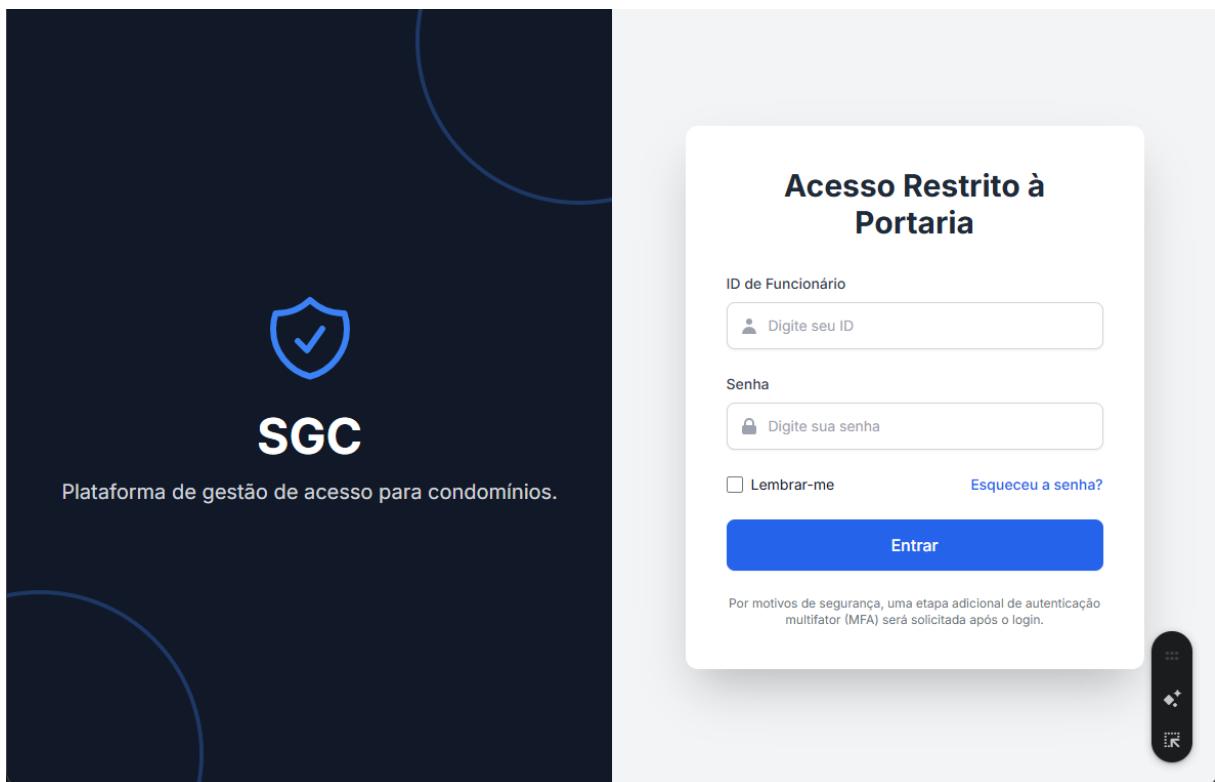
RF-18: Bloquear entradas fora da janela autorizada ou sem confirmação facial/dados pessoais.

Fontes: Requisito Funcional.txt

7.3.2 Resultados do Prompts criados para o cenário de Protótipos

Resultado do prompt para a geração de protótipo para a tela de aterrissagem (login) na Figura 5.

Figura 5 - Protótipo de tela de aterrissagem (login)



Fonte: Autoria própria.

Na Figura 6 abaixo, o protótipo criado do painel de controle principal (dashboard) que o porteiro visualizará após realizar o login.

Figura 6 - Protótipo do painel de controle principal

STATUS	NOME DO VISITANTE	DESTINO	HORÁRIO PREVISTO	PLACA
Aguardando Chegada	Ana Clara Souza ***123.456-**	Torre A, Apto 101	14:00	ABC-1234
Em Visita	Bruno Martins (Técnico) ***789.012-**	Torre C, Apto 1104	(Chegou 13:45)	---
Pendente (Morador)	Rafael Lima ***555.666-**	Torre B, Apto 505	15:00	---
Finalizada	Entrega iFood N/A	Torre A, Apto 101	(Saída 13:30)	MOTO-XYZ

Fonte: Autoria própria.

8. CONCLUSÃO

A análise desenvolvida ao longo deste trabalho permitiu aprofundar a compreensão de que a Engenharia de Software ainda enfrenta desafios significativos nas etapas iniciais do desenvolvimento, especialmente na definição de requisitos e na criação de protótipos. Constatou-se que essas fases são determinantes para o sucesso do projeto, e que falhas nesse estágio podem resultar em atrasos, custos elevados, retrabalho e divergências de entendimento entre usuários e desenvolvedores. Diante disso, este estudo demonstrou que a incorporação de ferramentas baseadas em Inteligência Artificial pode contribuir para minimizar tais dificuldades, proporcionando maior precisão, organização e eficiência ao processo.

A aplicação prática do NotebookLM evidenciou o seu potencial como um repositório centralizado de informações, capaz de sintetizar conteúdos, identificar inconsistências e apoiar a análise de requisitos. O uso da engenharia de prompt mostrou-se fundamental nessa etapa, pois permitiu orientar o modelo para interpretar corretamente o cenário, validar requisitos, classificar comportamentos do sistema e responder com precisão a partir das fontes fornecidas. Dessa forma, o NotebookLM não se limitou a atuar como uma ferramenta de consulta, desempenhando também o papel de assistente nas funções designadas.

De forma complementar, o Gemini Canvas demonstrou notável eficácia na etapa de prototipagem. A partir da engenharia de prompt aplicada ao “Gems” configurado com o contexto do projeto, foi possível orientar a geração de elementos visuais, estruturar protótipos alinhados ao objetivo do sistema e representar funcionalidades e fluxos de maneira clara. Essa abordagem facilitou a validação dos requisitos e promoveu uma compreensão mais precisa das interações previstas na interface do usuário.

Entretanto, é importante destacar que os resultados obtidos se baseiam em um cenário fictício, o que constitui uma limitação da abordagem adotada. A ausência de aplicação direta em projetos reais impede uma avaliação mais abrangente sobre fatores como resistência da equipe, integração com processos organizacionais existentes e impactos no cronograma e nos custos em ambientes produtivos. Além disso, o desempenho das ferramentas está diretamente relacionado à qualidade dos dados fornecidos e à elaboração adequada dos prompts, o que exige conhecimento prévio e atenção por parte dos profissionais envolvidos.

Portanto, recomenda-se que equipes em cenários reais utilizem essas ferramentas como suporte às atividades de requisitos e prototipagem, e não como substitutas do trabalho humano. É importante que analistas e desenvolvedores mantenham uma postura crítica, validando as informações geradas pelas ferramentas baseadas em Inteligência Artificial e combinando-as

com sua experiência técnica. A definição de diretrizes claras para o uso das ferramentas e a capacitação das equipes em engenharia de prompt também se mostram práticas relevantes para maximizar os benefícios da abordagem.

Evidencia-se, portanto, que a utilização estratégica de ferramentas de Inteligência Artificial, potencializadas pela engenharia de prompt, apresenta-se como uma alternativa viável e eficiente para minimizar desafios recorrentes na Engenharia de Software. Para trabalhos futuros, sugere-se a realização de experimentos em projetos reais, de modo a ampliar o entendimento sobre seu impacto e validar sua aplicação nas etapas iniciais do desenvolvimento de sistemas.

REFERÊNCIAS

- BARBOSA, L. M.; PORTES, L. A. F. A Inteligência Artificial. **Revista Tecnologia Educacional**, Rio de Janeiro, v. 1, n. 236, p. 16-27, jan. 2023. Trimestral. Disponível em: https://abt-br.org.br/wp-content/uploads/2023/03/RTE_236.pdf. Acesso em: 06 jun. 2025.
- BARENKAMP, M.; REBSTADT, J.; THOMAS, O. Applications of AI in classical software engineering. **Ai Perspectives**, [S.L.], v. 2, n. 1, p. 1-15, 26 jul. 2020. Springer Science and Business Media LLC. <http://dx.doi.org/10.1186/s42467-020-00005-4>
- CHAVES, I. G.; AGUIAR, Y. P. C. PRODUCT EXPERIENCE E REQUISITOS NÃO FUNCIONAIS: um framework de design e sua contribuição à engenharia de requisitos. **Blucher Design Proceedings**, [S.L.], p. 1382-1394, dez. 2016. Editora Blucher. <http://dx.doi.org/10.5151/despro-ped2016-0117>.
- CITRON, D. New ways to collaborate and get creative with Gemini. **The Keyword**, 18 mar. 2025. Disponível em: <https://blog.google/products/gemini/gemini-collaboration-features/>. Acesso em: 06 jun. 2025
- COSTA, E. C. da. IMPORTÂNCIA DA ENGENHARIA DE REQUISITOS NO PROCESSO DE DESENVOLVIMENTO DE SISTEMAS DE INFORMAÇÃO. **Revista Interface Tecnológica**, [S.L.], v. 15, n. 1, p. 203-214, 30 jun. 2018. Interface Tecnologica. <http://dx.doi.org/10.31510/infa.v15i1.322>.
- MAULANA, V. A. H.; SUBIYANTO, S.; IMRON, I; SAMSUDIN, A.; JUWANDA. K. A. Analysis of Programming and Artificial Intelligence Based on Gemini Canvas in Supporting Deep Learning: A Structural Equation Modelling (SEM) Approach. **ASEAN Artificial Intelligence Journal**, v. 3, n. 1, p. 28-48, 2025.
- MCCARTHY, J. MINSKY, M. L.; ROCHESTER, N.; SHANNON, C. E. A proposal for the dartmouth summer research project on artificial intelligence, august 31, 1955. **AI magazine**, v. 27, n. 4, p. 12-12, 2006.
- MENEZES, P. M.; CORREIA, T. de A.; COUTO, T.; SANTANA, L. A Engenharia de Requisitos: um caso de implementação de um sistema para engenharia de requisitos. **Interfaces Científicas - Exatas e Tecnológicas**, [S.L.], v. 1, n. 3, p. 43-54, 17 out. 2015. Universidade Tiradentes. <http://dx.doi.org/10.17564/2359-4942.2015v1n3p43-54>.
- NASCIMENTO, J. R. do. Exploração de técnicas de engenharia de prompt para aprimorar os resultados do uso de LLM no TCMRIO. 2024.
- PRESSMAN, R. S.; MAXIM, B. R. **Engenharia de software: uma abordagem profissional**. 9. ed. Porto Alegre: Amgh, 2021.
- REYNA, J. **The potential of Google NotebookLM for teaching and learning**. In: ELEARN – WORLD CONFERENCE ON E-LEARNING, 2025, Bangkok. *Proceedings...* Bangkok: Association for the Advancement of Computing in Education (AACE), 2025.

SANTOS, G.; MARTINS, J.; EVANGELISTA, G. Prompt Engineering com ChatGPT no contexto acadêmico de IHC: uma revisão rápida da literatura. **Simpósio Brasileiro sobre Fatores Humanos em Sistemas Computacionais (IHC)**, p. 144-148, 2024.

SCHAEFER, K. 5 ways to use NotebookLM Plus for your business. **The Keyword, 2 Google Labs Blog**, [s.l.], 2025. Disponível em: <https://blog.google/technology/google-labs/notebooklm-business-tips/>. Acesso em: 06 jun. 2025.

SOMMERVILLE, I. **Engenharia de Software**. 9. ed. São Paulo: Pearson, 2011.

SOMMERVILLE, I.; SAWYER, P. Requirements engineering: a good practice guide. [S.I]: John Wiley & Sons, 1997.

SAUREZ, D.; BROWN, V. NotebookLM: revolutionizing learning for students with neurodiverse challenges using AI and universal design principles. Boca Raton: Florida Atlantic University, 2025.

APÊNDICE A – DOCUMENTOS DESENVOLVIDOS

EMPRESA

O sistema (“SGC”) é uma plataforma web e mobile para administração de condomínios residenciais, com módulos para moradores, visitantes, portaria/controle de acesso. Abrange cadastro antecipado de visitas via link, checagens de identidade, reconhecimento facial e, quando aplicável, cadastro de placa para uso de vagas de visitantes.

ESCOPO

Administração do Condomínio: cadastro de torres, andares, apartamentos, áreas comuns e regras de visitantes;

Perfis de usuário, permissões e auditoria: gestão de moradores, convites de acesso (app/web), gestão de dependentes e prestadores, geração de links de visita e acompanhamento de aprovações.

Gestão de Portaria: painel de portaria com fila de visitas do dia, pendências de aprovação, checagem facial e liberação de acesso (pedestre e veículo).

Gestão de Visitantes: formulário de pré-cadastro via link (dados pessoais + destino), upload de selfie/foto para reconhecimento facial e, opcionalmente, cadastro de placa.

Controle de Acesso: integração com câmeras IP e motor de reconhecimento facial, integração com LPR (Leitura de Placa) quando existir, emissão de QR Code temporário e registro de entrada/saída.

Comunicação e Notificações: notificações push/SMS/e-mail ao morador para aprovar/recusar, alertas à portaria sobre status de aprovação.

Relatórios e Auditoria: relatórios de visitas, tentativas negadas, tempos de aprovação, ocupação de vagas de visitantes, LGPD, segurança e conformidade, consentimento, finalidade e retenção de dados, incluindo biometria facial.

REGRAS DE NEGÓCIO (RN)

RN-01: Acesso só é permitido se o visitante tiver dados pessoais completos e biometria facial cadastrada e conferida, e a visita estiver aprovada dentro da janela de tempo.

RN-02: A placa do veículo é opcional e só concede acesso a vaga de visitantes se a regra do condomínio permitir e houver vaga disponível.

RN-03: O link de visita é intransferível e expira conforme configuração; após expirar, novo cadastro é necessário.

RN-04: Em visitas sem pré-cadastro, a portaria deve coletar dados pessoais e contatar o morador para liberação; sem resposta ou com recusa, acesso negado.

RN-05: Morador pode revogar autorização a qualquer momento antes do check-in.

RN-06: Visitantes recorrentes podem ser pré-aprovados por período determinado, com consentimento válido.

RN-07: O morador não pode acessar dados pessoais (CPF, RG, telefone, e-mail) do visitante; apenas nome e foto são exibidos para a decisão de aprovação/recusa.

RN-08: O sistema deve inibir e registrar tentativas de exportação, download ou print de fotos de visitantes na interface de aprovação, mantendo as imagens restritas ao ambiente controlado do sistema.

RN-09: Visitantes e moradores somente utilizam o aplicativo mobile (não terão acesso ao painel desktop).

RN-10: Apenas funcionários do condomínio cadastrados podem acessar a visão desktop da portaria; demais perfis não têm permissão.

REQUISITOS FUNCIONAIS (RF)

Cadastro & Identidade

RF-01: Permitir ao morador gerar um link único e temporário para cada visita, contendo torre, andar e apartamento de destino (pré-preenchidos).

RF-02: O visitante, ao abrir o link, deve preencher dados pessoais obrigatórios: nome completo, RG, CPF, telefone e e-mail.

RF-03: O visitante deve enviar uma foto (selfie) para reconhecimento facial (formato JPG/PNG; validação de qualidade mínima).

RF-04: Permitir ao visitante informar placa do veículo (opcional); o campo só aparece se o condomínio habilitar vagas de visitantes.

RF-05: Registrar a finalidade e período da visita (ex.: entrega, serviço, social; data/intervalo estimado).

Aprovação & Fluxo com o Morador

RF-06: Notificar o morador para aprovar/recusar a visita (nome + foto do visitante visíveis).

RF-07: Permitir ao morador ver/atualizar janela de tempo autorizada e número de acompanhantes.

RF-08: Em caso de recusa, registrar motivo e notificar portaria/visitante.

Portaria

RF-09: Exibir na portaria uma fila de chegadas esperadas por data/hora com status (Aguardando cadastro, aguardando aprovação, Aprovada, Expirada, Concluída, Recusada).

RF-10: Quando o visitante chegar sem pré-cadastro, o porteiro deve cadastrar manualmente dados pessoais e destino; o sistema contata o morador para liberação (impedimento original).

RF-11: No momento do acesso, o sistema deve verificar reconhecimento facial contra o template cadastrado e validar dados pessoais; acesso só é liberado se ambos existirem e estiverem válidos.

RF-12: Se houver placa cadastrada e LPR disponível, validar a placa nas cancelas de visitantes e abrir automaticamente quando permitido.

RF-13: Permitir a emissão de QR Code temporário como alternativa de autenticação (ex.: leitor no totem).

RF-14: Registrar entrada e saída (pessoa e veículo), com carimbo de data/hora, porteiro responsável e ponto de acesso.

Regras e Parâmetros

RF-15: Configurar tempo de validade do link (ex.: 24–168h) e do QR Code (ex.: 15–120 min).

RF-16: Configurar políticas de aprovação automática (ex.: visitas recorrentes, empresas de entrega previamente validadas), por condomínio.

RF-17: Habilitar/desabilitar vaga de visitantes, com limites por unidade (ex.: máx. 2 visitas simultâneas estacionadas).

RF-18: Bloquear entradas fora da janela autorizada ou sem confirmação facial/dados pessoais.

RF-19: Regras para reaproveitamento de cadastro de visitantes frequentes (com consentimento e prazo de retenção).

Comunicação

RF-20: Enviar push/SMS/e-mail ao visitante com confirmações, QR Code e instruções de chegada.

RF-21: Notificar alertas à portaria sobre inconsistências (foto de baixa qualidade, documento inválido, placa divergente).

Relatórios & Auditoria

RF-22: Relatórios por período: visitas por unidade, negadas, tempo médio de aprovação, tempo de espera na portaria, uso de vagas.

RF-23: Trilhas de auditoria completas: quem aprovou/negou, quando, qual câmera/leitor confirmou, mudanças de parâmetros.

Integrações

RF-24: Integração com motores de reconhecimento facial (SDK/serviço) e câmeras IP (ONVIF ou especificado).

RF-25: Integração com LPR (leitura de placas) e controladoras de acesso/cancelas.

RF-26: Webhooks/APIs para exportar eventos a sistemas de terceiros (administradora, segurança, BI).

RF-27: A visão do visitante será acessada exclusivamente via aplicativo mobile (link gerado abre versão responsiva ou app).

RF-28: A visão do morador será disponibilizada em aplicativo mobile, com funcionalidades de geração de links, aprovação/recusa e histórico de visitas.

RF-29: A visão do porteiro será disponibilizada em plataforma desktop (web), com painel de controle de acessos e funcionalidades de cadastro manual.

RF-30: O acesso à visão de porteiro será permitido apenas para funcionários previamente cadastrados como colaboradores do condomínio (perfil restrito).

REQUISITOS NÃO FUNCIONAIS (RNF)

Segurança & LGPD

RNF-01: Consentimento explícito para tratamento de dados pessoais e biométricos (finalidade: controle de acesso).

RNF-02: Criptografia em trânsito (TLS 1.2+) e em repouso (AES-256 para dados sensíveis; fotos biométricas separadas).

RNF-03: Segregação de dados por condomínio (multi-tenant) e controle de acesso por perfil (RBAC).

RNF-04: Retenção configurável (ex.: fotos biométricas por 12 meses, logs por 24 meses) com exclusão segura.

RNF-05: Registro e alerta de incidentes de segurança; MFA para perfis de administração.

RNF-06: O sistema deve bloquear tentativas de captura de tela (printscreens) ou download das fotos de visitantes durante o processo de aprovação/recusa pelo morador, garantindo que as imagens não sejam copiadas ou armazenadas fora do sistema.

RNF-07: O morador terá acesso apenas ao nome e foto do visitante para validação, sem exibição dos dados pessoais (CPF, RG, telefone, e-mail), que ficam restritos à portaria e administradora.

RNF-08: O acesso de porteiros será restrito por autenticação multifator e só liberado a usuários vinculados ao condomínio.

RNF-09: O sistema deve diferenciar a experiência mobile (moradores e visitantes) e a experiência desktop (porteiros), garantindo que cada perfil tenha acesso apenas ao necessário para seu papel.

CASOS DE USO (UC)

UC-01 — Gerar link de visita (Morador)

Fluxo principal: informar data/horário, destino já conhecido, limites de acompanhantes > gerar link > enviar ao visitante.

Pós-condição: link ativo e pendente de cadastro do visitante.

Exceções: limite de visitas simultâneas atingido; morador sem permissão.

UC-02 — Pré-cadastro do visitante (Visitante)

Fluxo: abrir link > aceitar termos/LGPD > preencher dados pessoais > anexar selfie > (opcional) informar placa > enviar.

Pós-condição: registro criado e aguardando aprovação do morador.

Validações: formatação CPF/RG, duplicidade básica (mesmo CPF em curto período), qualidade de foto (face detectada).

UC-03 — Aprovar/recusar visita (Morador)

Fluxo: receber notificação > visualizar nome/foto > aprovar ou recusar (com motivo) > definir janela.

Pós-condição: status atualizado; portaria notificada.

UC-04 — Check-in na portaria (Porteiro)

Fluxo: localizar visita (nome/CPF/QR) > captura facial (câmera) > motor compara com template > se aprovado e dentro da janela: liberar acesso; se veículo, validar placa e abrir cancela.

Pós-condição: entrada registrada; se aplicável, vaga marcada como ocupada.

UC-05 — Cadastro manual na portaria (Porteiro)

Fluxo: coletar dados pessoais + destino > contatar morador > registrar decisão > (opcional) capturar foto local para biometria ad hoc.

Pós-condição: acesso liberado/negado conforme resposta.

UC-06 — Check-out (Portaria/Automático)

Fluxo: registrar saída manualmente ou por evento (leitor de saída/LPR).

Pós-condição: liberar vaga e encerrar visita.

ANALISTA DE REQUISITOS

Papel: O Analista de Requisitos atua como elo entre as partes interessadas e a equipe técnica, garantindo que as necessidades do negócio sejam corretamente compreendidas, documentadas e transformadas em especificações claras para desenvolvimento.

Responsabilidades Principais:

- Levantamento de necessidades;
- Entrevistar usuários, gestores e stakeholders para identificar demandas;
- Análise e validação;
- Confirmar viabilidade técnica e aderência ao objetivo do projeto;
- Documentação;
- Redigir requisitos funcionais e não funcionais de forma clara, padronizada e rastreável;
- Gerenciamento de mudanças;
- Controlar solicitações de alteração e avaliar impactos;
- Mediação entre áreas;
- Facilitar a comunicação entre equipe de negócios e equipe de desenvolvimento;
- Apoio a testes e homologação;
- Garantir que casos de teste estejam alinhados aos requisitos;
- Acompanhamento pós-implantação;
- Coletar feedback e propor melhorias.

DEFINIÇÕES

O que são requisitos?

Requisitos são definidos durante os estágios iniciais do desenvolvimento de um sistema como uma especificação do que deve ser implementado. Eles são descrições de como o sistema deve se comportar, ou de uma propriedade ou atributo do sistema. Eles podem ser uma restrição no processo de desenvolvimento do sistema. Portanto, um requisito pode descrever:

- uma funcionalidade de nível de usuário;
- uma propriedade geral do sistema;
- uma restrição específica no sistema;
- uma restrição no desenvolvimento do sistema.
-

Portanto, os requisitos invariavelmente contêm uma mistura de informações sobre o problema, declarações de comportamento e propriedades do sistema, e restrições de projeto e fabricação.

O que é a Engenharia de Requisitos?

Engenharia de requisitos é um termo relativamente novo que foi inventado para cobrir todas as atividades envolvidas na descoberta, documentação e manutenção de um conjunto de requisitos para um sistema baseado em computador. O uso do termo "engenharia" implica que técnicas sistemáticas e repetíveis devem ser usadas para garantir que os requisitos do sistema sejam completos, consistentes, relevantes etc.

O que é um Documento de Requisitos?

O documento de requisitos é uma declaração oficial dos requisitos do sistema para clientes, usuários finais e desenvolvedores de software.

Diferença entre Requisitos Funcionais e Não Funcionais

Requisitos funcionais descrevem o que o sistema deve fazer, e requisitos não funcionais impõem restrições sobre como esses requisitos funcionais são implementados. Um requisito funcional pode afirmar que um sistema deve fornecer algum recurso para autenticar a identidade de um usuário do sistema; um requisito não funcional pode afirmar que o processo de autenticação deve ser concluído em quatro segundos ou menos.