

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE GOIÁS
ESCOLA POLITÉCNICA E DE ARTES
GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO



**APLICAÇÃO WEB INTEGRADA COM CHATGPT PARA ANÁLISE DE DADOS DE
PROJETOS BIM UTILIZANDO AUTODESK PLATFORM SERVICES**

PEDRO OLIVEIRA MONTEIRO

GOIÂNIA
2025

PEDRO OLIVEIRA MONTEIRO

**APLICAÇÃO WEB INTEGRADA COM CHATGPT PARA ANÁLISE DE
DADOS DE PROJETOS BIM UTILIZANDO AUTODESK PLATFORM SERVICES**

Trabalho de Conclusão de Curso apresentado à
Escola Politécnica e de Artes, da Pontifícia
Universidade de Goiás, como parte dos
requisitos para obtenção do título de Bacharel
em Ciência da Computação

Orientador (a): Prof. Me. Daniel Correa da Silva.

GOIÂNIA
2025

AGRADECIMENTOS

Agradeço primeiramente a Deus, pela oportunidade de chegar até aqui, pela sabedoria concedida ao longo dessa caminhada e por ter iluminado meus passos nos momentos de incerteza. Sem Sua presença e direção, esta conquista não seria possível.

Expresso minha profunda gratidão aos meus pais, que sempre acreditaram no meu potencial e me apoiaram incondicionalmente. Cada palavra de incentivo, cada gesto de cuidado e cada sacrifício feito ao longo da minha vida foram fundamentais para que eu pudesse alcançar mais essa etapa.

À minha esposa, deixo um agradecimento especial. Sua paciência, compreensão e apoio constante foram indispensáveis durante todos os dias de dedicação intensa ao trabalho e à faculdade. Obrigado por estar ao meu lado, por compreender minhas ausências e por me fortalecer mesmo nos períodos mais difíceis. Esta conquista também é sua.

Agradeço imensamente ao meu orientador, Daniel, cuja orientação me acompanha desde a iniciação científica. Seus insights tecnológicos, materiais de estudo, conselhos e dedicação foram essenciais para o desenvolvimento deste trabalho. Sua visão e comprometimento contribuíram significativamente para meu crescimento acadêmico e profissional.

Por fim, deixo meu agradecimento a todos que, de alguma forma, fizeram parte desta jornada, colegas, amigos e profissionais com quem tive a oportunidade de aprender e conviver. Cada contribuição, direta ou indireta, marcou positivamente minha trajetória.

RESUMO

Este trabalho apresenta o desenvolvimento de uma aplicação web integrada ao ChatGPT para análise de dados de projetos BIM, utilizando os recursos da Autodesk Platform Services (APS). O projeto abrange todas as etapas, desde a concepção da ideia até a implementação das funcionalidades que permitem o processamento e a interpretação automatizada de informações provenientes de modelos BIM. Durante o desenvolvimento, alguns desafios foram enfrentados, como a definição do fluxo de integração com serviços externos, o tratamento de dados complexos de projetos e a estruturação de dois fluxos distintos de inteligência artificial, um voltado à análise dos dados e outro dedicado à comunicação com o usuário via chatbot. Apesar das limitações relacionadas ao tempo, à complexidade técnica e ao domínio das tecnologias envolvidas, os resultados obtidos foram positivos. A solução desenvolvida possibilita a visualização de dados BIM, a execução de análises assistidas por IA e a interação direta com um assistente inteligente capaz de interpretar e contextualizar informações de projetos. O trabalho contribui para demonstrar o potencial da integração entre IA generativa e plataformas BIM, ampliando possibilidades no apoio à tomada de decisões, automação de processos e eficiência na gestão de projetos.

Palavras-chave: Aplicação web, BIM, Autodesk Platform Services, Inteligência Artificial, ChatGPT.

ABSTRACT

This work presents the development of a web application integrated with ChatGPT for analyzing BIM project data using Autodesk Platform Services (APS). The project covers all stages, from the conceptual design to the implementation of features that enable automated processing and interpretation of information extracted from BIM models. Throughout development, several challenges were encountered, such as defining the integration flow with external services, handling complex project data, and structuring two distinct AI workflows, one for data analysis and another for user interaction through a chatbot. Despite limitations related to time, technical complexity, and mastering the technologies involved, the results were satisfactory. The developed solution allows BIM data visualization, AI-assisted analysis, and direct interaction with an intelligent assistant capable of interpreting and contextualizing project information. This work highlights the potential of combining generative AI with BIM platforms, expanding possibilities for decision-making support, process automation, and improved project management efficiency.

Keywords: Web application, BIM, Autodesk Platform Services, Artificial Intelligence, ChatGPT.

LISTA DE FIGURAS

Figura 1 - Arquitetura Cliente-Servidor.....	18
Figura 2 - Fluxo de uma API.....	19
Figura 3 – Ilustração da modelo de DDD – Domain Driven Design.....	21
Figura 4 - Arquitetura Limpa.....	22
Figura 5 - Tipos de definição de Requisitos de um sistema.....	33
Figura 6 - Modelo conceitual.....	45
Figura 7 - Modelo Lógico.....	45
Figura 8 - Modelagem do banco de dados.....	46
Figura 9 - Diagrama de Casos de Uso do Sistema.....	47
Figura 10 - Logo do projeto.....	49
Figura 11 - Paleta de cores do projeto.....	49
Figura 12 - Protótipo Tela de Login.....	50
Figura 13 - Protótipo Tela de Inicial.....	51
Figura 14 - Protótipo Tela de dashboard.....	51
Figura 15 - Protótipo Tela de visualizar empreendimento.....	52
Figura 16 - Protótipo Tela de visualizar modelo 3D.....	52
Figura 17 - Protótipo Tela de visualizar modelo 3D com chatbot.....	53
Figura 18 - Identidade Visual.....	53
Figura 19 - Estrutura de pastas do backend.....	56
Figura 20 - Modularização e trecho de código de inversão de dependência.....	57
Figura 21 - Estrutura de pastas do frontend.....	58
Figura 22 - Organização e trecho de código da tela de empreendimentos.....	59
Figura 23 - Infraestrutura do sistema.....	59
Figura 24 - Autodesk Viewer SDK.....	62
Figura 25 - Autodesk Model Derivative API.....	63
Figura 26 - Fluxo n8n para análise dos dados.....	64
Figura 27 - Fluxo n8n para análise dos dados.....	65
Figura 28 - Tela de Login.....	65
Figura 28 – Tela de cadastro.....	66
Figura 29 – Tela Inicial.....	66
Figura 30 – Tela de cadastro de empreendimento.....	67
Figura 31 – Tela inicial com empreendimento.....	67
Figura 32 – Tela cadastro de projeto.....	68
Figura 33 – Tela de visualizar o empreendimento.....	68
Figura 34 – Tela de projetos.....	69
Figura 35 – Tela de visualização de modelos 3D.....	69
Figura 36 – Tela de visualização com modelo 3D.....	70
Figura 37 – Modal de análise de projetos.....	70

LISTA DE QUADROS

Quadro 1 - Principais verbos HTTP.....	20
Quadro 2 - Requisitos Funcionais (RF).....	36
Quadro 3 - Requisitos Não Funcionais (RNF).....	41
Quadro 4 - Atores do sistema.....	43

LISTA DE ABREVIATURAS E SIGLAS

API	Application Programming Interface
APS	Autodesk Platform Services
AWS	Amazon Web Services
BIM	Building Information Modeling
CRUD	Create, Read, Update and Delete
CSS	Cascading Style Sheets
DDD	Domain-Driven Design
ETL	Extract, Transform and Load
GPT	Generative Pre-trained Transformer
HTTP	Hypertext Transfer Protocol
IA	Inteligência Artificial
JSON	JavaScript Object Notation
JWT	JSON WEB TOKEN
N8N	Node-Node (nome da ferramenta de automação)
REST	Representational State Transfer
SDK	Software Development Kit
SQL	Structured Query Language
TI	Tecnologia da Informação
UI	User Interface
URL	Uniform Resource Locator
UX	User Experience
3D	Três dimensões

SUMÁRIO

1. INTRODUÇÃO.....	9
1.1. Contextualização.....	10
1.2. Definição do Problema.....	12
1.3. Justificativa.....	12
1.4. Objetivos.....	13
1.4.1. Objetivo Geral.....	13
1.4.2. Objetivos Específicos.....	13
1.5. Metodologia de Pesquisa.....	13
1.6. Organização do Trabalho.....	14
2. DESENVOLVIMENTO.....	16
2.1 Conceitos Fundamentais.....	16
2.1.1 Desenvolvimento Web.....	16
2.1.2 Arquitetura Cliente Servidor.....	17
2.1.3 Application Programming Interface (API).....	18
2.1.4 Representational State Transfer (REST API).....	19
2.1.5 Domain-Driven Design (DDD).....	21
2.1.6 Clean Architecture.....	22
2.2. Tecnologias e Ferramentas.....	23
2.2.1 Frontend.....	23
2.2.2 Backend.....	27
2.2.3 Banco de Dados.....	29
2.2.4 Inteligência Artificial.....	30
2.3. Trabalhos Correlatos.....	31
3. ENGENHARIA DE SOFTWARE E METODOLOGIA.....	32
3.1 Levantamento de Requisitos.....	32
3.1.1. Requisitos Funcionais (RF).....	34
3.1.2. Requisitos Não Funcionais (RNF).....	40
3.2. Atores do Sistema.....	43
4. ARQUITETURA E MODELAGEM DO SISTEMA.....	44
4.1. Arquitetura da Solução.....	44
4.2. Modelagem de Dados.....	44
4.3. Modelagem UML.....	46
4.3.1 Diagrama de Casos de Uso.....	46
4.4. Design de Interface (UI/UX).....	47
4.4.1 Identidade Visual.....	47
4.4.2 Protótipos.....	50

5. DESENVOLVIMENTO E IMPLEMENTAÇÃO.....	54
5.1. Ambiente de Desenvolvimento.....	54
5.2. Estrutura do Projeto.....	55
5.3. Implementação das Funcionalidades Principais.....	60
5.4. Segurança e Controle de Acesso.....	60
5.5. Integração com APIs Externas.....	61
6. RESULTADOS.....	65
6.1. Comparativo: Objetivos e Resultados Alcançados.....	71
7. CONSIDERAÇÕES FINAIS.....	72
7.1. Conclusão.....	72
7.2. Dificuldades Encontradas.....	73
7.3. Trabalhos Futuros.....	74
8. Referências.....	75

1. INTRODUÇÃO

Nas últimas décadas, o setor da construção civil tem passado por uma profunda transformação impulsionada pelo avanço tecnológico, o aumento da capacidade computacional e a demanda por processos mais eficientes e sustentáveis. Fatores como a pandemia da COVID-19 aceleraram a digitalização do setor, promovendo a busca por soluções inovadoras que otimizem tempo, recursos e resultados (SOUZA; FABRICIO, 2021).

Entre as tecnologias emergentes, destacam-se o Building Information Modeling (BIM) e a Inteligência Artificial (IA), que vêm sendo cada vez mais integradas aos fluxos de trabalho em arquitetura e engenharia. O BIM, ao permitir a modelagem tridimensional associada a dados ricos e parametrizados, oferece vantagens como a compatibilização de projetos, o levantamento automático de quantitativos e o trabalho colaborativo entre profissionais (ALMEIDA, 2016). Esse modelo viabiliza o alinhamento entre tempo, custo e qualidade, historicamente conquistado por meio do uso de ferramentas tecnológicas (PAIVA, 2016).

A Inteligência Artificial, por sua vez, tem ampliado sua aplicação no suporte à tomada de decisões em diversas áreas, inclusive na arquitetura. A inteligência artificial é uma ferramenta útil, mas como um elemento para ajudar o arquiteto, cuja expertise e eficiência como profissional são os fatores mais importantes para o planejamento e a execução de um projeto bem-sucedido (MORAES, 2023).

Essa integração tem o potencial de automatizar tarefas repetitivas, detectar erros de compatibilidade e oferecer sugestões proativas, contribuindo para a qualidade final do empreendimento e a redução de custos (FERREIRA et al., 2022).

Diante desse cenário, observa-se uma crescente necessidade por soluções acessíveis que combinem essas tecnologias em ambientes web, democratizando o acesso e oferecendo suporte especialmente a profissionais iniciantes.

1.1. Contextualização

O desenvolvimento do BIM teve início no final da década de 1980, quando especialistas em Tecnologia da Informação e interoperabilidade começaram a pesquisar formas de aprimorar a elaboração de projetos. Em 1987, foi lançado o primeiro software com ferramentas BIM, marcando um marco importante na evolução da tecnologia.

De acordo com Eastman et al. (2011, p.13), o BIM (Building Information Modeling) é “uma tecnologia de modelagem e um grupo associado de processos para produção, comunicação e análise do modelo de construção”. No entanto, segundo SOUZA, AMORIM e LYRIO (2009), são poucas as utilizações de ferramentas e dos processos BIM para o desenvolvimento e implantação dela.

O BIM não é apenas uma tecnologia que gera modelos em 3D, mas um processo de inovação que trabalha com várias outras dimensões de acordo com a necessidade do empreendimento. De acordo com Neil Calvert (2013), podemos as seguintes classificações de dimensões para o BIM:

- 3D – Dimensões tridimensionais
- 4D – Planejamento
- 5D – Orçamento
- 6D – Sustentabilidade
- 7D – Gestão de instalações

Segundo Vanderlei (2004), o mercado da construção civil tem se tornado cada vez mais exigente, demandando dos construtores métodos mais eficientes de planejamento, orçamentação e execução, sempre aliando redução de custos, qualidade, durabilidade e segurança.

Nesse cenário competitivo, o Building Information Modeling (BIM), em sinergia com a Inteligência Artificial (IA) e tecnologias web, surge como uma solução estratégica, oferecendo ganhos em precisão, agilidade e desempenho em todo o ciclo do projeto. O BIM permite a modelagem inteligente de dados, enquanto a inteligência artificial (IA) automatiza análises preditivas (como detecção de incompatibilidades orçamentárias ou falhas estruturais), e as plataformas web facilitam o acesso a essa ferramenta. 14

No entanto, apesar desses avanços, a implementação dessas inovações ainda enfrenta resistência em setores tradicionalmente vinculados a métodos convencionais, como desenhos em CAD e planilhas manuais. Essa barreira cultural, somada à necessidade de investimentos em capacitação e infraestrutura tecnológica, pode retardar a adoção em larga escala.

Por isso, é fundamental demonstrar, na prática, como a integração entre BIM, IA e desenvolvimento web pode superar esses desafios, trazendo não apenas eficiência operacional, mas também vantagens competitivas para arquitetos, engenheiros e construtoras menores.

De acordo com Zhang et al. (2022) e Rane (2023) a integração entre BIM e IA oferece uma oportunidade única para transformar as práticas tradicionais de construção em práticas automatizadas e orientadas a dados o que tem o potencial de abordar de maneira eficaz desafios persistentes na construção civil, tais como: atrasos em projetos, orçamentos imprecisos e mal dimensionados, variações de qualidade e riscos à segurança.

1.2. Definição do Problema

Como destacado por Paiva (2016), o uso da tecnologia foi e continua sendo um fator essencial para superar limitações de mercado, como a competição acirrada e a dificuldade de agregar valor ao serviço sem comprometer prazos e qualidade. Diante deste contexto, este trabalho visa responder a seguinte problemática: **Como uma aplicação web pode integrar BIM e Inteligência Artificial para auxiliar arquitetos na análise e desenvolvimento de projetos arquitetônicos?**

1.3. Justificativa

A transição do 2D para o BIM gerou modelos ricos em dados, mas a complexidade e volume dessas informações limitam a análise manual e a extração de insights cruciais para a tomada de decisões, especialmente em pequenos escritórios.

Neste contexto, a integração da Inteligência Artificial Generativa (ChatGPT) com as plataformas de dados BIM (via Autodesk Platform Services - APS) é indispensável. Essa sinergia automatiza a extração e adiciona uma camada de

interpretação inteligente. A IA processa dados complexos, identifica padrões, prevê conflitos/ineficiências (custos, energia) e traduz essas descobertas em sugestões proativas em linguagem natural.

1.4. Objetivos

1.4.1. Objetivo Geral

Este trabalho tem como objetivo geral desenvolver uma aplicação web que integre os conhecimentos em Engenharia de Software, Inteligência Artificial (IA) e Building Information Modeling (BIM) para permitir com que os projetos arquitetônicos sejam mais robustos, eficientes em relação a custos e consumo energético sustentável.

1.4.2. Objetivos Específicos

- Recorrer às bibliografias de boas práticas de desenvolvimento de software e metodologias adequadas e relevantes para desenvolvimento da aplicação.
- Integrar a API Autodesk Forge Viewer para visualização interativa dos modelos BIM diretamente no navegador
- Extrair e estruturar dados relevantes dos modelos BIM enviados
- Desenvolver uma funcionalidade baseada em IA capaz de analisar os dados dos modelos e fornecer sugestões ou alertas úteis para o arquiteto.
- Propor o uso da plataforma como ferramenta de apoio em escritórios de arquitetura, visando maior eficiência, qualidade e acessibilidade no desenvolvimento de projetos.

1.5. Metodologia de Pesquisa

Esta pesquisa, segundo sua natureza, é um resumo de assunto, buscando a compreensão acerca do tema e a partir da investigação e observação das informações contíguas para compreender suas causas e explicações (Wazlawick, 2014)

Segundo os objetivos é uma pesquisa exploratória. A pesquisa exploratória é aquela em que o autor não tem necessariamente uma hipótese ou objetivo definido

em mente. Ela pode ser considerada, muitas vezes, como o primeiro estágio de um processo de pesquisa mais longo. Na pesquisa exploratória, o autor vai examinar um conjunto de fenômenos, buscando anomalias que não sejam ainda conhecidas e que possam ser, então, a base para uma pesquisa mais elaborada (Wazlawick, 2014).

Quanto aos procedimentos técnicos, esta pesquisa é bibliográfica e experimental. A pesquisa bibliográfica requer o estudo de teses, artigos, entre outros. A pesquisa experimental é caracterizada por ter uma ou mais variáveis experimentais que podem ser coordenadas pelo pesquisador (Wazlawick, 2014).

A pesquisa bibliográfica, será elaborada a partir de materiais já publicados, podendo incluir livros, teses, materiais disponibilizados na Internet, revistas, entre outros. A principal vantagem é permitir uma sucessão de fenômenos maior do que seria capaz de pesquisar diretamente (Gil, 2021).

Para Manzo (1971), a bibliografia oferece meios para definir, resolver, não somente problemas já conhecidos, como também explorar novas áreas onde os problemas não se cristalizaram suficientemente e tem por objetivo permitir ao cientista o reforço paralelo na análise de suas pesquisas ou manipulação de suas informações. Referente aos procedimentos técnicos, como a revisão bibliográfica envolve a análise de materiais já publicados, como livros, teses, recursos online e revistas, entre outros.

Sua principal vantagem é permitir uma abordagem mais ampla de diversos fenômenos, além do que seria possível se pesquisar diretamente (Gil, 2021).

1.6. Organização do Trabalho

Esse trabalho foi desenvolvido seguindo os princípios de Gil (2021), onde de acordo com ele as etapas para uma revisão bibliográfica são:

a) Escolha do tema: Integração de BIM e Inteligência Artificial: Desenvolvimento de uma plataforma web para análise e suporte a projetos arquitetônicos

b) Revisão preliminar da literatura: levantamento bibliográfico de acordo com área de interesse do pesquisador com intuito de familiarização com o tema e facilitação na definição do problema de pesquisa.

c) **Formulação da pergunta de pesquisa:** Como uma aplicação web pode integrar BIM e Inteligência Artificial para auxiliar arquitetos na análise e desenvolvimento de projetos?

d) **Identificação das fontes:** Identificar as fontes bibliográficas que forneçam informações relevantes para responder ao problema proposto, consultando dissertações, periódicos científicos, obras de referência, plataforma Capes e Google Academy.

e) **Análise do conteúdo:** Examinar as informações e dados coletados das fontes selecionadas, relacionando-os à pergunta de pesquisa e avaliando a consistência dos argumentos apresentados.

f) **Fichamento:** O fichamento foi realizado com ênfase em trabalhos relevantes para a pesquisa. No entanto, documentos com contribuições médias ou baixas também foram analisados para enriquecer as reflexões do autor. Essa abordagem permitiu que o TCC fosse escrito com uma base sólida.

g) **Escrita do Trabalho de Conclusão de Curso** A pesquisa experimental consiste em estabelecer um objeto de estudo, escolher as variáveis que a influenciam e determinam as formas de controle e 25 observar os efeitos que a variável gera no objeto. Realiza pelo menos um dos elementos que julga ser responsável pela circunstância que está sendo pesquisado (Gil, 2017). Pesquisa Experimental, tem como passos:

a) Formulação do problema: Como uma aplicação web pode integrar BIM e Inteligência Artificial para auxiliar arquitetos na análise e desenvolvimento de projetos arquitetônicos mais robustos e sustentáveis?

b) Construção das hipóteses:

- O uso de Inteligência Artificial permite identificar padrões e sugerir melhorias automatizadas nos projetos arquitetônicos.
- Uma aplicação web com suporte a BIM e IA pode aumentar a eficiência dos arquitetos, promovendo análises mais rápidas e precisas.
- A combinação de BIM e IA em um ambiente web pode reduzir erros humanos e otimizar o processo de tomada de decisões

c) Operacionalização das variáveis:

- **Variável Independente:** Integração de BIM e Inteligência Artificial em uma aplicação web

- Variável Dependente: Eficiência na análise e desenvolvimento de projetos arquitetônicos
- Variável de Controle:
 - Complexidade do projeto
 - Nível de experiência dos arquitetos
 - Recursos computacionais disponíveis
 - Qualidade do modelo BIM utilizado

d) Determinação do ambiente: Para desenvolver foi utilizado um Laptop: CPU: 12th Gen Intel(R) Core(TM) i7-12650H 2.30 GHz, RAM: 32gb a 3200mhz, 25 MotherBoard: unknown e Placa Gráfica: Geforce RTX 4050 Laptop, Sistema Operacional: WIN 11 Home version 24H2, trabalhando com o ambiente de desenvolvimento integrado (IDE) Visual Studio Code, além do Github Desktop para controle de versão do projeto.

e) Coleta de dados: A coleta de dados foi realizada por meio de documentações disponíveis na internet, informações de arquitetos e suas experiências com plataformas web, estudo de dados feitos por arquitetos, engenheiros e urbanistas.

f) Análise e interpretação dos dados: A análise dos dados consistiu na comparação entre os resultados obtidos com o uso da aplicação e os métodos tradicionais de análise de projetos arquitetônicos. Indicadores como tempo de resposta, precisão das sugestões da IA e melhoria na tomada de decisões foram utilizados para interpretar o impacto da integração entre BIM e IA.

g) Redação do relatório: A redação do TCC será feita com base nos dados coletados e analisados, estruturando o relatório conforme as normas acadêmicas, apresentando referencial teórico, a metodologia aplicada, os resultados alcançados e as conclusões sobre a eficácia da aplicação web proposta.

2. DESENVOLVIMENTO

Este capítulo tem por objetivo apresentar os conceitos fundamentais do sistema, suas tecnologias empregadas durante o processo de desenvolvimento, ferramentas e arquitetura. Para a construção da solução proposta, foram adotados conceitos e ferramentas amplamente utilizados no desenvolvimento web contemporâneo.

2.1 Conceitos Fundamentais

2.1.1 Desenvolvimento Web

O desenvolvimento WEB concentra-se na criação de sites, softwares, bancos de dados e diversos outros serviços disponíveis na Internet. A prática dessa atividade exige o uso de ferramentas e frameworks que adotam metodologias ágeis, visando otimizar o processo de programação e entrega de soluções.

Nesse cenário, destacam-se as ferramentas responsáveis pela construção da interface gráfica da aplicação, denominadas front-end, além de um conjunto estruturado de aplicações voltadas à manipulação e ao controle dos dados, conhecido formalmente como back-end (VOOMP, 2021).

Na arquitetura cliente-servidor, esses componentes se distribuem em dois ambientes:

- client-side: onde está hospedado o front-end;
- server-side: que armazena e executa o back-end.

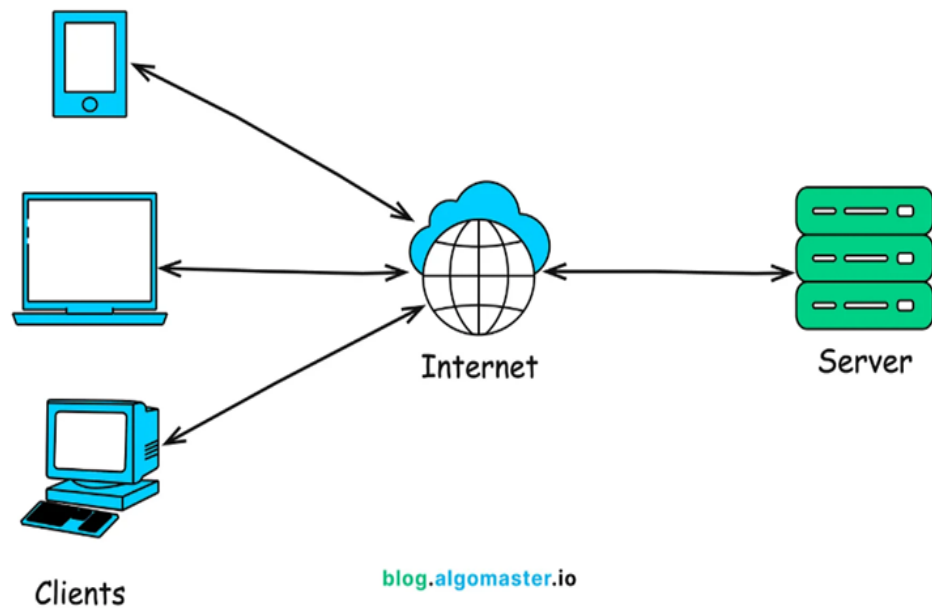
No contexto do desenvolvimento formal para WEB, cabe ao desenvolvedor front-end garantir a qualidade da experiência do usuário dentro da aplicação. Esse profissional é responsável por projetar e construir as páginas com as quais o usuário irá interagir, incorporando elementos visuais que definem a identidade do site ou sistema (DIGITAL HOUSE, 2019).

2.1.2 Arquitetura Cliente Servidor

A aplicação adota a arquitetura cliente-servidor, na qual o cliente corresponde à aplicação web e o servidor é representado pelo backend. O cliente é responsável pela interface com o usuário, capturando suas interações e enviando requisições ao servidor. O backend, por sua vez, processa essas requisições, executa a lógica de negócios e retorna as respostas adequadas.

A comunicação entre cliente e servidor ocorre por meio dos protocolos HTTP/HTTPS, utilizando APIs RESTful para a troca estruturada de informações. O backend também é encarregado do gerenciamento do banco de dados, assegurando a integridade, consistência e segurança dos dados manipulados pela aplicação. A Figura 1 apresenta a visão geral da arquitetura cliente-servidor adotada pelos sistemas web.

Figura 1 - Arquitetura Cliente-Servidor

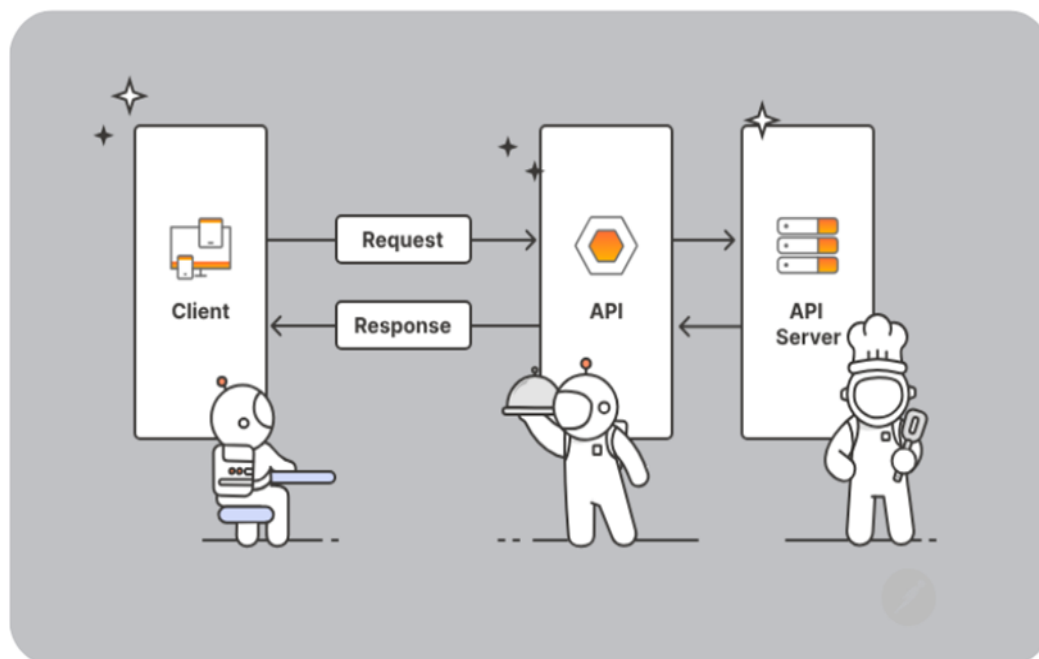


Fonte: blog.algomaster.io

2.1.3 Application Programming Interface (API)

Segundo o site oficial da Red Hat (2023), APIs são ferramentas utilizadas para integrar sistemas sem que seja necessário conhecer, de fato, como o serviço foi implementado. Elas dispõem de um protocolo e de uma documentação bem estruturada, o que garante que o interessado possa formular sua solicitação de maneira precisa e receber, em retorno, uma resposta adequada da API. Na Figura 2, é apresentado, de forma ilustrativa, o funcionamento de uma API: o usuário envia uma requisição, à API realiza a comunicação com o banco de dados e retorna o conteúdo solicitado, atuando como um verdadeiro “garçom”.

Figura 2 - Fluxo de uma API.



Fonte: Postman

2.1.4 Representational State Transfer (REST API)

As APIs utilizam a comunicação via **Hypertext Transfer Protocol (HTTP)**, um protocolo utilizado para a transmissão de documentos de hipermídia. De acordo com **Patni (2017)**, APIs baseadas nas regras do REST apresentam características que contribuem para a qualidade, escalabilidade e eficiência de sua arquitetura. Algumas delas são:

- **Escalabilidade:** a comunicação simplificada entre componentes é uma característica essencial no REST, facilitando o crescimento de sistemas distribuídos e permitindo que novos componentes sejam adicionados sem comprometer a eficiência da aplicação.
- **Modificabilidade:** a estrutura REST possibilita a modificação de componentes de forma independente, promovendo a separação de responsabilidades, versionamento e reduzindo riscos de falhas quando alterações são realizadas em

partes específicas do sistema.

- **Confiabilidade:** a restrição de não manter estado (*stateless*) favorece a recuperação em caso de falhas, uma vez que cada solicitação inclui todas as informações necessárias para seu processamento.

No quadro 1 temos os principais verbos HTTP e sua respectiva descrição

Quadro 1 - Principais verbos HTTP

Verbos HTTP	Descrição
GET	O método GET solicita a representação de um recurso específico. Requisições utilizando o método GET devem retornar apenas dados.
HEAD	O método HEAD solicita uma resposta de forma idêntica ao método GET, porém sem conter o corpo da resposta.
POST	O método POST é utilizado para submeter uma entidade a um recurso específico, frequentemente causando uma mudança no estado do recurso ou efeitos colaterais no servidor.
PUT	O método PUT substitui todas as atuais representações do recurso de destino pela carga de dados da requisição.
DELETE	O método DELETE remove um recurso específico.
OPTIONS	O método OPTIONS é usado para descrever as opções de comunicação com o recurso de destino

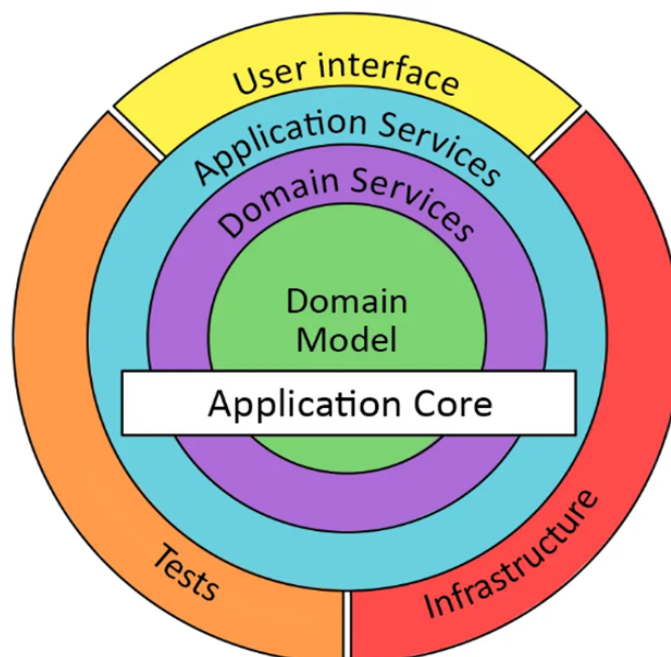
Fonte: Mozilla Corporation

2.1.5 Domain-Driven Design (DDD)

A arquitetura do sistema foi planejada para garantir alta modularidade, facilidade de manutenção e capacidade de evolução da plataforma ao longo do tempo. Para isso, foram adotadas abordagens arquiteturais distintas, porém complementares, no frontend e no backend.

No frontend, utilizou-se o Domain-Driven Design (DDD) como estratégia de organização estrutural, segmentando a aplicação em domínios específicos, como autenticação, gerenciamento de projetos, empreendimentos, visualização 3D e comunicação com a IA, cada um contendo suas próprias entidades, casos de uso, modelos e serviços. Essa divisão favorece a clareza na separação de responsabilidades, facilita a expansão de funcionalidades e reduz o impacto de mudanças localizadas, garantindo um fluxo de código mais consistente e previsível. Onde podemos ver na figura 3 uma ilustração para entender como a parte de visualização, os serviços, a separação dos domínios é separados em camadas que revestem o domínio.

Figura 3 – Ilustração da modelo de DDD – Domain Driven Design



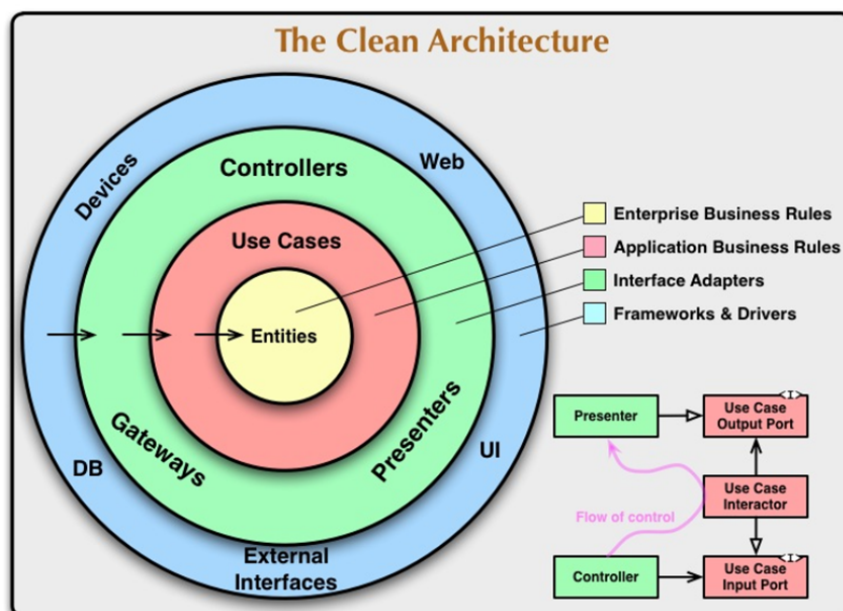
Fonte: Medium

2.1.6 Clean Architecture

No backend, foi adotada a Clean Architecture, permitindo que regras de negócio e lógica de aplicação sejam totalmente independentes de frameworks, banco de dados ou serviços externos. Esse modelo possibilita a substituição de tecnologias como Amazon S3, Amazon SES, provedores de armazenamento, bancos de dados ou bibliotecas de terceiros sem que seja necessário modificar a lógica central do sistema. A arquitetura é estruturada por camadas bem definidas, com entidades, regras de negócios, interfaces e adaptadores, proporcionando alto grau de desacoplamento, estabilidade e robustez.

Segundo Martin (2018), a Clean Architecture é um conjunto de princípios destinados a criar sistemas independentes de frameworks, testáveis, flexíveis e facilmente evolutivos. A ideia central é que a arquitetura deve permitir que regras de negócio existam de forma isolada de detalhes externos, de modo que mudanças em tecnologias ou integrações não afetem o núcleo da aplicação. Onde a seguir podemos ver uma ilustração da arquitetura limpa na figura 4.

Figura 4 - Arquitetura Limpa



Fonte: [blog.cleancoder](http://blog.cleancoder.com)

2.2. Tecnologias e Ferramentas

Para a construção da solução proposta, foram adotadas abordagens e ferramentas modernas e consolidadas no mercado de desenvolvimento de software. A arquitetura da aplicação é baseada em uma clara separação entre a interface do usuário (frontend) e a lógica de negócios e persistência de dados (backend), utilizando tecnologias específicas para garantir a performance, escalabilidade e manutenção em ambas as frentes. A seguir, serão detalhadas as principais tecnologias e frameworks utilizados em cada uma dessas camadas, bem como o sistema de gerenciamento de banco de dados escolhido para a aplicação.

2.2.1 Frontend

Para o desenvolvimento da camada de interação com o usuário, foi realizada uma seleção das tecnologias emergentes, modernas, robustas e escaláveis a fim de proporcionar ao usuário uma navegação fluida e simplificada, além de facilitar o processo de desenvolvimento e manutenção do código. A seguir, são apresentadas as principais tecnologias utilizadas no frontend do sistema:

Typescript

O TypeScript foi escolhido como linguagem principal para o desenvolvimento da plataforma web por fornecer um modelo de tipagem estática robusto ao JavaScript, trazendo benefícios diretos para a confiabilidade e manutenção do sistema. Entre suas principais contribuições, destacam-se:

- **Prevenção de erros em tempo de desenvolvimento:** A verificação de tipos permite identificar inconsistências antes da execução, evitando falhas comuns relacionadas ao uso incorreto de dados.
- **Código mais claro e bem documentado:** A definição explícita de tipos funciona como uma documentação automática, facilitando o entendimento das estruturas, interfaces e fluxos de dados entre os módulos da aplicação.
- **Refatoração mais segura:** Alterações estruturais no código podem ser

realizadas com maior segurança, pois o compilador aponta automaticamente trechos que precisam ser ajustados após uma modificação.

- **Suporte avançado das IDEs:** O TypeScript melhora significativamente a experiência de desenvolvimento ao fornecer autocompletar inteligente, sugestões contextuais e identificação de erros em tempo real.
- **Integração eficiente com frameworks modernos:** A compatibilidade do TypeScript com tecnologias como React e Next.js permite tipar componentes, propriedades, estados, hooks e contextos, resultando em um código mais previsível, escalável e de fácil manutenção.

Next.js

O Next.js foi adotado como framework principal da camada frontend por oferecer um ecossistema maduro e recursos avançados de renderização, alinhados às necessidades de desempenho e escalabilidade da aplicação. Baseado em React, o framework disponibiliza diferentes estratégias de renderização que tornam a entrega de conteúdo mais eficiente:

- **Server-Side Rendering (SSR):** As páginas são montadas no servidor a cada requisição, garantindo conteúdos atualizados e melhorando o tempo de resposta inicial para o usuário.
- **Static Site Generation (SSG):** Durante o processo de build, o Next.js gera páginas estáticas que são servidas rapidamente, reduzindo o esforço do servidor e aumentando a velocidade de carregamento.

De acordo com a documentação oficial (VERCEL, 2024), o Next.js também apresenta um sistema de rotas baseado na estrutura de diretórios, simplificando a organização do projeto e eliminando configurações manuais extensas. Entre outros diferenciais, o framework inclui: otimização automática de imagens, divisão inteligente de código, pré-carregamento de rotas (prefetching) e suporte nativo para API Routes, que permitem implementar endpoints dentro do próprio ambiente do frontend.

Shadcn UI

Para o desenvolvimento dos componentes visuais, foi utilizada a biblioteca de componentes Shadcn UI, que se destaca por fornecer componentes reutilizáveis e altamente customizáveis, garantindo maior liberdade e agilidade no design da interface. Diferentemente de outras bibliotecas tradicionais, o Shadcn UI funciona como um conjunto de componentes incorporados diretamente ao código-fonte (SHADCN UI, 2025), o

que possibilita:

- **Personalização completa:** Cada componente pode ser adaptado conforme a necessidade do projeto, sem limitações impostas por estilos rígidos.
- **Uniformidade visual:** Apesar da flexibilidade, a biblioteca segue um padrão estético consistente, favorecendo interfaces limpas e coerentes.
- **Acessibilidade integrada:** Os componentes são desenvolvidos com boas práticas de acessibilidade, aprimorando a usabilidade para diferentes perfis de usuários.
- **Coleção abrangente de elementos:** Inclui botões, formulários, caixas de diálogo, tabelas, menus e outros componentes essenciais, todos orientados à experiência do usuário e responsividade.

Zod

Para validação de dados na camada frontend, foi utilizada a biblioteca Zod, que oferece uma abordagem declarativa para construção de esquemas de validação com suporte pleno ao TypeScript (ZOD, 2025). A adoção do Zod permitiu:

- **Definição declarativa e reutilizável de esquemas:** As regras são estruturadas de forma clara, facilitando alterações futuras e mantendo o código organizado.
- **Inferência automática de tipos:** A partir dos esquemas definidos, o Zod gera automaticamente os tipos correspondentes em TypeScript, evitando duplicação de definições.
- **Validação em tempo de execução:** Além do suporte estático do TypeScript, o Zod valida dados recebidos da API ou fornecidos pelo usuário, garantindo integridade em múltiplas camadas.
- **Customização de mensagens de erro:** A biblioteca permite mensagens detalhadas para feedback preciso ao usuário.

A integração entre Zod e TypeScript resultou em validações consistentes e confiáveis, reduzindo significativamente erros relacionados a dados inválidos.

React Hook Form

A gestão de formulários foi implementada com o React Hook Form, uma biblioteca projetada para oferecer alto desempenho e baixo custo de renderização. Entre suas principais vantagens:

- **Renderizações minimizadas:** Os campos são controlados por referências (refs), evitando atualizações desnecessárias na interface.
- **Alto desempenho interno:** A manipulação de estados é otimizada, o que garante fluidez mesmo em formulários extensos.
- **Compatibilidade com bibliotecas de validação:** A integração com Zod — visa aplicar os mesmos esquemas de validação em diferentes partes da aplicação.
- **Tratamento simplificado de estruturas complexas:** Arrays de campos, dependências condicionais e formulários aninhados são suportados de maneira intuitiva.
- **Flexibilidade de componentes controlados ou não controlados:** Permitindo ajustar o comportamento conforme a necessidade de cada caso.

Outras Tecnologias e Bibliotecas

A combinação entre React Hook Form e Zod resultou em formulários sólidos, tipados e fáceis de manter. Além dos recursos centrais já citados, o frontend utiliza bibliotecas complementares que reforçam a qualidade do desenvolvimento:

- **Tailwind CSS:** Framework de utilitários CSS que permite compor interfaces responsivas por meio de classes funcionais, acelerando o desenvolvimento sem abrir mão da personalização.
- **Axios:** Cliente HTTP com suporte a Promises, utilizado para comunicação com o backend, incluindo interceptors e tratamento padronizado de respostas e erros.
- **Date-fns:** Biblioteca para manipulação de datas, oferecendo funções modernas e performáticas para formatação e cálculos temporais.

- React Context API: Usada para compartilhamento de estado global, como informações do usuário autenticado ou preferências da aplicação.
- React Query: Solução para gerenciamento de estado do servidor, fornecendo cache inteligente, sincronização automática e estratégias de refetch otimizadas.

O conjunto dessas tecnologias resultou em uma base frontend robusta, escalável e alinhada às necessidades da aplicação.

2.2.2 Backend

Nest.js

O backend foi estruturado utilizando o Nest.js, um framework para Node.js que adota uma arquitetura modular inspirada em padrões amplamente utilizados no desenvolvimento corporativo. Suas principais características incluem:

- Organização modular: As funcionalidades são divididas em módulos independentes, favorecendo a escalabilidade, manutenção e desenvolvimento simultâneo.
- Sistema de injeção de dependências: Facilita o desacoplamento entre componentes e possibilita a criação de testes mais eficazes por meio de mocks e substituições.
- Uso de decoradores: Através de recursos avançados do TypeScript, o Nest.js permite definir rotas, serviços e middlewares de forma clara e declarativa.
- Compatibilidade com múltiplos protocolos: Além do HTTP, o framework suporta WebSockets, gRPC e soluções baseadas em mensageria, ampliando as possibilidades arquiteturais.
- Componentes especializados: Interceptores, Guards e Pipes facilitam tarefas como validação, transformação e controle de acesso.

Essa abordagem opinativa contribuiu para um código padronizado, limpo e consistente em toda a aplicação.

Prisma ORM

O acesso ao banco de dados foi implementado por meio do Prisma ORM, que oferece uma experiência moderna, segura e fortemente tipada. Entre seus benefícios:

- Schema unificado: A definição da estrutura de dados é feita por meio do Prisma Schema, que facilita a leitura e o entendimento do modelo relacional.
- Migrações automáticas: A partir do schema, o Prisma gera e controla migrações SQL versionadas, assegurando uma evolução estável do banco.
- Cliente Type-safe: O Prisma Client fornece autocompletar e verificação estática de queries, reduzindo erros durante o desenvolvimento.

Com isso, o Prisma simplificou significativamente o trabalho de persistência de dados e aumentou a confiabilidade do backend.

AWS SES

Para envio de e-mails transacionais, especialmente no processo de autenticação via links temporários, foi utilizado o serviço de e-mail da AWS, conhecido por sua confiabilidade e alto desempenho. Seus principais benefícios incluem:

- API REST simples e eficiente: Permite integrar rapidamente o envio de mensagens ao backend, com documentação completa.
- Templates personalizáveis: Possibilidade de utilizar modelos HTML responsivos para padronizar comunicações com o usuário.
- Alta escalabilidade: O serviço é capaz de lidar com grande volume de e-mails sem necessidade de gerenciar infraestrutura própria.

Essa integração assegura confiabilidade no processo de autenticação e demais notificações enviadas pelo sistema.

AWS S3

Para o armazenamento e gestão de arquivos digitais, como os modelos BIM e seus dados associados, foi escolhido o AWS Simple Storage Service (S3). Este serviço de *storage* em nuvem se destaca por sua durabilidade, disponibilidade e

escalabilidade, essenciais para lidar com o volume e a natureza dos arquivos de projetos arquitetônicos. O S3 permitiu o armazenamento seguro e a recuperação eficiente dos dados, além de facilitar a integração com a Autodesk Platform Services (APS), servindo como repositório primário para os arquivos a serem processados. A flexibilidade do S3 em termos de políticas de acesso e segurança reforça a confiabilidade da plataforma para a gestão de informações sensíveis.

Outras Tecnologias e Bibliotecas

O backend também faz uso de ferramentas complementares que reforçam a segurança e documentação do sistema:

- JSON Web Token (JWT): Utilizado no controle de autenticação e autorização, garantindo acesso seguro aos recursos da API.
- bcrypt: Implementado para gerar hashes seguros de senhas, impedindo o armazenamento em texto plano.
- Swagger/OpenAPI: Responsável pela documentação automatizada dos endpoints da API, facilitando o entendimento e testes dos recursos.

A composição dessas tecnologias resultou em um backend consistente, seguro e preparado para lidar com requisitos de escalabilidade e performance.

2.2.3 Banco de Dados

PostgreSQL

Para armazenamento de dados, foi utilizado o PostgreSQL, um sistema de gerenciamento relacional maduro e amplamente consolidado. Sua adoção se justifica pelos seguintes fatores:

- Variedade avançada de tipos de dados: Além dos tipos primários, suporta JSON/JSONB, arrays, UUID e tipos personalizados, adequando-se a cenários complexos.
- Capacidade de extensão: Extensões como PostGIS e pgcrypto ampliam o escopo do banco para cenários específicos, como georreferenciamento e criptografia.

- Recursos de replicação e disponibilidade: Funcionalidades nativas de replicação permitem maior segurança contra falhas e suporte a grandes cargas.

A combinação de robustez, desempenho e flexibilidade consolidou o PostgreSQL como a escolha ideal para o sistema.

2.2.4 Inteligência Artificial

Para a implementação do fluxo de Inteligência Artificial da plataforma, optou-se pelo uso do n8n combinado a modelos de linguagem avançados, como o ChatGPT. A escolha pelo n8n foi estratégica, considerando sua natureza de código aberto, sua modularidade e a ampla capacidade de integração com APIs externas. Com mais de 200 integrações disponíveis, o n8n permite a criação de fluxos complexos de forma visual, mantendo flexibilidade e reduzindo a necessidade de codificação manual. Essa característica tornou a ferramenta particularmente adequada ao cenário do projeto, que exige constantes adaptações e testes com diferentes modelos de IA.

O fluxo de Inteligência Artificial foi estruturado em duas partes distintas, com o objetivo de organizar responsabilidades e otimizar o desempenho. O primeiro fluxo foi destinado exclusivamente à análise dos dados, processando informações enviadas pela plataforma, estruturando entradas, validando formatos e preparando o conteúdo que seria encaminhado ao modelo de IA. Já o segundo fluxo foi projetado para manipular a troca de mensagens entre o modelo de IA e o chat integrado à interface do usuário. Essa divisão permitiu maior controle sobre cada etapa do processo, facilitando manutenção, escalabilidade e rastreamento de erros.

Em relação à segurança, foram adotadas boas práticas que envolvem o uso de credenciais protegidas, isolamento de fluxos e controle de permissões dentro do n8n. Todas as chaves de API utilizadas para comunicação com provedores de IA foram armazenadas de forma segura, impedindo acesso direto no código da aplicação. Além disso, foram aplicadas camadas adicionais de verificação de dados antes da transmissão, garantindo que apenas informações necessárias e devidamente filtradas fossem enviadas aos modelos de linguagem.

A modularidade do n8n também proporciona alta flexibilidade na escolha dos provedores de Inteligência Artificial. A plataforma permite alternar ou combinar

diferentes modelos, como GPT, Gemini ou outros serviços compatíveis, sem a necessidade de grandes alterações no código principal. Essa característica assegura que o sistema permaneça adaptável a inovações do mercado de IA tornando possível testar diferentes modelos de forma rápida e configurar o que melhor se adequa às demandas do projeto.

Por fim, o uso do n8n contribui diretamente para a produtividade e para a simplificação da automação de processos. Sua interface visual facilita o desenvolvimento de fluxos mesmo por usuários com pouca experiência em programação, garantindo que o projeto possa evoluir de maneira ágil e sustentável, ao mesmo tempo em que mantém robustez e escalabilidade no fluxo de Inteligência Artificial.

2.3. Trabalhos Correlatos

A integração entre a Metodologia BIM (Building Information Modeling) e a Inteligência Artificial (IA) tem impulsionado o desenvolvimento de diversas ferramentas voltadas para a otimização do fluxo de trabalho na construção civil. No entanto, ao analisar as soluções disponíveis no mercado, nota-se uma predominância de aplicações voltadas para a renderização visual e assistência ao design estético, operando majoritariamente como extensões (plug-ins) de softwares de autoria desktop.

Um exemplo proeminente neste cenário é o usBIM.codesign (ACCA Software,) AI. Trata-se de um plug-in desenvolvido para o ambiente Autodesk Revit, cujo objetivo central é integrar a inteligência artificial generativa diretamente ao fluxo de modelagem. A ferramenta atua como um "co-designer", focando na transformação de modelos BIM em representações visuais de alta fidelidade. Suas funcionalidades permitem a criação de renderizações fotorrealistas e a experimentação de materiais e acabamentos em tempo real, sem a necessidade de exportar o arquivo para softwares externos de renderização.

O usBIM.codesign AI destaca-se por recursos como o Virtual Staging AI, que enriquece ambientes com mobiliário e decoração automaticamente, e o gerador de arquitetura, que auxilia na concepção de fachadas e estudos volumétricos a partir de esboços ou modelos brutos. Embora ofereça ganhos significativos em produtividade visual e na tomada de decisões estéticas — atendendo arquitetos e designers de

interiores , sua operação é dependente da instalação local do software Revit e seu foco reside na geração de imagens (saída visual), e não na análise semântica dos dados construtivos.

Paralelamente, existem soluções voltadas para a acessibilidade via web, como a plataforma de visualização usBIM. Esta ferramenta permite a renderização e navegação de modelos arquitetônicos diretamente no navegador, democratizando o acesso às informações do projeto sem exigir hardware robusto. Contudo, suas funcionalidades principais restringem-se à visualização geométrica, navegação e ferramentas básicas de medição. Observa-se, neste caso, certas limitações quanto à fidelidade de texturas e, principalmente, a ausência de uma camada de inteligência artificial capaz de interpretar os dados do modelo para além da representação gráfica.

Diante deste cenário, identifica-se uma lacuna tecnológica relevante. Enquanto as ferramentas citadas focam na visualização estética (renderização via plug-in) ou na visualização geométrica passiva (visualizador web tradicional), a proposta deste trabalho busca uma abordagem distinta. O sistema aqui desenvolvido não se limita a renderizar o modelo no navegador, mas propõe a utilização de Inteligência Artificial para a extração e análise de dados. O objetivo é transcender a representação visual, permitindo que a IA gere insights analíticos sobre o projeto, estabelecendo uma troca ativa de informações entre o usuário e o banco de dados do modelo 3D, funcionalidade esta não atendida plenamente pelas soluções de visualização ou renderização atualmente disponíveis.

3. ENGENHARIA DE SOFTWARE E METODOLOGIA

3.1 Levantamento de Requisitos

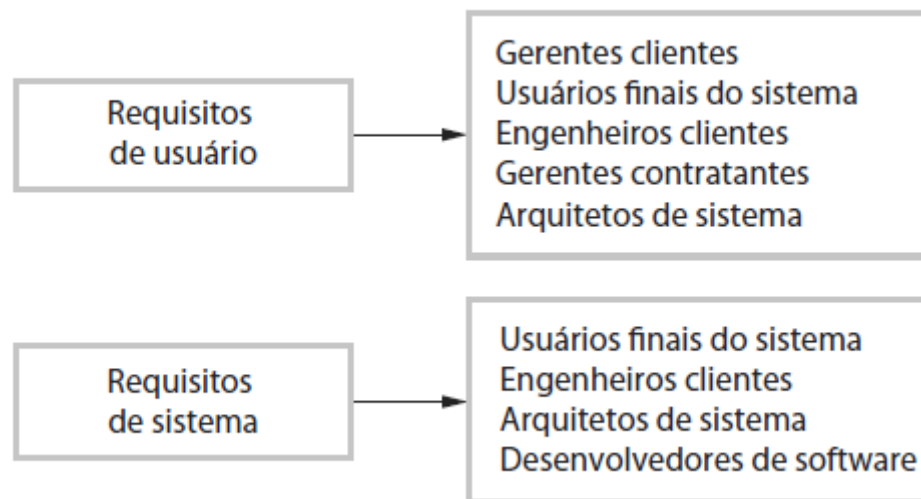
O Levantamento de Requisitos é a etapa inicial e crucial do desenvolvimento de software, na qual são identificadas, documentadas e validadas as necessidades dos usuários e as restrições do sistema. Este processo visa estabelecer uma compreensão clara e mútua entre o desenvolvedor e o cliente sobre o que o sistema deve fazer, servindo como base para todas as fases subsequentes, desde o design até os testes. A correta especificação dos requisitos, sejam eles funcionais (o que o sistema faz) ou não funcionais (como o sistema deve se comportar), é essencial

para o sucesso do projeto, garantindo que a solução final atenda precisamente aos objetivos propostos e às expectativas dos stakeholders.

O levantamento de requisitos foi realizado em colaboração com o arquiteto e professor Dr. Leônidas. Diversas reuniões foram conduzidas com ele e sua equipe para aprofundar o conhecimento na área, sendo este um trabalho que está em estudo e planejamento desde 2023.

Segundo Sommerville (2011), essas perguntas são separadas em requisitos funcionais e não funcionais , basicamente, requisitos do sistema. O autor afirma que estes requisitos retratam as necessidades dos clientes para serem atendidas pelo sistema.

Figura 5 - Tipos de definição de Requisitos de um sistema



Fonte: Sommerville

Segundo Sommerville (2011), os requisitos de usuário são direcionados principalmente aos gerentes de projetos, sendo expressos em linguagem natural e detalhando as necessidades que o sistema deve satisfazer. Tais requisitos são frequentemente empregados em processos de licitação. Em contraste, os requisitos de sistema são elaborados em linguagem técnica e definem de forma precisa o comportamento do sistema, sendo destinados a arquitetos e desenvolvedores.

A Figura 5 ilustra a distinção entre esses dois tipos de requisitos e seus respectivos públicos-alvo. A Engenharia de Requisitos classifica os Requisitos de Software em duas grandes categorias: funcionais e não-funcionais, as quais serão detalhadas no próximo tópico.

3.1.1. Requisitos Funcionais (RF)

Os requisitos funcionais descrevem o comportamento esperado de um sistema, especificando as funcionalidades que ele deve implementar para atender às necessidades dos usuários e cumprir seus objetivos (Sommerville, 2011). Esses requisitos exercem papel fundamental no direcionamento do desenvolvimento e asseguram que a solução proporcionará valor e corresponderá às expectativas dos stakeholders (Larman, 2004). No contexto desta aplicação, os requisitos funcionais foram definidos com o propósito de estabelecer as principais operações e interações previstas entre os usuários e o sistema web destinado à gestão de empreendimentos e projetos BIM, incluindo o processamento de modelos Revit, visualização 3D e análise inteligente utilizando Chat GPT.

Segundo Pressman (2015), os requisitos funcionais devem ser apresentados de forma clara e detalhada, descrevendo as funcionalidades do sistema a partir de suas entradas, processos e saídas. Dessa forma, para a aplicação proposta, que integra Autodesk Platform Services e recursos de IA, os principais requisitos funcionais incluem:

- **Autenticação e Gerenciamento de Acesso:** Como primeira etapa, a aplicação deve garantir que somente usuários autorizados consigam ter acesso a plataforma e suas funcionalidades. O sistema permite o cadastro e autenticação de usuários, assegurando que cada um tenha acesso apenas às operações relacionadas aos seus próprios empreendimentos e projetos BIM. Esse processo segue os princípios de segurança em sistemas de informação, conforme sugerido por Sommerville (2011).
- **Criação e Gestão de Empreendimentos:** O usuário deve poder registrar novos empreendimentos, inserindo informações como nome, descrição, localização e demais atributos relevantes. Cada empreendimento funciona como um agrupador de projetos BIM, permitindo o gerenciamento estruturado dos dados.
- **Criação e Gestão de Projetos BIM:** Dentro de cada empreendimento, o usuário pode criar projetos, definindo nome, descrição, área construída, preço,

custo e outros metadados. Esses projetos podem ser posteriormente vinculados aos modelos Revit enviados pelo usuário.

- Upload e Processamento de Arquivos Revit (RVT): O sistema deve permitir que o usuário selecione um arquivo Revit e o envie ao backend, que por sua vez o encaminha à Autodesk Platform Services (APS) para processamento. Após o término da conversão, a APS retorna uma URN, que é registrada no banco de dados para permitir a visualização 3D e extração de informações do modelo.

- Consulta, Visualização e Atualização de Modelos 3D: Os usuários devem ser capazes de visualizar seus modelos BIM por meio do APS Viewer da Autodesk integrado à aplicação. Também podem consultar o status do processamento, verificar propriedades extraídas e substituir modelos quando necessário. Segundo Sommerville (2011), oferecer essa flexibilidade ao usuário para gerenciar suas interações com o sistema é uma característica essencial abordada em sistemas bem projetados.

- Geração de Análises Inteligentes com IA: A aplicação deve possibilitar que o usuário solicite uma análise automatizada dos dados extraídos do modelo BIM. Para isso, o backend utiliza a URN para obter informações do APS e encaminha o conteúdo a um webhook do n8n, que integra o agente baseado em ChatGPT. A resposta analisada retorna ao sistema, permitindo interpretação técnica, insights e recomendações relacionadas ao projeto.

- Perguntas e Interações com o Assistente Inteligente: Além das análises automáticas, o usuário pode realizar perguntas diretamente ao assistente de IA. O sistema utiliza os dados extraídos do modelo para contextualizar a pergunta e gerar respostas especializadas sobre o projeto BIM.

Uma outra abordagem para requisitos funcionais é o de Mendes Silva(2008), que foca no usuário, porém trazendo termos entradas e saídas:

Um requisito de sistema de software especifica uma função que o sistema ou componente deve ser capaz de realizar. Estes são requisitos

de software que definem o comportamento do sistema, ou seja, o processo ou transformação que componentes de software ou hardware efetuam sobre as entradas para gerar as saídas. Estes requisitos capturam as funcionalidades sob o ponto de vista do usuário. (Mendes Silva Filho, 2008).

Portanto, os requisitos funcionais desta aplicação foram desenhados para otimizar o gerenciamento de empreendimentos e projetos BIM, oferecendo uma interface intuitiva, recursos avançados de visualização 3D e a integração com ferramentas de análise inteligente. Dessa forma, o sistema atende às necessidades específicas dos usuários ao proporcionar eficiência, clareza e suporte à tomada de decisão no contexto da modelagem de informações da construção.

O quadro abaixo representa requisitos funcionais do sistema.

Quadro 2 - Requisitos Funcionais (RF)

ID	Categoria	Requisito Funcional
1. Autenticação, Usuários e Acesso		
RF01	Usuários	Cadastro de Usuários: O sistema deve permitir o cadastro de novos usuários contendo nome, e-mail e senha.
RF02	Autenticação	Autenticação: O sistema deve permitir que o usuário realize login por meio de e-mail e senha, gerando um token JWT.
RF03	Acesso	Acesso Restrito: O sistema deve garantir que cada usuário tenha acesso somente aos dados relacionados à sua própria conta, incluindo empreendimentos, projetos e modelos.

2. Gestão de Empreendimentos (Enterprises)		
RF04	Empreendimentos	Criar Empreendimento: O sistema deve permitir que o usuário cadastre um novo empreendimento, informando nome, descrição e localização.
RF05	Empreendimentos	Listar Empreendimentos: O sistema deve exibir todos os empreendimentos cadastrados pelo usuário autenticado.
RF06	Empreendimentos	Atualizar Empreendimento: O usuário deve poder editar informações previamente cadastradas.
RF07	Empreendimentos	Excluir Empreendimento: O usuário deve poder excluir um empreendimento, removendo seus respectivos projetos associados.
3. Gestão de Projetos BIM (Projects)		
RF08	Projetos	Criar Projeto: O sistema deve permitir a criação de projetos vinculados a um empreendimento, contendo nome, descrição, área construída, custo, preço e demais metadados.
RF09	Projetos	Listar Projetos: O sistema deve permitir consultar todos os projetos associados a um empreendimento.
RF10	Projetos	Atualizar Projeto: O usuário deve poder modificar informações do projeto, incluindo metadados

		técnicos.
RF11	Projetos	Remover Projeto: O sistema deve permitir a exclusão de um projeto, mantendo a integridade relacional.
4. Upload, Conversão e Processamento de Modelos (Models / APS)		
RF12	Modelos	Upload de Arquivos RVT: O sistema deve permitir o envio de arquivos Revit (RVT) ao backend.
RF13	Modelos	Envio do Modelo para APS: Após o upload, o backend deve encaminhar o arquivo para a Autodesk Platform Services para conversão e processamento.
RF14	Modelos	Registrar URN: O sistema deve armazenar no banco de dados a URN retornada pela APS após o processamento.
RF15	Modelos	Acompanhar Status do Modelo: O sistema deve permitir a consulta do status de conversão (em processamento, finalizado ou erro).
RF16	Modelos	Substituir Modelo: O usuário deve poder atualizar um modelo existente, reenviando um novo arquivo RVT.
5. Visualização e Consulta de Modelos 3D		
RF17	Visualização	Visualizar Modelo em 3D: O sistema deve integrar o APS Viewer para permitir a visualização interativa do

		modelo BIM.
RF18	Visualização	Consultar Propriedades: O sistema deve permitir ao usuário visualizar propriedades, metadados e informações extraídas do modelo.
RF19	Visualização	Interação com Componentes: O usuário deve conseguir selecionar, isolar ou ocultar elementos diretamente no viewer.
6. Análises Inteligentes com IA (ChatGPT + n8n)		
RF20	Análise IA	Gerar Análise Automática: O sistema deve enviar dados extraídos do modelo a um webhook do n8n para análise por um agente de IA baseado em ChatGPT.
RF21	Análise IA	Exibir Resultado da Análise: A aplicação deve exibir a resposta da IA com insights, interpretações técnicas e recomendações.
7. Perguntas e Interação com o Assistente de IA		
RF22	Assistente IA	Enviar Pergunta ao Assistente: O sistema deve permitir que o usuário envie perguntas ao assistente inteligente.
RF23	Assistente IA	Contextualização com Dados do Modelo: O assistente deve utilizar as informações extraídas do modelo BIM para responder com base no contexto técnico.

RF24	Assistente IA	Histórico de Interações: O sistema deve permitir visualizar as conversas e análises anteriores.
8. Organização e Relacionamento de Dados		
RF25	Dados	Relacionar Empreendimento-Projeto-Modelo: O sistema deve manter o vínculo entre tabelas Users → Enterprises → Projects → Models.
RF26	Dados	Garantir Integridade: O sistema deve assegurar que um projeto só exista dentro de um empreendimento e que um modelo só exista dentro de um projeto.

Fonte: Autoria própria

3.1.2. Requisitos Não Funcionais (RNF)

Os Requisitos Não Funcionais (RNFs) definem a forma como o sistema deve operar, em contraste com os requisitos funcionais, que especificam o que ele faz.

Enquanto os requisitos funcionais estabelecem as funcionalidades e os comportamentos do sistema, os RNFs determinam atributos de qualidade cruciais, como desempenho, segurança, confiabilidade e restrições tecnológicas. Eles são essenciais para garantir que o sistema execute suas funções de maneira eficiente e consistente.

Segundo a engenharia de software, a importância dos RNFs é equivalente à dos requisitos funcionais, pois eles impactam diretamente a experiência do usuário, o nível de segurança, a escalabilidade e a capacidade de suportar o crescimento futuro do sistema.

Desse modo, os requisitos não funcionais, ao invés de dizerem o que o sistema faz, definem como ele fará. Consequentemente, eles se relacionam diretamente com a arquitetura do sistema, atuando como critérios a serem

atendidos, o que justifica sua denominação como atributos de qualidade (MENDES SILVA FILHO, 2008).

Quadro 3 - Requisitos Não Funcionais (RNF)

Identificador	Categoria	Requisito Não Funcional
RNF01	Desempenho	O sistema deve responder às operações principais em até 2 segundos.
RNF02	Desempenho	A arquitetura deve permitir subir múltiplas instâncias via Docker para suportar aumento de usuários (Escalabilidade).
RNF03	Segurança	O acesso deve ocorrer via JWT e comunicação criptografada (Autenticação Segura).
RNF04	Segurança	Senhas devem ser armazenadas com hashing seguro e consultas protegidas contra ataques comuns (SQL Injection, XSS) (Proteção de Dados).
RNF05	Disponibilidade	O sistema deve manter funcionamento

Identificador	Categoria	Requisito Não Funcional
		contínuo com disponibilidade mínima estimada em 99% (Alta Disponibilidade).
RNF06	Disponibilidade	O banco deve ter backups automáticos para recuperação em caso de falhas (Backup).
RNF07	Usabilidade	A interface deve ser intuitiva e de fácil uso para usuários não técnicos (Interface Simples).
RNF08	Usabilidade	O sistema deve funcionar corretamente em desktop e dispositivos móveis (Responsividade).
RNF09	Manutenibilidade	O código deve seguir padrões de organização e formatação, facilitando manutenção futura (Padrões de Código).
RNF10	Manutenibilidade	A API deve possuir documentação

Identificador	Categoria	Requisito Não Funcional
		clara (ex.: Swagger) (Documentação de API).

Fonte: Autoria própria

3.2. Atores do Sistema

Os atores envolvidos no sistema incluem:

Quadro 4 - Atores do sistema

Ator	Descrição
Usuário Autenticado	Pessoa que acessa o sistema, envia modelos BIM, solicita análises e visualiza resultados.
Sistema Autodesk Platform Services (APS)	Serviço externo responsável por receber arquivos, processar modelos BIM e devolver metadados.
API do ChatGPT (OpenAI)	Serviço de IA que recebe prompts e retorna análises inteligentes sobre o modelo.
Navegador/Cliente Web	Interface que envia requisições ao backend e apresenta os resultados ao usuário.

Fonte: Autoria própria

4. ARQUITETURA E MODELAGEM DO SISTEMA

4.1. Arquitetura da Solução

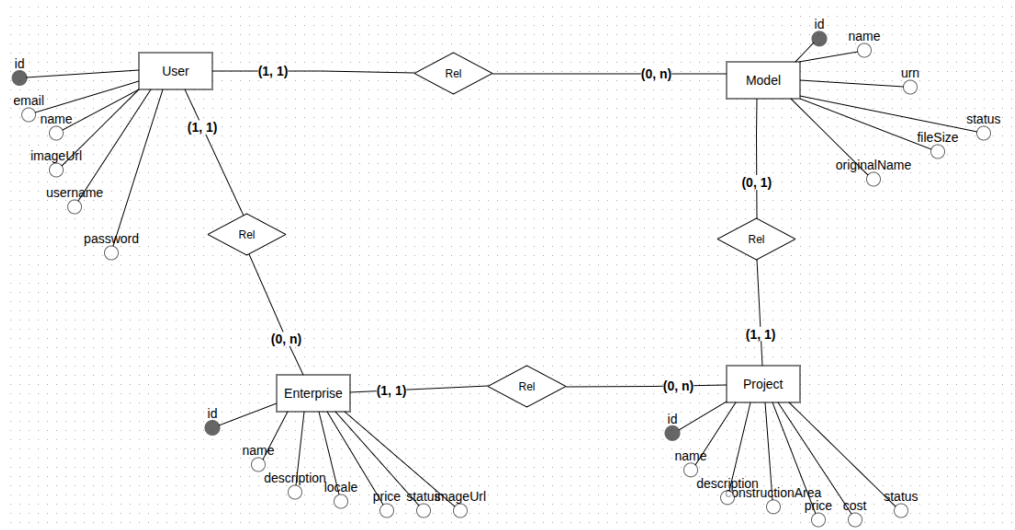
* (Diagrama da arquitetura macro: Frontend, Backend, Banco, Serviços Externos)

4.2. Modelagem de Dados

A Modelagem de Dados é uma etapa essencial da Engenharia de Software, cujo objetivo é definir a estrutura lógica do banco de dados, representando as informações que serão armazenadas e a forma como se relacionam. Para o desenvolvimento desta aplicação, que lida com a persistência de entidades complexas como Empreendimentos, Projetos BIM, usuários e dados de análise da IA, a clareza na modelagem é fundamental.

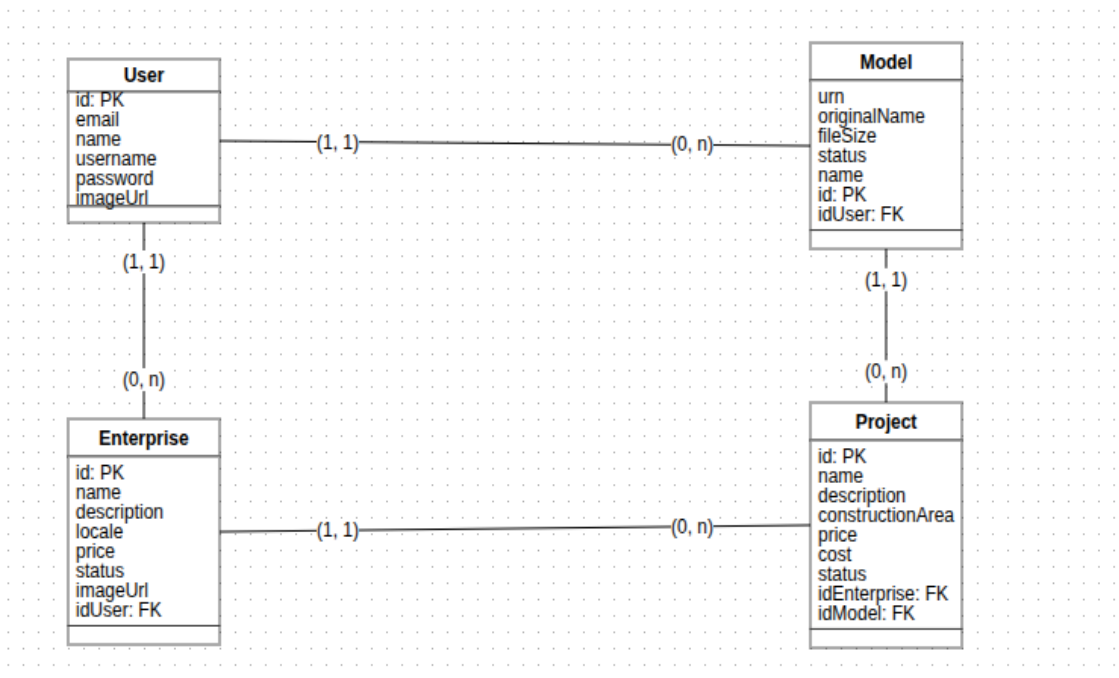
Neste trabalho, a estrutura de dados é formalizada através de duas representações complementares: o Modelo Entidade-Relacionamento (MER) e o Diagrama Entidade-Relacionamento (DER). O MER atua no nível conceitual, definindo as entidades principais do sistema (pessoas, lugares, coisas ou eventos) e os relacionamentos significativos entre elas. Já o DER é a representação gráfica do MER, utilizando notações específicas para visualizar as entidades, seus atributos e a cardinalidade dos relacionamentos. Juntos, esses modelos garantem que a lógica de negócios seja traduzida em um esquema de banco de dados consistente, escalável e otimizado para as necessidades da aplicação.

Figura 6 - Modelo conceitual



Fonte: Autoria própria

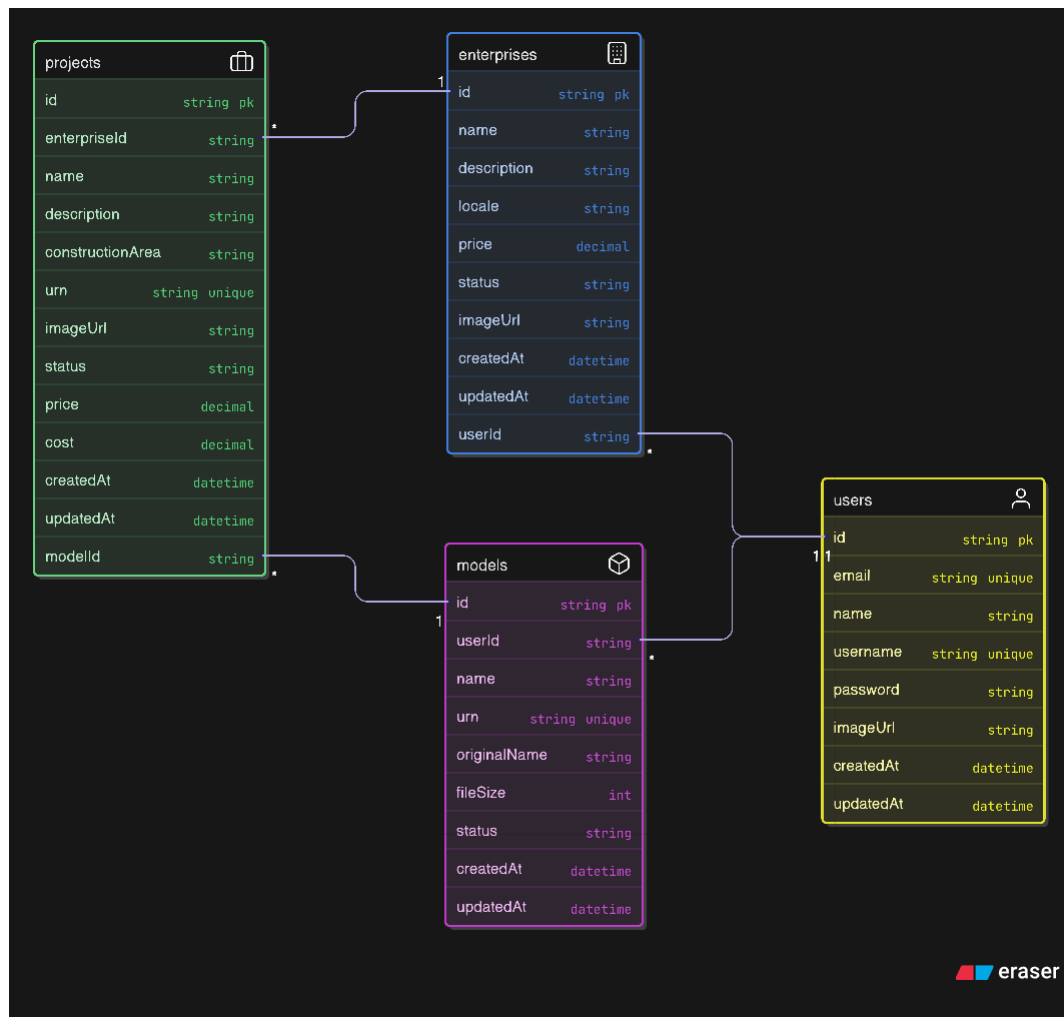
Figura 7 - Modelo Lógico



Fonte: Autoria própria

A figura 8 representa toda a modelagem do banco de dados e suas respectivas tabelas e relacionamentos.

Figura 8 - Modelagem do banco de dados



Fonte: Autoria própria

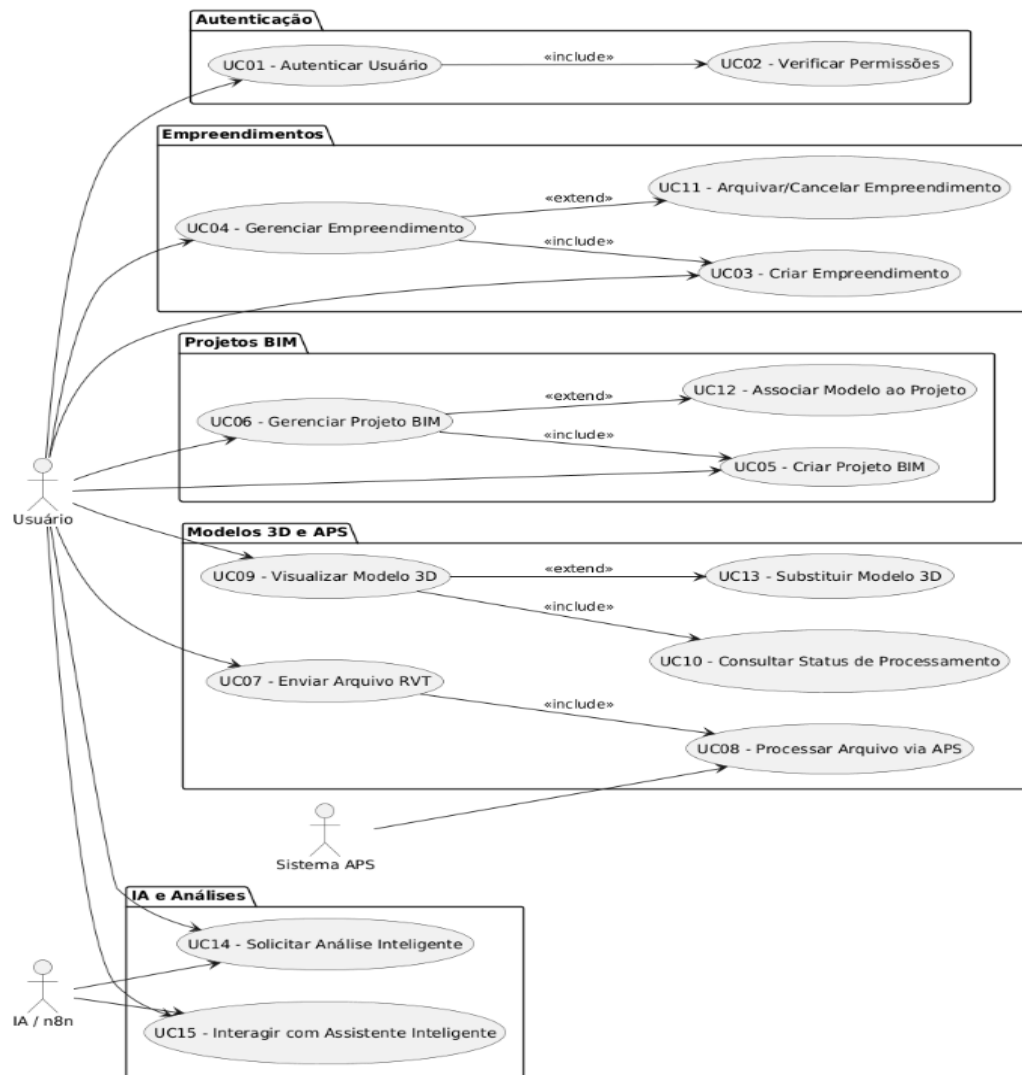
4.3. Modelagem UML

4.3.1 Diagrama de Casos de Uso

O Diagrama de Casos de Uso é uma ferramenta da UML (Unified Modeling Language) essencial para visualizar as interações entre os atores (usuários) do sistema e as funcionalidades que ele oferece (Larman, 2004). Ele representa o comportamento funcional do sistema a partir da perspectiva do usuário, identificando o que o sistema deve fazer sem detalhar como as tarefas serão executadas. Para a

aplicação web proposta, a Figura 9 ilustra as interações e os principais casos de uso que envolvem o Ator Principal, responsável pela gestão dos projetos e comunicação com a IA.

Figura 9 - Diagrama de Casos de Uso do Sistema



Fonte: Autoria própria

4.4. Design de Interface (UI/UX)

4.4.1 Identidade Visual

A Identidade Visual é um componente estratégico no desenvolvimento de software, especialmente em aplicações web que buscam estabelecer credibilidade,

usabilidade e uma experiência de usuário memorável (UX). Vai além da mera estética; é a representação gráfica e sensorial dos valores, propósitos e funcionalidade do sistema.

No contexto desta aplicação web, que integra tecnologias avançadas como BIM, Autodesk Platform Services e Inteligência Artificial, a Identidade Visual tem o papel crucial de transmitir confiabilidade, inovação e profissionalismo. Ela é composta por elementos-chave que, juntos, garantem a coerência e o reconhecimento da plataforma, facilitando a navegação e o engajamento do usuário. Os principais elementos da Identidade Visual abordados neste projeto são:

1. **Logotipo:** O símbolo visual primário da aplicação. Ele deve ser simples, relevante ao setor de Arquitetura/Engenharia e capaz de representar a fusão entre a construção (BIM) e a tecnologia (IA/Web).
2. **Paleta de Cores:** A seleção de cores que definirá o tom emocional da interface. Cores adequadas podem melhorar a legibilidade, organizar a informação e guiar a atenção do usuário para elementos críticos.
3. **Tipografia:** As fontes escolhidas para o sistema, que impactam diretamente a clareza e a hierarquia do conteúdo.

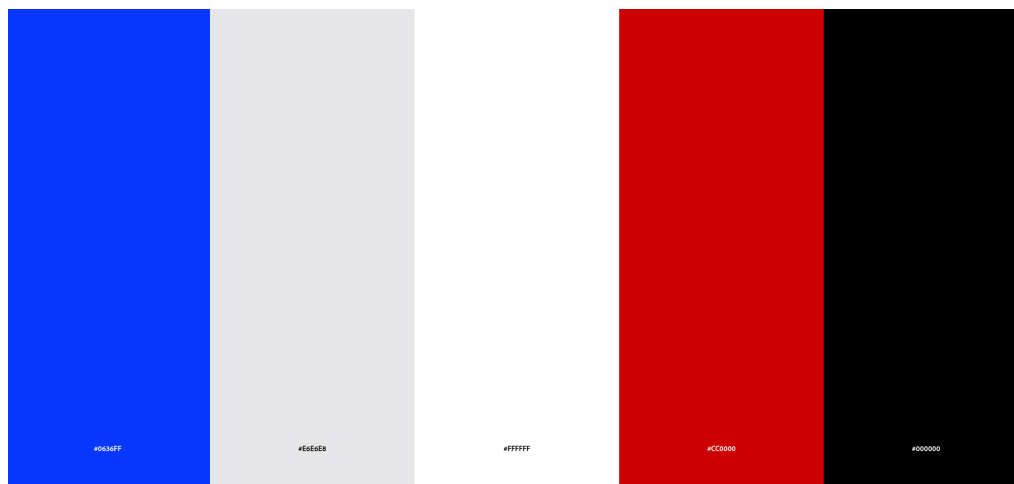
A seguir, serão apresentados o logotipo e a paleta de cores desenvolvidos para esta plataforma, servindo como base para a criação dos protótipos de interface.

Figura 10 - Logo do projeto



Fonte: Autoria própria

Figura 11 - Paleta de cores do projeto



Fonte: Autoria própria

4.4.2 Protótipos

A fase de prototipação é crucial para traduzir a Identidade Visual e os Requisitos de Sistema em uma representação tangível e interativa. Os Wireframes (protótipos de baixa fidelidade) são o primeiro passo, atuando como o esqueleto da aplicação. Eles definem a estrutura, a hierarquia e o layout do conteúdo, focando na funcionalidade e no fluxo de navegação, sem a distração de elementos visuais complexos.

Posteriormente, a evolução para protótipos de alta fidelidade incorpora a Paleta de Cores, a Tipografia e os componentes visuais definidos na Identidade Visual. Esses protótipos simulam a experiência final do usuário (UX), permitindo validar o design, o fluxo de trabalho e a usabilidade antes da implementação final do código.

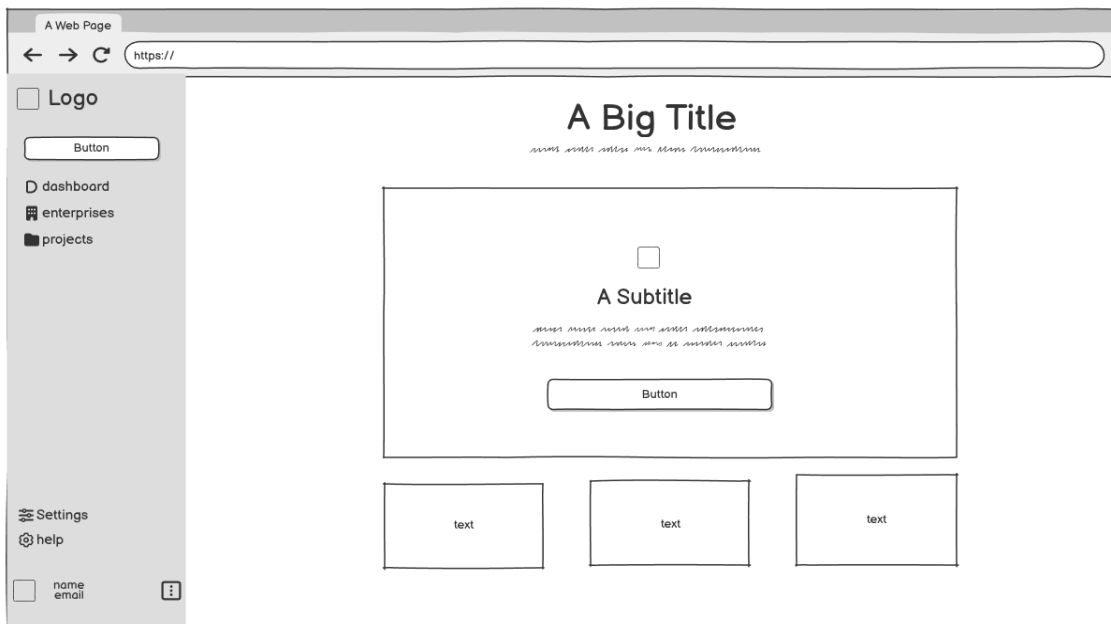
A seguir, são apresentados os wireframes das telas principais da aplicação, seguidos pelo Guia de Estilos, que demonstra a aplicação da Identidade Visual nos componentes da interface.

Figura 12 - Protótipo Tela de Login



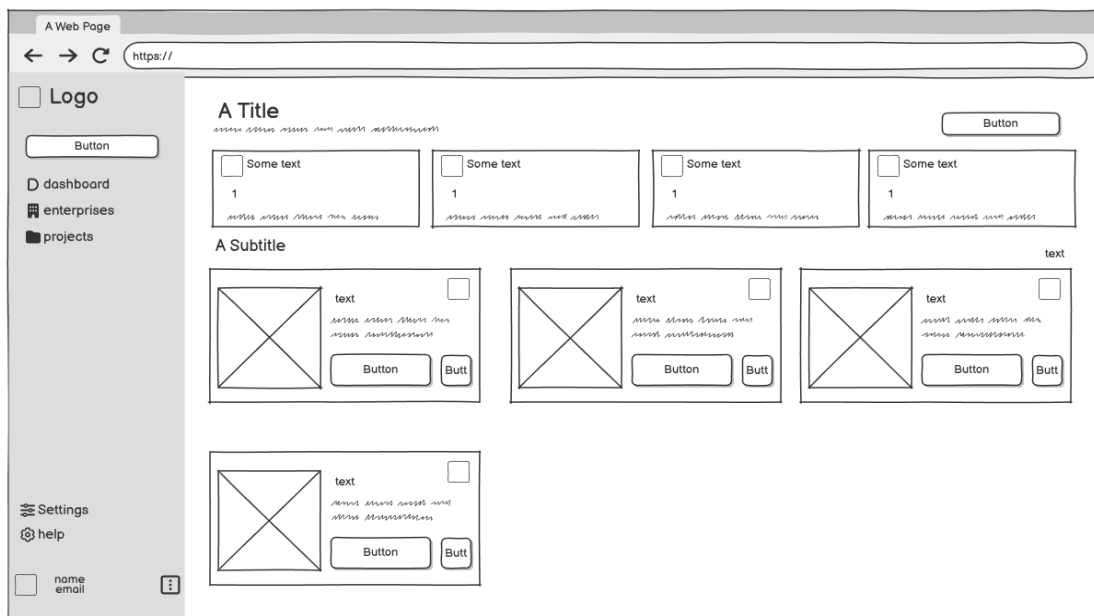
Fonte: Autoria própria

Figura 13 - Protótipo Tela de Inicial



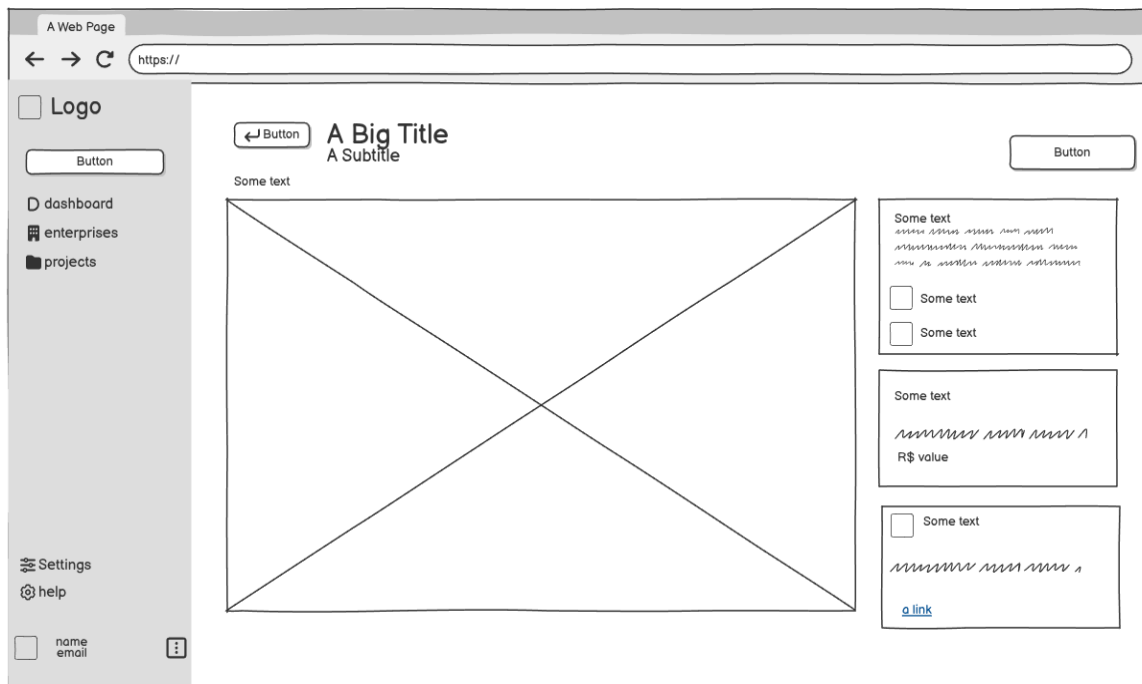
Fonte: Autoria própria

Figura 14 - Protótipo Tela de dashboard



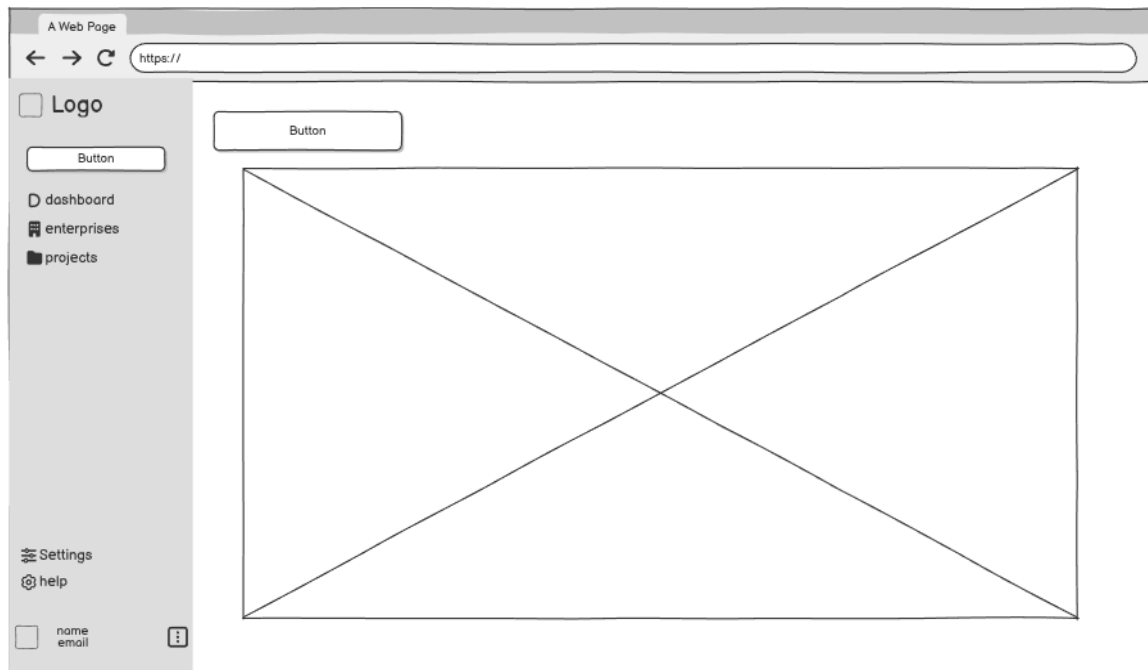
Fonte: Autoria própria

Figura 15 - Protótipo Tela de visualizar empreendimento



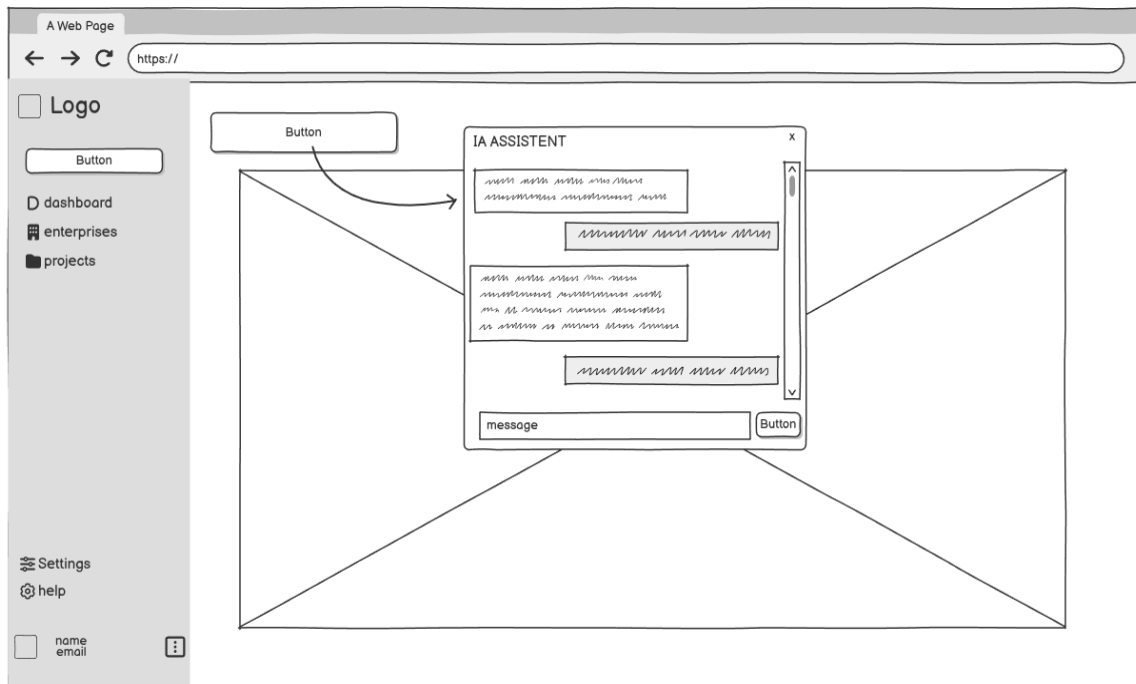
Fonte: Autoria própria

Figura 16 - Protótipo Tela de visualizar modelo 3D



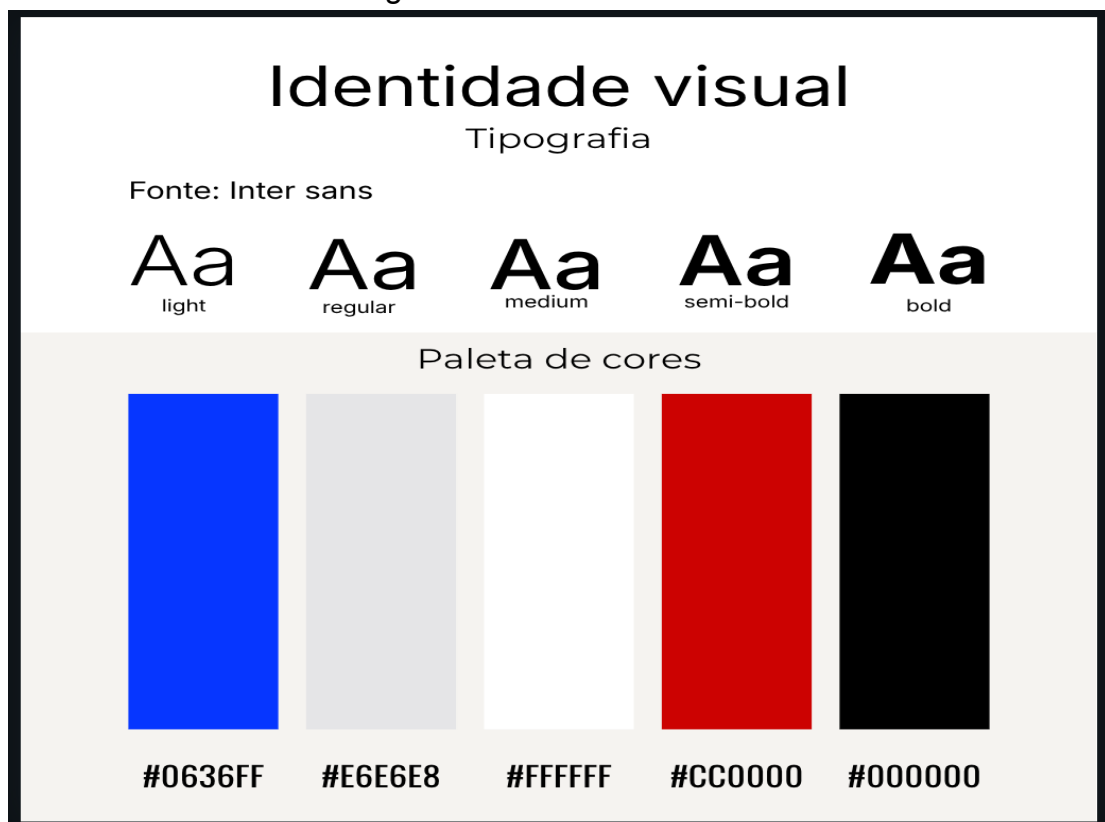
Fonte: Autoria própria

Figura 17 - Protótipo Tela de visualizar modelo 3D com chatbot



Fonte: Autoria própria

Figura 18 - Identidade Visual



Fonte: Autoria própria

5. DESENVOLVIMENTO E IMPLEMENTAÇÃO

O ambiente de desenvolvimento da aplicação foi estruturado para garantir produtividade, escalabilidade e facilidade de manutenção. O backend foi construído com NestJS, utilizando TypeScript e executado em ambiente containerizado via Docker, permitindo padronização da execução e isolamento dos serviços. O banco de dados utilizado é PostgreSQL, também executado como container, facilitando a orquestração e consistência entre ambientes.

No frontend foi utilizado Next.js, com suporte a Server Components, React, Tailwind e ShadCN para a interface. A comunicação entre front e back é feita via API REST autenticada por JWT. Durante o desenvolvimento foram utilizadas ferramentas como VSCode, Postman, Node.js, além de serviços externos como APS (Autodesk Platform Services) para visualização e manipulação de modelos BIM. O controle de versão foi mantido com Git e GitHub.

5.1. Ambiente de Desenvolvimento

O ambiente de desenvolvimento adotado para a construção da aplicação foi planejado de forma a garantir padronização, reprodutibilidade e eficiência no ciclo de desenvolvimento. O backend foi implementado utilizando o framework NestJS, escrito em TypeScript, e executado em ambiente containerizado por meio do Docker, possibilitando isolamento dos serviços, facilidade na configuração e maior confiabilidade durante a implantação e testes. O sistema utiliza PostgreSQL como banco de dados principal, igualmente configurado em container, assegurando consistência e integridade entre os diferentes ambientes.

No frontend, empregou-se o framework Next.js, que oferece suporte a Server Components, integração fluida com React e uma estrutura moderna para desenvolvimento de interfaces dinâmicas. Para construção da interface foram utilizados Tailwind CSS e o conjunto de componentes ShadCN, permitindo maior consistência visual e produtividade. A comunicação entre frontend e backend ocorre via API REST, utilizando JWT como mecanismo de autenticação e autorização. Além disso, durante o processo de desenvolvimento foram utilizadas ferramentas como

VS Code, Postman e Git, possibilitando versionamento, testes e organização das atividades.

5.2. Estrutura do Projeto

A estrutura da aplicação foi organizada de acordo com uma arquitetura modular, contemplando separação clara entre camadas de apresentação, lógica de negócio e persistência de dados.

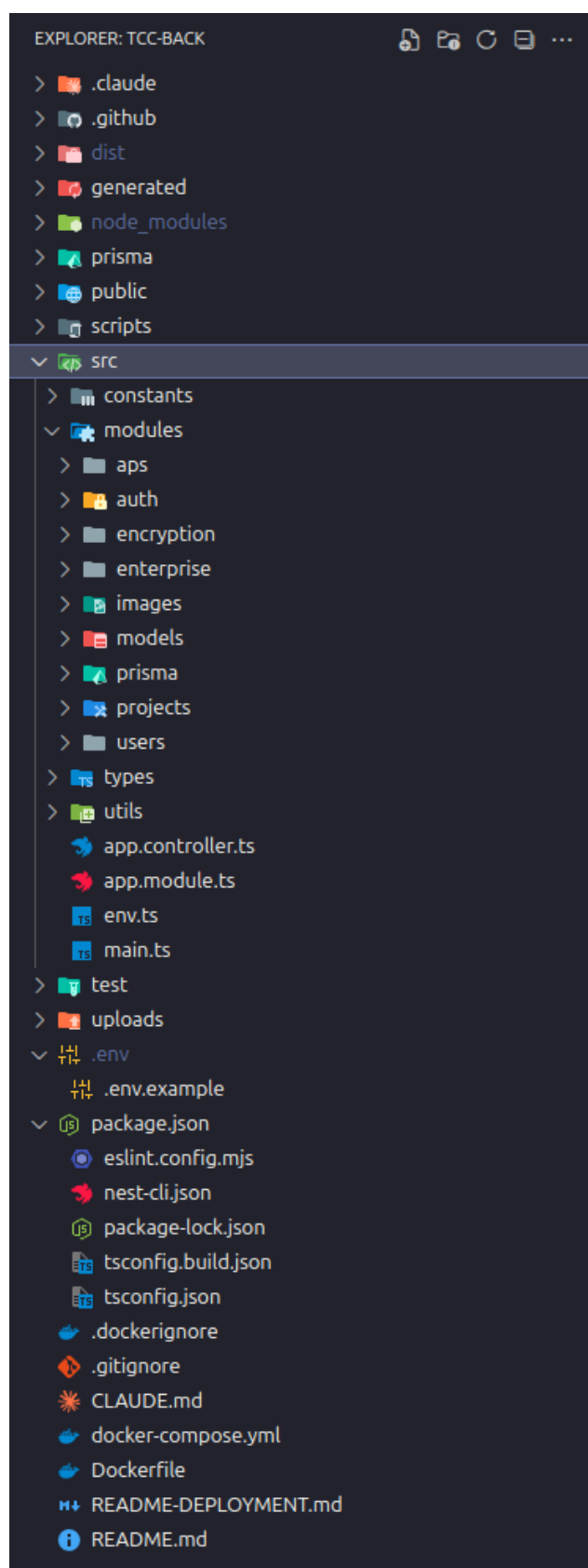
Backend

O backend foi construído seguindo o padrão modular do NestJS, no qual cada módulo é responsável por uma funcionalidade específica do sistema. Entre os principais módulos desenvolvidos estão: módulo de autenticação, módulo de upload de arquivos, módulo de integração com o Autodesk Platform Services (APS) e módulo de comunicação com a API do ChatGPT. O módulo de upload foi responsável pela validação dos arquivos enviados, incluindo modelos tridimensionais e imagens, assegurando conformidade com os tipos permitidos e realizando o armazenamento temporário antes de enviá-los ao APS.

Os serviços de integração com o APS foram projetados para realizar autenticação, envio de arquivos, conversão dos modelos para formatos adequados à visualização e consulta de metadados e propriedades BIM. Dessa forma, o backend atua como intermediário entre o frontend, o APS e a API de IA utilizada para análise.

A Figura 19 ilustra a estrutura de pastas e sua organização, a qual segue as boas práticas do DDD (Domain-Driven Design) com a devida separação de domínio e a aplicação da Clean Architecture, visando garantir a independência de bibliotecas e a implementação dos serviços.

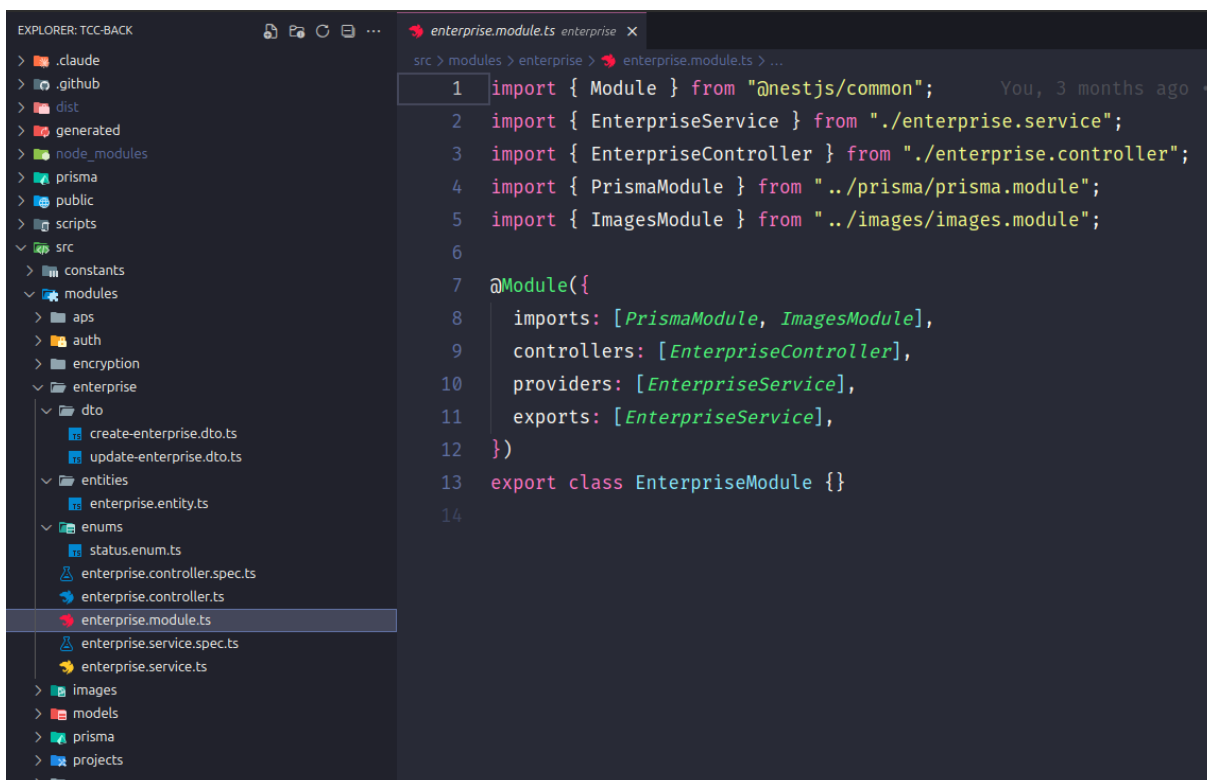
Figura 19 - Estrutura de pastas do backend



Fonte: Autoria própria

Na figura 20 temos um trecho de código onde temos a separação dos respectivos arquivos de acordo com o clean architectures é um trecho de código onde temos a parte de inversão de dependências.

Figura 20 - Modularização e trecho de código de inversão de dependência



```
1 import { Module } from "@nestjs/common";
2 import { EnterpriseService } from "../enterprise.service";
3 import { EnterpriseController } from "../enterprise.controller";
4 import { PrismaModule } from "../prisma/prisma.module";
5 import { ImagesModule } from "../images/images.module";
6
7 @Module({
8   imports: [PrismaModule, ImagesModule],
9   controllers: [EnterpriseController],
10  providers: [EnterpriseService],
11  exports: [EnterpriseService],
12 })
13 export class EnterpriseModule {}
14
```

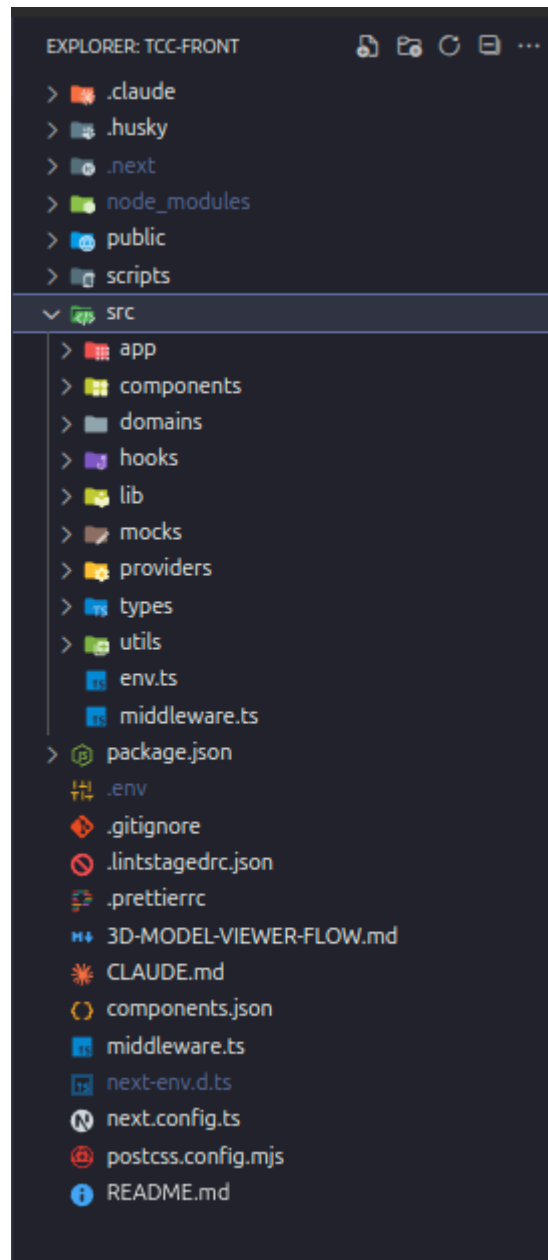
Fonte: Autoria própria

Frontend

O frontend foi implementado em Next.js, estruturado em páginas e componentes reutilizáveis que permitem modularidade e fácil manutenção. A aplicação organiza suas funcionalidades em interfaces distintas, tais como área de upload de modelos BIM, área de visualização utilizando o APS Viewer e páginas destinadas à apresentação das análises geradas pela inteligência artificial. As interfaces foram desenvolvidas com foco em usabilidade, responsividade e clareza visual, utilizando Tailwind e ShadCN como base estilística.

Na imagem abaixo temos a estruturação e organização do projeto frontend, onde temos a separação por domínios, utilizado no Domain Driven Design (DDD).

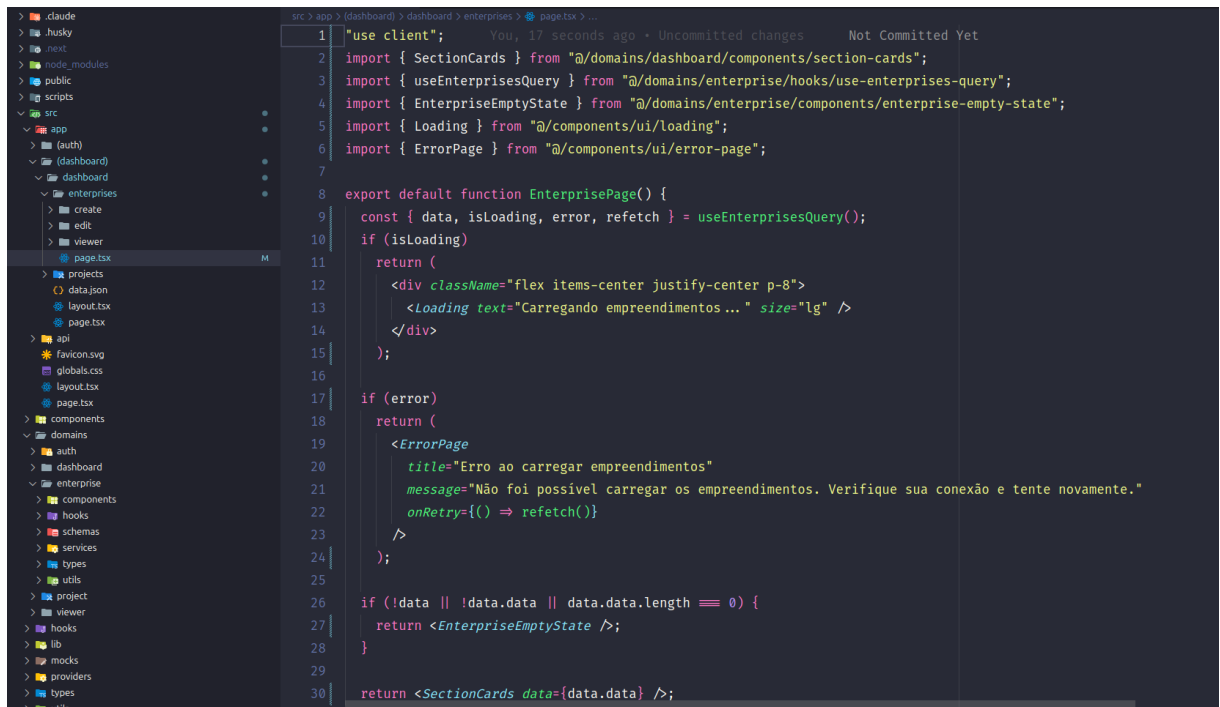
Figura 21 - Estrutura de pastas do frontend



Fonte: Autoria própria

Abaixo está um segmento de código que ilustra a implementação da tela principal de empreendimentos. Adotamos o hook pattern com o intuito de garantir que a página seja escalável e simplificar a execução de testes unitários, de integração e E2E. Além disso podemos visualizar a organização dos arquivos e pastas desse segmento.

Figura 22 - Organização e trecho de código da tela de empreendimentos

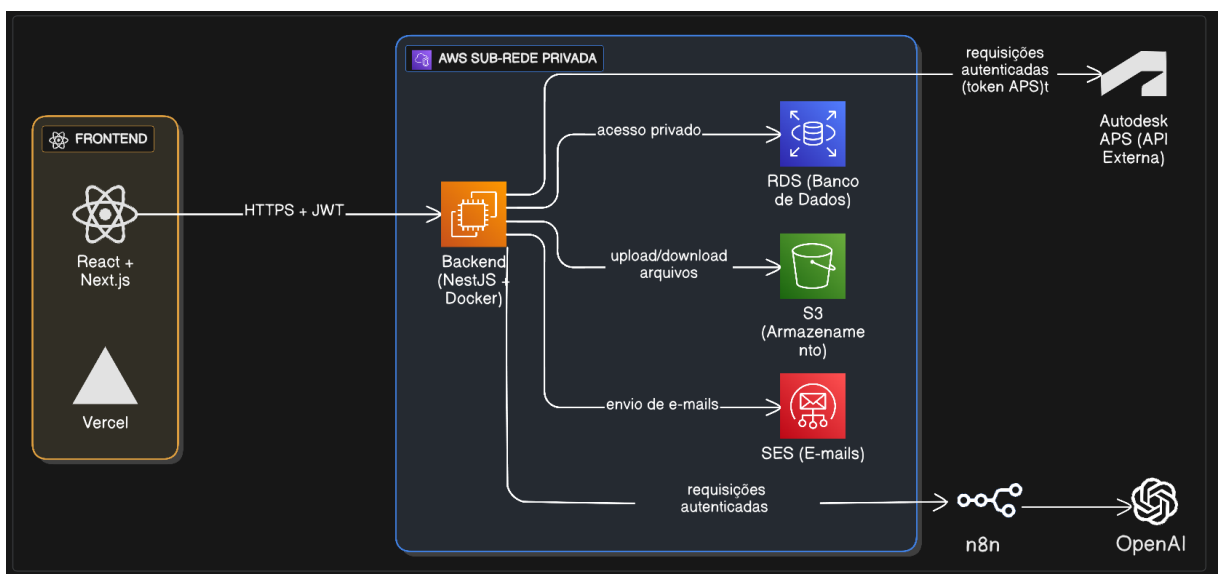


```
1 "use client";
2 import { SectionCards } from "@domains/dashboard/components/section-cards";
3 import { useEnterprisesQuery } from "@domains/enterprise/hooks/use-enterprises-query";
4 import { EnterpriseEmptyState } from "@domains/enterprise/components/enterprise-empty-state";
5 import { Loading } from "@components/ui/loading";
6 import { ErrorPage } from "@components/ui/error-page";
7
8 export default function EnterprisePage() {
9   const { data, isLoading, error, refetch } = useEnterprisesQuery();
10   if (isLoading)
11     return (
12       <div className="flex items-center justify-center p-8">
13         <Loading text="Carregando empreendimentos ..." size="lg" />
14       </div>
15     );
16
17   if (error)
18     return (
19       <ErrorPage
20         title="Erro ao carregar empreendimentos"
21         message="Não foi possível carregar os empreendimentos. Verifique sua conexão e tente novamente."
22         onRetry={() => refetch()}
23       />
24     );
25
26   if (!data || !data.data || data.data.length === 0) {
27     return <EnterpriseEmptyState />;
28   }
29
30   return <SectionCards data={data.data} />;
31 }
```

Fonte: Autoria própria

A figura 23 representa de maneira simplificada a infraestrutura utilizada, demonstrando a interação entre os componentes frontend (Next.js), backend (NestJS/PostgreSQL), os serviços de nuvem (AWS S3, AWS SES) e as APIs externas (Autodesk Platform Services e ChatGPT/n8n).

Figura 23 - Infraestrutura do sistema



Fonte: Autoria própria

5.3. Implementação das Funcionalidades Principais

A aplicação contempla um conjunto de funcionalidades essenciais para o processamento, visualização e análise inteligente de modelos BIM.

Foi desenvolvido um fluxo completo para o upload de arquivos, envolvendo validação inicial, armazenamento temporário e envio ao Autodesk Platform Services. A implementação inclui tratamento de erros para evitar inconsistências, como arquivos armazenados sem sucesso na criação de registros no sistema.

Após o envio, o backend solicita ao APS a conversão dos arquivos para formatos adequados ao consumo no visualizador tridimensional. Além da conversão, foram implementadas rotinas de consulta a metadados dos modelos BIM, permitindo que propriedades estruturais, geométricas e materiais sejam extraídas e organizadas.

Análise via ChatGPT

As informações extraídas do modelo BIM são utilizadas para compor prompts estruturados encaminhados à API do ChatGPT. O sistema envia ao modelo de linguagem dados relevantes, especificações do projeto e objetivos definidos pelo usuário, de modo que a IA retorne análises interpretativas, recomendações, possíveis inconsistências e sugestões de melhoria. A implementação considera aspectos como controle de tokens, organização de prompts e validação das respostas.

5.4. Segurança e Controle de Acesso

A segurança da aplicação foi estruturada considerando mecanismos adequados de autenticação, autorização e proteção de dados sensíveis. O sistema utiliza JSON Web Tokens (JWT) como mecanismo principal de autenticação, garantindo que apenas usuários devidamente autenticados possam acessar recursos protegidos da API.

Além disso, foram implementadas práticas de segurança como hashing de senhas, controle de CORS, validação robusta de entradas e utilização de variáveis de ambiente para armazenamento de credenciais e chaves sensíveis. A arquitetura também considera a segregação de serviços e a execução containerizada, contribuindo para o isolamento e proteção do ambiente de execução.

5.5. Integração com APIs Externas

A aplicação integra dois serviços externos fundamentais para seu funcionamento:

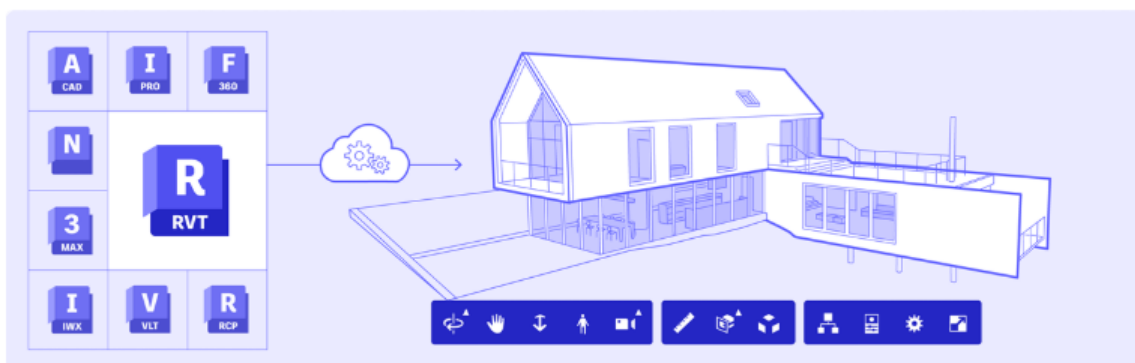
Autodesk Platform Services (APS)

O APS é responsável pelo processamento e visualização dos modelos BIM. A aplicação faz uso de suas principais APIs, como Authentication, Model Derivative e Viewer. Entre as operações realizadas estão:

- envio de arquivos para um bucket;
- conversão para formatos visualizáveis;
extração de propriedades e metadados dos modelos;
- renderização utilizando o viewer incorporado ao frontend.

Essa integração permite que o sistema receba, processe e apresente modelos BIM de forma eficiente e interativa. Para a visualização dos modelos 3D é utilizado o SDK Viewer mostrado na figura 24.

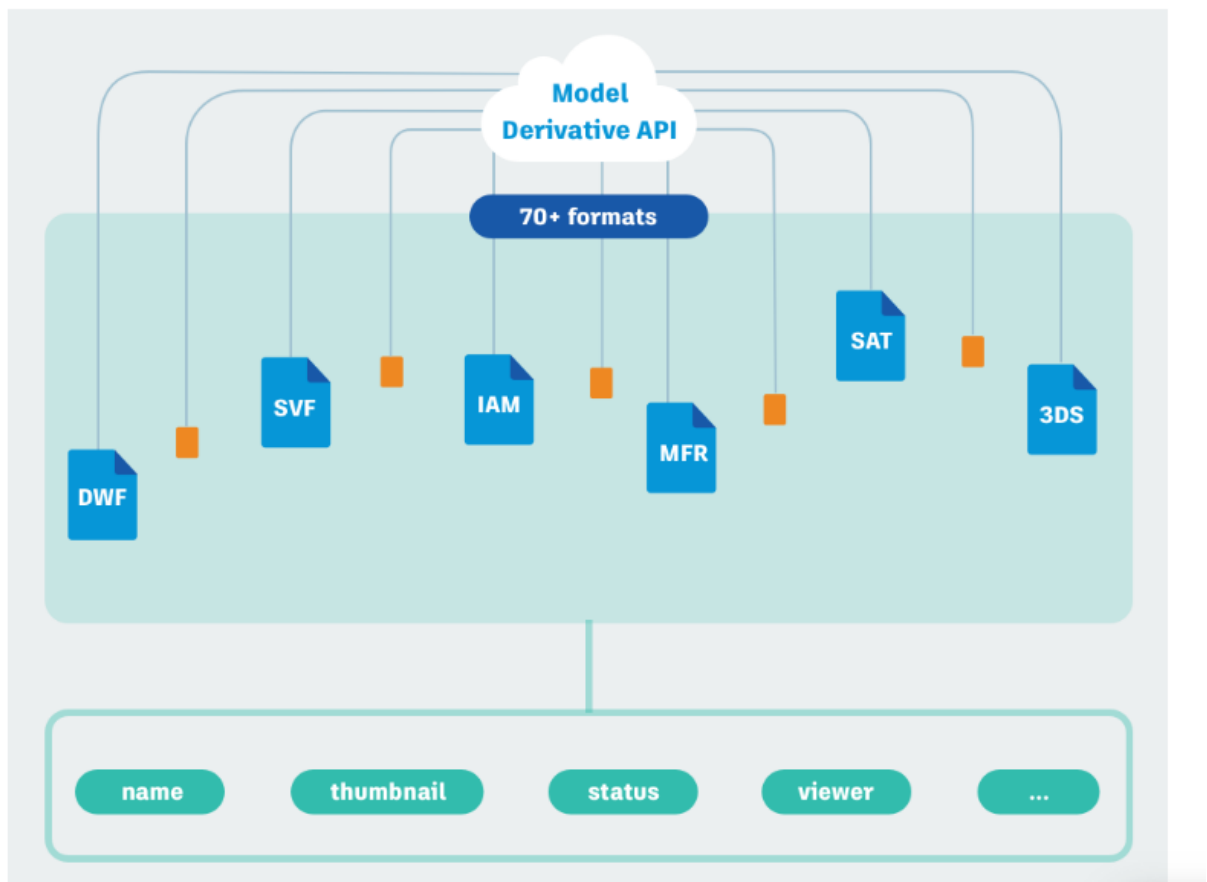
Figura 24 - Autodesk Viewer SDK



Fonte: Autodesk Services

Outra API de grande relevância para o desenvolvimento deste projeto foi a Model Derivative API da Autodesk, que se destaca por sua capacidade de processar diversos formatos de arquivo de engenharia e construção, permitindo a extração de dados e a conversão para formatos visualizáveis. A Figura 25 ilustra essa API.

Figura 25 - Autodesk Model Derivative API



Fonte: Autodesk Services

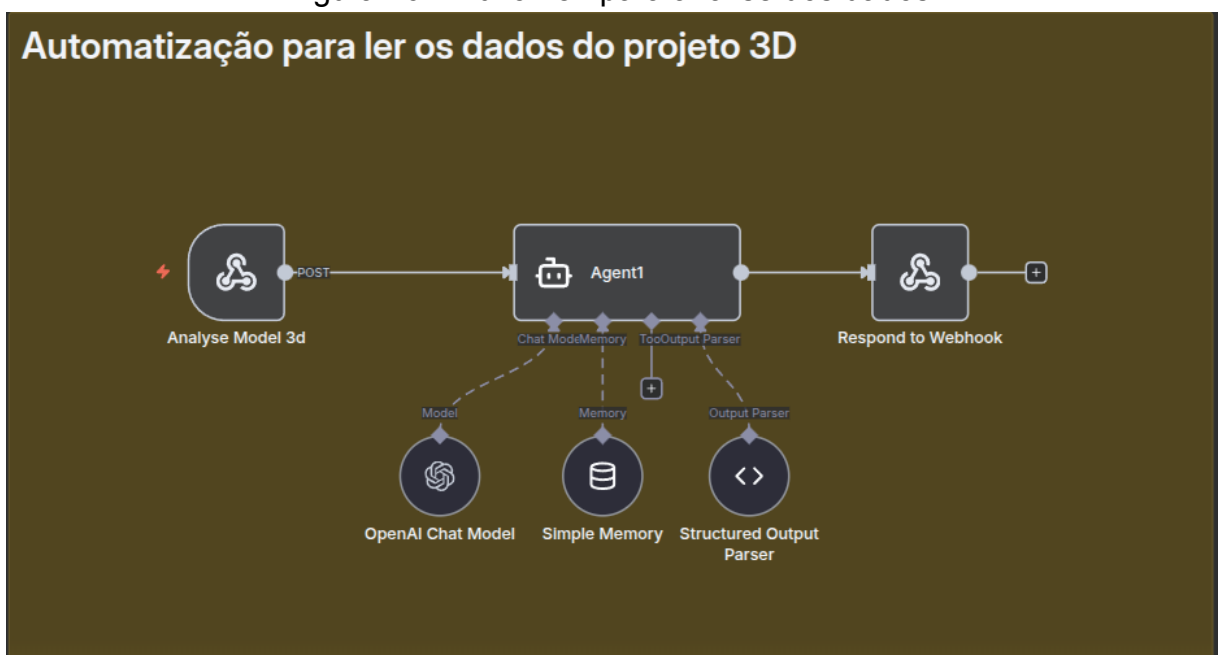
API do Chat GPT e N8n

A API do ChatGPT é utilizada para gerar análises inteligentes com base nas informações extraídas dos modelos BIM. A aplicação envia prompts detalhados contendo dados técnicos, especificações do projeto e objetivos da consulta. O retorno da IA é interpretado e exibido ao usuário como relatórios, recomendações e observações relevantes para o contexto arquitetônico ou construtivo.

A integração com ambas as APIs foi encapsulada em serviços específicos no backend, garantindo organização, modularidade e a possibilidade de reaproveitamento de código em futuras expansões do sistema seguindo os princípios de DDD e Clean Architecture.

O fluxo ilustrado detalha o processamento de dados da API da Autodesk. Após extração pelo backend, os dados brutos são enviados para a plataforma low-code n8n, que atua como orquestrador central. No n8n, são realizadas a transformação e enriquecimento dos dados. A análise subsequente, utilizando módulos especializados, gera insights e métricas. Estes resultados analíticos são temporariamente armazenados em memória (cache) para garantir rápido acesso e processamento, otimizando a performance na geração de mensagens.

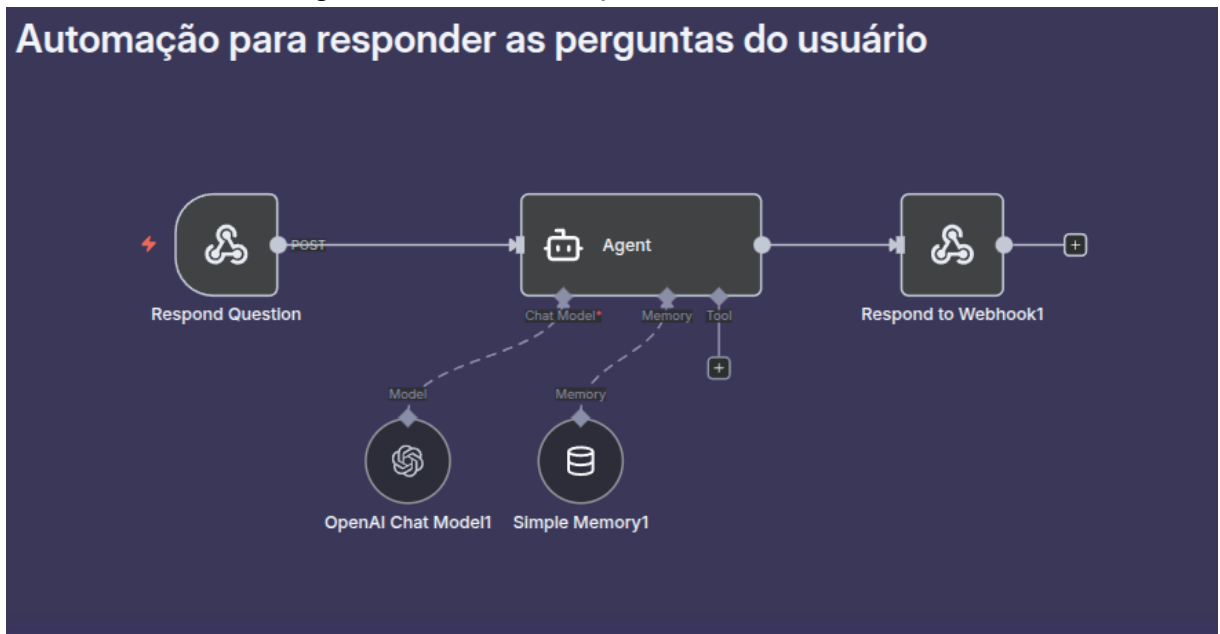
Figura 26 - Fluxo n8n para análise dos dados



Fonte: Autoria própria

A figura a seguir representa o fluxo de interação entre o usuário e o modelo de inteligência artificial da OpenAI. Nesse processo, a aplicação estabelece comunicação por meio de chamadas HTTP, permitindo que o usuário realize consultas e receba respostas geradas pelo modelo. As interações são baseadas nos dados previamente enviados, processados e analisados pelo sistema, possibilitando uma conversação contextualizada e coerente com as informações do projeto.

Figura 27 - Fluxo n8n para análise dos dados

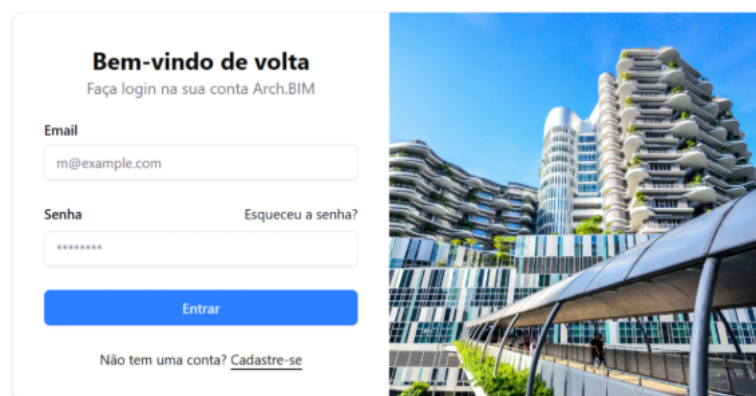


Fonte: Autoria própria

6. RESULTADOS

Nesta seção, são apresentadas algumas imagens das principais telas do sistema, com o objetivo de ilustrar o funcionamento das funcionalidades implementadas e demonstrar a aplicação prática das tecnologias descritas anteriormente. A figura 28 mostra a tela real de login da aplicação, contendo o input do e-mail e senha e botão de acesso.

Figura 28 - Tela de Login

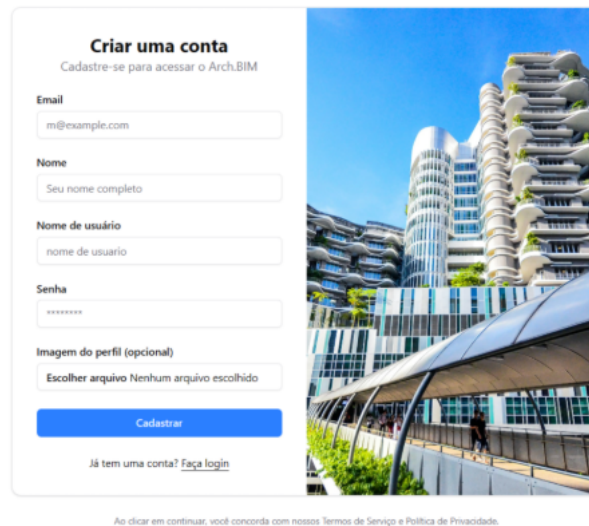


Ao clicar em continuar, você concorda com nossos [Termos de Serviço](#) e [Política de Privacidade](#).

Fonte: Autoria própria

Na figura 28 vemos a tela de cadastro, para que um novo usuário possa se cadastrar e ter acesso a todas as funcionalidades.

Figura 28 – Tela de cadastro



A tela de cadastro do Arch.BIM apresenta um formulário para criar uma nova conta. O formulário contém campos para Email, Nome, Nome de usuário, Senha e Imagem do perfil (opcional). Um botão azul "Cadastrar" está localizado abaixo dos campos. À direita do formulário, há uma imagem de uma moderníssima edificação residencial com varandas e jardins verticais. Abaixo do formulário, há um link para "Faça login" e uma linha de texto informando que ao clicar em continuar, o usuário concorda com os Termos de Serviço e Política de Privacidade.

Criar uma conta
Cadastre-se para acessar o Arch.BIM

Email
m@example.com

Nome
Seu nome completo

Nome de usuário
nome de usuario

Senha

Imagem do perfil (opcional)
Escolher arquivo Nenhum arquivo escolhido

Cadastrar

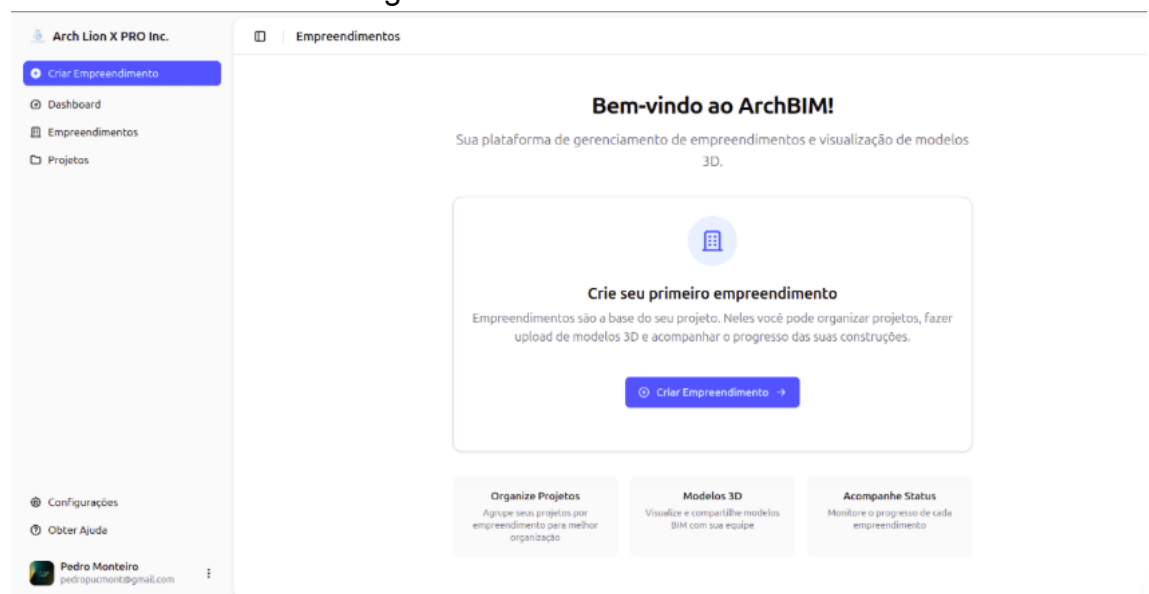
Já tem uma conta? [Faça login](#)

Ao clicar em continuar, você concorda com nossos [Termos de Serviço](#) e [Política de Privacidade](#).

Fonte: Autoria própria

Após realizar o cadastro na plataforma e efetuar o login com sucesso, o usuário é redirecionado para a tela inicial do sistema. Nessa tela, são exibidos os empreendimentos previamente cadastrados pelo usuário ou pela empresa. A figura 29 apresenta um exemplo da interface inicial.

Figura 29 – Tela Inicial



Fonte: Autoria própria

Para cadastrar um empreendimento o usuário poderá utilizar um dos botões de ação presentes na interface e no menu lateral, onde será redirecionado para a interface de cadastro de empreendimentos mostrado na figura 30 abaixo.

Figura 30 – Tela de cadastro de empreendimento

A interface de cadastro de novo empreendimento apresenta um formulário com os seguintes campos:

- Informações do Empreendimento:**
 - Título ***: Campo de texto com o exemplo "Ex: Residencial Copas do Bosque".
 - Descrição**: Campo de texto com o placeholder "Descreva as características e diferenciais do empreendimento...".
 - Localização**: Campo de texto com o exemplo "Ex: Setor Marista, Goiânia - GO".
 - Preço**: Campo de texto com o exemplo "Ex: 450.000".
 - Status ***: Menu suspenso com a opção "Planejamento".
- Imagem do Empreendimento**: Seção com o texto "Upload de arquivo" e um botão "Escolher arquivo".

Na base do formulário, há uma linha com o texto "Campos obrigatórios marcados com *" e um botão azul "Cadastrar Empreendimento".

À esquerda, o menu lateral contém: "Criar Empreendimento" (destacado), "Dashboard", "Empreendimentos" e "Projetos".

Na barra inferior, há o nome de usuário "Pedro Monteiro" e o e-mail "pedro.monteiro@gmail.com".

Fonte: Autoria própria

Após todos os campos obrigatórios preenchidos, o usuário pode concluir o cadastro clicando no botão de criar empreendimento, que irá redirecionar para a tela inicial, onde ele poderá ver seu novo empreendimento na figura 31.

Figura 31 – Tela inicial com empreendimento

A tela inicial (Dashboard) apresenta uma visão geral dos empreendimentos e projetos. No topo, há um botão azul "Novo Empreendimento".

Abaixo, há quatro cartões de resumo:

- Empreendimentos**: 1 total cadastrados.
- Projetos**: 0 em todos os empreendimentos.
- Em Andamento**: 0 empreendimentos ativos.
- Concluídos**: 0 empreendimentos finalizados.

Abaixo dos cartões, há uma seção "Seus Empreendimentos" com um card para o "Empreendimento OPUS". O card mostra uma imagem de uma casa, o status "Planejamento", o preço "R\$ 1.000.000,00" e um botão "Ver Projetos".

À esquerda, o menu lateral contém: "Criar Empreendimento" (destacado), "Dashboard", "Empreendimentos" e "Projetos".

Na barra inferior, há o nome de usuário "Pedro Monteiro" e o e-mail "pedro.monteiro@gmail.com".

Fonte: Autoria própria

A figura 32 apresenta a página de cadastro de um projeto, onde nela o usuário poderá anexar o seu arquivo 3d feito no revit na plataforma.

Figura 32 – Tela cadastro de projeto

Arch Lion X PRO Inc.

Novo Projeto

[Voltar](#) **Cadastrar Novo Projeto**
Preencha as informações abaixo para registrar um novo projeto com modelo 3D

Informações do Projeto

Nome do Projeto *
Ex: Torre Residencial A

Descrição
Descreva as características e detalhes do projeto...

Empreendimento *
Selecione o empreendimento

Status *
Planejamento

Área de Construção
Ex: 85m²

Preço
Ex: 450.000

Custo de Construção
Ex: 350.000

Imagem do Projeto
Upload de arquivo
Escolher arquivo

Modelo 3D (.rvt ou .ifc)
Upload de modelo 3D
Escolher arquivo

Formatos aceitos: .rvt (Revit) ou .ifc (Autodesk Foundation Clipping)

Configurações
Obter Ajuda
Pedro Monteiro
pedromonteiro@gmail.com

Fonte: Autoria própria

A figura 33 representa a tela de detalhes para que o usuário consiga ver rapidamente todos os detalhes do empreendimento e os projetos vinculados a ele.

Figura 33 – Tela de visualizar o empreendimento

Arch Lion X PRO Inc.

Visualizar Empreendimento

[Voltar](#) **Empreendimento OPUS** [Editar](#)
Setor Nova Vila

Imagem do Empreendimento

Informações Básicas

Descrição
Este empreendimento é para um empresário em particular que irá construir uma casa na praia

Localização
Setor Nova Vila

Status
Planejamento

Informações Financeiras

Valor do Empreendimento
R\$ 1.000.000,00

Projetos

Total de Projetos
0

[Ver todos os projetos](#)

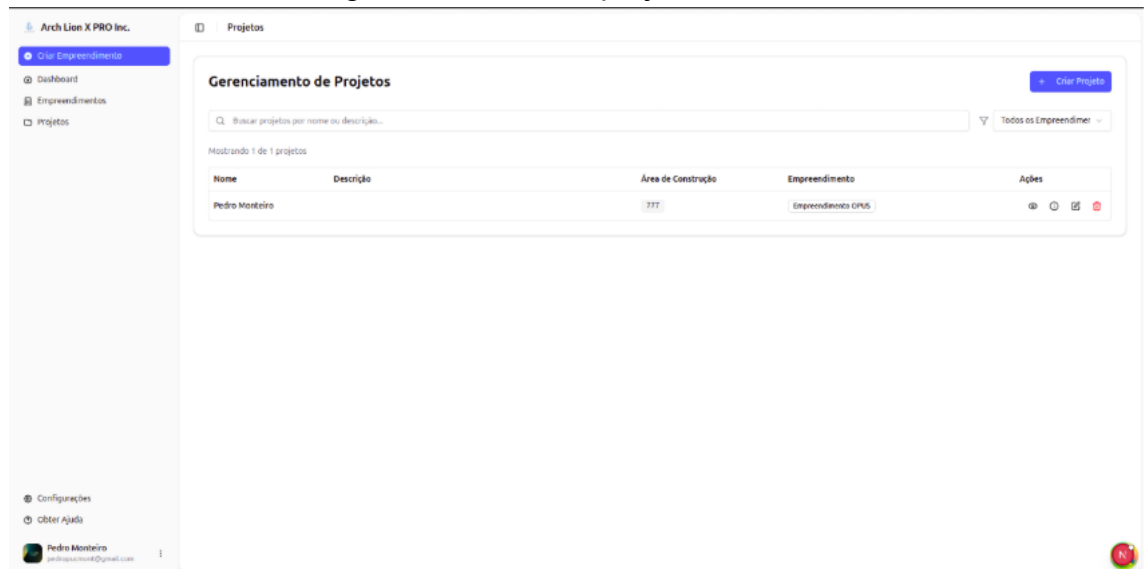
Configurações
Obter Ajuda
Pedro Monteiro
pedromonteiro@gmail.com

Fonte: Autoria própria

Visando facilidade a interface foi projetada para que o usuário consiga rapidamente visualizar todos os projetos vinculados ao empreendimento, com isso

ele pode através do card de projetos ser redirecionado para uma interface para ver os projetos em uma tabela e as possíveis ações, como: Visualizar modelo 3d, visualizar detalhes, editar projeto e excluir projeto. Representado na figura 34.

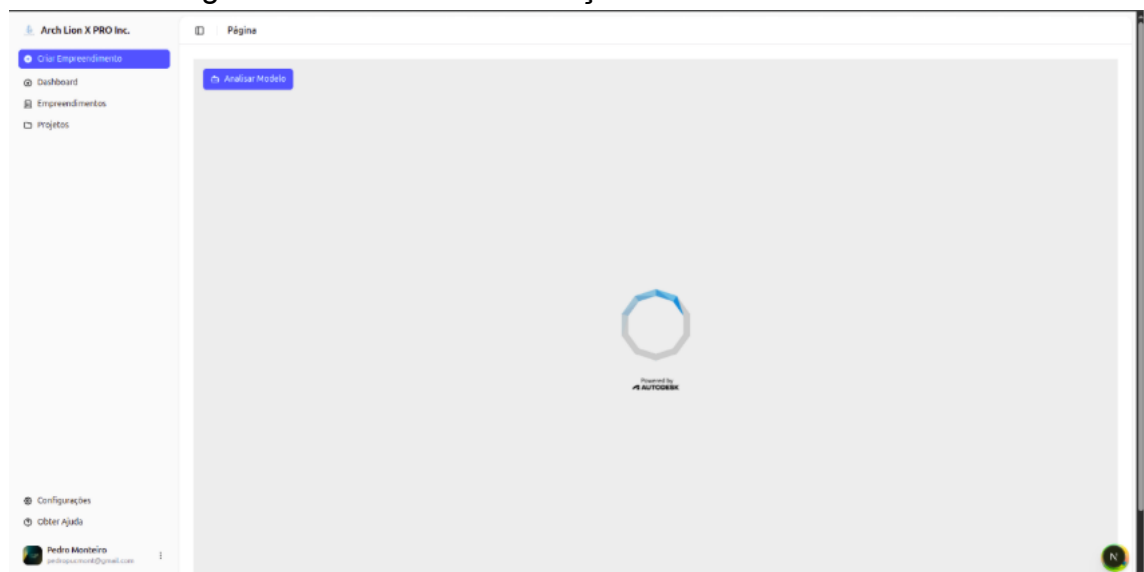
Figura 34 – Tela de projetos



Fonte: Autoria própria

Para visualizar o seu projeto 3D basta clicar no botão de ação visualizar, que será redirecionado para a tela de visualização de modelos, conforme a figura 35 e 16 apresenta abaixo.

Figura 35 – Tela de visualização de modelos 3D



Fonte: Autoria própria

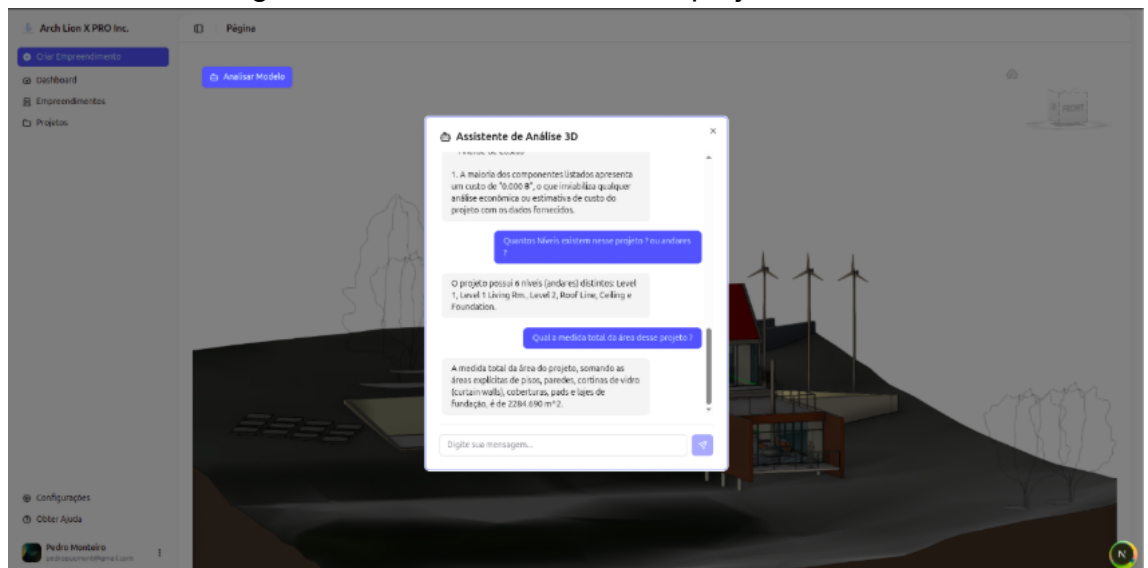
Figura 36 – Tela de visualização com modelo 3D



Fonte: Autoria própria

O usuário poderá utilizar o botão no lado superior esquerdo da tela para que os dados obtidos do projeto 3D sejam enviados para o agente de IA que utiliza do modelo GPT4.1 para realizar as análises e responder às perguntas do usuário de acordo com dados obtidos pelo modelo. A figura 37 mostra uma conversa entre o usuário e o agente de IA sobre o projeto 3D, trazendo insights pré analisados pelo modelo de IA.

Figura 37 – Modal de análise de projetos



Fonte: Autoria própria

O desenvolvimento do sistema foi orientado por decisões técnicas fundamentadas em princípios de segurança, usabilidade, modularidade e escalabilidade, bem como pelo uso de tecnologias modernas aplicadas ao front-end, ao back-end e ao fluxo de inteligência artificial (IA).

A adoção de frameworks, bibliotecas e arquiteturas contemporâneas, como Next.js, TypeScript, React, Nest.js, Prisma ORM, Clean Architecture e Domain-Driven Design (DDD), contribuiu significativamente para a organização e a clareza do código, além de facilitar processos de manutenção e substituição de bibliotecas ou serviços externos. Ademais, a implementação de mecanismos de validação e o uso de autenticação JWT (JSON Web Token) garantiram maior segurança à aplicação em todas as suas camadas, incluindo a plataforma principal, a comunicação com a inteligência artificial e a integração com a API da Autodesk.

6.1. Comparativo: Objetivos e Resultados Alcançados

Os objetivos estabelecidos para o desenvolvimento da aplicação incluíam a integração entre modelos BIM processados pelo Autodesk Platform Services (APS), a visualização desses modelos em ambiente web e a geração de análises inteligentes por meio da API do ChatGPT. Esses objetivos gerais foram atingidos de forma satisfatória, uma vez que o sistema resultante permite o upload de arquivos, sua conversão e visualização no navegador, além da extração e organização dos metadados necessários para posterior análise. A arquitetura proposta também foi implementada conforme planejado, utilizando NestJS no backend e Next.js no frontend, garantindo modularidade, segurança por meio de JWT e comunicação eficiente com serviços externos.

Os resultados obtidos demonstram aderência significativa às metas inicialmente definidas. A aplicação mostrou-se capaz de integrar de maneira fluida as funcionalidades de processamento BIM com a camada de inteligência artificial, fornecendo respostas contextualizadas e coerentes com os dados do modelo. A interface gráfica apresenta boa usabilidade e clareza, e o backend opera de forma estável ao manipular arquivos, realizar autenticação e coordenar chamadas às APIs.

externas. Dessa forma, conclui-se que o sistema alcançou plenamente seus objetivos funcionais e técnicos, validando a viabilidade da solução proposta.

7. CONSIDERAÇÕES FINAIS

7.1. Conclusão

O objetivo geral desse trabalho foi desenvolver uma aplicação para que novos engenheiros, arquitetos, urbanistas e pessoas em geral, consiga ter insights valiosos sobre seus projetos. Por isso uma interface acessível e intuitiva permite com que todos os tipos de usuários consigam dicas valiosas para seu projeto.

O desenvolvimento da aplicação proposta envolveu desafios significativos relacionados à modelagem e estruturação do banco de dados, bem como à elaboração dos diagramas de casos de uso que serviram de base para o entendimento e a organização das funcionalidades do sistema. Ao longo desse processo, foi possível constatar que o desenvolvimento web, aliado às tecnologias de inteligência artificial e aos serviços da Autodesk, apresenta grande versatilidade e capacidade de adaptação, permitindo a construção de soluções eficientes, escaláveis e alinhadas à realidade dos fluxos BIM.

O projeto me permitiu a vivência completa das etapas do ciclo de desenvolvimento de software, como o levantamento e análise de requisitos, passando pela definição da arquitetura, modelagem das regras de negócios, design da interface e implementação das funcionalidades, até a resolução de desafios técnicos específicos relacionados à integração com o Autodesk Platform Services e à comunicação com o ChatGPT utilizando o n8n. Essa experiência resultou em uma aplicação funcional, capaz de analisar dados de modelos BIM e auxiliar na interpretação das informações por meio de interações inteligentes.

Na fase inicial, o levantamento de requisitos desempenhou papel fundamental para o direcionamento de todas as etapas subsequentes. Essa atividade foi conduzida por meio de reuniões com o stakeholder, com foco na compreensão detalhada do problema, na identificação das necessidades operacionais e na definição das funcionalidades essenciais ao sistema.

Diante dos resultados obtidos, conclui-se que a aplicação desenvolvida atingiu os objetivos propostos, ao integrar de forma eficiente tecnologias modernas e

inteligência artificial para aprimorar a análise de dados de projetos BIM. A pesquisa também demonstrou a viabilidade técnica dessa integração, evidenciando seu potencial de uso em ambientes institucionais e corporativos. Espera-se que este trabalho sirva de base para futuras expansões, aprimoramentos e novas abordagens envolvendo IA aplicada a processos BIM.

7.2. Dificuldades Encontradas

O processo de desenvolvimento da aplicação, apesar de bem-sucedido, envolveu desafios técnicos e estruturais significativos, que exigiram adaptação e pesquisa aprofundada:

1. **Defasagem e Adaptação da Documentação do Autodesk Platform Services (APS):** A principal barreira encontrada foi a documentação desatualizada das APIs do APS. Grande parte dos exemplos e tutoriais oficiais estava baseada em *stacks* de tecnologia legadas, demandando um esforço considerável para adaptar os procedimentos de autenticação, upload e conversão de modelos (Model Derivative API) aos frameworks modernos adotados (Next.js e NestJS).
2. **Complexidade na Arquitetura e Integração Multifacetada:** A estruturação da arquitetura para conciliar três sistemas independentes (Frontend/Backend, Autodesk Platform Services e API de IA/n8n) representou um desafio complexo. Foi necessário um planejamento rigoroso para garantir o baixo acoplamento e a consistência do fluxo de dados entre camadas distintas, como:
 - **Upload e Persistência Temporária:** Gerenciamento seguro de arquivos (AWS S3) antes do envio ao APS.
 - **Orquestração Assíncrona:** Sincronização entre os processos de conversão do APS (assíncronos) e a subsequente extração de dados e visualização.
 - **Fluxo de IA:** Garantir que o agente de IA recebesse os prompts contextualizados com os dados estruturados do BIM, exigindo um controle apurado do fluxo no *low-code* n8n.
3. **Tratamento de Erros e Estabilidade de Processos:** Devido à natureza distribuída e à dependência de serviços externos, o tratamento de erros foi

particularmente desafiador. Foi crucial implementar rotinas robustas para lidar com falhas de comunicação com o APS, *timeouts* durante a conversão de modelos volumosos e inconsistências na criação ou atualização de registros no banco de dados, assegurando a estabilidade e a confiabilidade do sistema final.

7.3. Trabalhos Futuros

Considerando a evolução natural da aplicação e as possibilidades oferecidas pelos serviços BIM e pelas tecnologias de inteligência artificial, diversos aprimoramentos podem ser implementados em versões futuras do sistema. Essas melhorias têm potencial para ampliar a precisão das análises, aumentar o valor prático da ferramenta e proporcionar uma experiência mais completa aos profissionais que trabalham com modelos BIM em ambiente web.

Possíveis extensões do sistema incluem:

1. **Integração com APIs de Geolocalização e Dados Climáticos (Google Maps, Google Places e Weather API):** Permitir a coleta automática de informações ambientais, como clima, insolação, fatores urbanos e características geográficas, enriquecendo as análises realizadas pela IA.
2. **Geração Automática de Relatórios (PDF, Excel, ODS ou HTML dinâmico):** Implementar um módulo responsável por consolidar insights, propriedades do modelo e dados relevantes em documentos exportáveis, facilitando a comunicação técnica e o registro formal das análises.
3. **IA Avançada para Detecção Automática de Conflitos (Clash Detection):** Utilizar modelos de IA ou metadados processados pelo APS para identificar colisões, inconsistências geométricas ou problemas estruturais no modelo BIM, aumentando o rigor e a confiabilidade das avaliações.
4. **Ferramentas de Colaboração em Tempo Real:** Criar um ambiente colaborativo semelhante ao Google Docs, no qual múltiplos usuários possam

visualizar, comentar, anotar e solicitar análises simultâneas sobre o mesmo modelo BIM, tornando o processo de revisão mais dinâmico e eficiente.

8. Referências

ALMEIDA, T. G. S. O BIM em escritórios de arquitetura. 2016. 97 f. Dissertação (Mestrado em Arquitetura e Urbanismo) - Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2016. Disponível em: <<http://www.teses.usp.br/teses/disponiveis/18/18146/tde-17082016-161502/pt-br.php>>. Acesso em: 28 nov. 2025.

AWS. O que é uma aplicação Web? – Explicação sobre aplicações Web — AWS. Disponível em: <https://aws.amazon.com/pt/what-is/web-application/>. Acesso em: 19 nov. 2025.

AXIOS. Introdução | Axios Docs. Disponível em: <https://axios-http.com/ptbr/docs/intro>. Acesso em: 19 nov. 2025.

CALVERT, N. Why we care about BIM. 2013. Disponível em: <https://revistatecie.crea-pr.org.br/index.php/revista/article/view/684>. Acesso em: 07 mai. 2025.

Clean Coder Blog. Disponível em: <https://blog.cleancoder.com/uncle-bob/2012/08/13/the-clean-architecture.html>. Acesso em: 22 nov. 2025.

EASTMAN, C. et al. BIM Handbook: A Guide to Building Information Modeling. 2. ed. New Jersey: John Wiley & Sons, 2011.

FERREIRA, F. C. A. et al. O futuro do projeto de arquitetura: uma revisão sistemática da literatura sobre a integração de inteligência artificial com a metodologia BIM. Ambiente Construído, Porto Alegre, v. 22, n. 4, p. 195-212, out./dez. 2022. Disponível em: <<https://seer.ufrgs.br/ambienteconstruido/article/view/118318>>. Acesso em: 28 nov. 2025.

GIL, A. C. Como Elaborar Projetos de Pesquisa. 7. ed. São Paulo: Atlas, 2021.

MANZO, A. J. Manual para elaboração de monografias: guia para apresentação de relatórios e teses. Buenos Aires: Humanitas, 1971.

MENDES DA SILVA FILHO, A. Requisitos não funcionais. Disponível em: Nuseibeh, B. & Easterbrook, S. Requirements engineering: a roadmap, in 'ICSE '00: Proceedings of the Conference on The Future of Engineering', ACM Press, New York, NY, USA, pp. 35-46, 2000.

MORAES, A. L. O Impacto da Inteligência Artificial Generativa no Processo Criativo e na Representação da Arquitetura. 2023. Trabalho de Conclusão de Curso (Graduação em Arquitetura e Urbanismo) - Universidade Federal de Santa Catarina, Florianópolis, 2023. Disponível em: <https://repositorio.ufsc.br/handle/123456789/246476>. Acesso em: 28 nov. 2025.

NESTJS. Documentation | NestJS - A progressive Node.js framework. Disponível em: <https://docs.nestjs.com>. Acesso em: 05 jan. 2025.

NIELSEN, J. Usability Engineering. Boston: Academic Press, 1993.

NPM. npm | build amazing things. Disponível em: <https://www.npmjs.com/>. Acesso em: 19 nov. 2025.

PAIVA, A. F. Tecnologia BIM: O Uso da Informação no Processo de Projeto de Arquitetura. 2016. Trabalho de Conclusão de Curso (Graduação em Arquitetura e Urbanismo) - Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2016. Disponível em: <<https://pantheon.ufrj.br/handle/11422/3599>>. Acesso em: 28 nov. 2025.

PATNI, S. Pro RESTful APIs: design, build and integrate with REST, JSON, XML and JAX-RS. Santa Clara, CA: Apress, 2017.

POSTGRESQL. Documentation. Disponível em: <https://www.postgresql.org/docs/>. Acesso em: 19 nov. 2025.

POSTMAN. A beginner's guide to APIs | Postman. Disponível em: <https://www.postman.com/what-is-an-api/>. Acesso em: 15 nov. 2025.

PRESSMAN, R. S. Engenharia de Software. 8. ed. São Paulo: McGraw-Hill, 2015.

PRESSMAN, R. S.; MAXIM, B. Engenharia de Software: uma abordagem profissional. 9. ed. Porto Alegre: AMGH, 2021.

PRISMA. Prisma documentation: concepts, guides, and reference. Disponível em: <https://www.prisma.io/docs/>. Acesso em: 19 nov. 2025.

RANE, A. V. P. Building Information Modeling (BIM) and Artificial Intelligence (AI) for Project Management: An Integrated Approach. 2023. Tese (Doutorado em Construção Civil) - University of Colorado, Boulder, 2023. Disponível em: <https://scholar.colorado.edu/concern/graduate_thesis_or_dissertations/6h441v65r>. Acesso em: 28 nov. 2025.

REDHAT. O que é API? Disponível em: <https://www.redhat.com/pt-br/topics/api/what-are-application-programming-interfaces>. Acesso em: 09 nov. 2025.

SHADCN UI. Documentation. 2025. Disponível em: <https://ui.shadcn.com/docs>. Acesso em: 28 nov. 2025.

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

SOMMERVILLE, I. Engenharia de Software. 10. ed. São Paulo: Pearson, 2019.

SOMMERVILLE, I.; SAWYER, P. Requirements engineering: a good practice guide. Chichester, England: John Wiley, 1997.

SOUZA, L.; FABRICIO, M. M. A digitalização acelerada da construção e as novas competências para o projeto: uma revisão da literatura. Ambiente Construído, Porto Alegre, v. 21, n. 4, p. 283-298, out./dez. 2021. Disponível em: <<https://seer.ufrgs.br/ambienteconstruido/article/view/109787>>. Acesso em: 28 nov. 2025.

SOUZA, L.; LYRIO, A.; AMORIM, S. Impactos do Uso do BIM em Escritórios de Arquitetura: Oportunidades no Mercado Imobiliário. In: ENCONTRO DE TECNOLOGIA DE INFORMAÇÃO E COMUNICAÇÃO NA CONSTRUÇÃO CIVIL, 4., 2009, Rio de Janeiro. Anais... Rio de Janeiro, 2009.

SOUZA, L. L. A. Diagnóstico do uso do BIM em empresas de projeto de Arquitetura. 2009. 107 f. Dissertação (Mestrado em Engenharia Civil) – Universidade Federal Fluminense, Niterói, 2009.

TAILWIND CSS. Documentation - Tailwind CSS. Disponível em: <https://tailwindcss.com/docs>. Acesso em: 19 nov. 2025.

TANSTACK. Overview | TanStack Query Docs. Disponível em: <https://tanstack.com/query/latest/docs/framework/react/overview>. Acesso em: 19 nov. 2025.

VANDERLEI, R. C. G. Orçamento de obras. 3. ed. São Paulo: Érica, 2004.

VENKATESHKONDI. Domain-Driven Design (DDD) Architecture. Disponível em: <https://medium.com/@venkateshkondi1533/domain-driven-design-ddd-architecture-49bdec77a715>. Acesso em: 22 nov. 2025.

VERCEL. Getting started | Next.js. Disponível em: <https://nextjs.org/docs>. Acesso em: 19 nov. 2025.

VOOMP. O que é Back-end e Front-end?. 2021. Disponível em: <<https://voomp.com.br/blog/o-que-e-back-end-e-front-end/>>. Acesso em: 28 nov. 2025.

WAZLAWICK, R. S. Metodologia da Pesquisa para Ciência da Computação. 2. ed. Rio de Janeiro: Campus, 2014.

ZHANG, K. et al. A systematic review of artificial intelligence in BIM. Engineering, Construction and Architectural Management, v. 29, n. 2, p. 245-267, 2022.

ZOD. Zod Documentation. 2025. Disponível em: <https://zod.dev/>. Acesso em: 28 nov. 2025.