

**INSTITUTO FEDERAL GOIANO - CAMPUS MORRINHOS
CURSO SUPERIOR DE TECNÓLOGO EM SISTEMAS PARA
INTERNET**

RENNER BORGES FERREIRA

**PLATAFORMA DE GERENCIAMENTO DE ARTEFATOS
ASSOCIADOS NA CONSTRUÇÃO DE AMBIENTES VIRTUAIS
TRIDIMENSIONAIS**

**MORRINHOS - GO
2023**

RENNER BORGES FERREIRA

**PLATAFORMA DE GERENCIAMENTO DE ARTEFATOS
ASSOCIADOS NA CONSTRUÇÃO DE AMBIENTES VIRTUAIS
TRIDIMENSIONAIS**

Monografia apresentada ao Curso Superior de Tecnólogo em sistemas para internet do Instituto Federal Goiano – Campus Morrinhos, como requisito parcial para obtenção de título de Tecnólogo em sistemas para internet.

Área de concentração: Ciência da Computação

Orientador: Prof. Dr. Alexandre Carvalho Silva

**MORRINHOS - GO
2023**

Sistema desenvolvido pelo ICMC/USP
Dados Internacionais de Catalogação na Publicação (CIP)
Sistema Integrado de Bibliotecas - Instituto Federal Goiano

FF383p Ferreira, Renner Borges
PLATAFORMA DE GERENCIAMENTO DE ARTEFATOS
ASSOCIADOS NA CONSTRUÇÃO DE AMBIENTES VIRTUAIS
TRIDIMENSIONAIS / Renner Borges Ferreira; orientador
Alexandre Carvalho Silva. -- Morrinhos, 2023.
50 p.

TCC (Graduação em TECNÓLOGO EM SISTEMAS PARA
INTERNET) -- Instituto Federal Goiano, Campus
Morrinhos, 2023.

1. Ambientes 3D. 2. API Rest. 3. Desenvolvimento
React. 4. Banco NoSQL. I. Silva, Alexandre Carvalho,
orient. II. Título.

TERMO DE CIÊNCIA E DE AUTORIZAÇÃO

PARA DISPONIBILIZAR PRODUÇÕES TÉCNICO-CIENTÍFICAS

NO REPOSITÓRIO INSTITUCIONAL DO IF GOIANO

Com base no disposto na Lei Federal nº 9.610, de 19 de fevereiro de 1998, AUTORIZO o Instituto Federal de Educação, Ciência e Tecnologia Goiano a disponibilizar gratuitamente o documento em formato digital no Repositório Institucional do IF Goiano (RIIF Goiano), sem ressarcimento de direitos autorais, conforme permissão assinada abaixo, para fins de leitura, download e impressão, a título de divulgação da produção técnico-científica no IF Goiano.

IDENTIFICAÇÃO DA PRODUÇÃO TÉCNICO-CIENTÍFICA

Tese (doutorado)

Dissertação (mestrado)

Monografia (especialização)

TCC (graduação)

Artigo científico

Capítulo de livro

Livro

Trabalho apresentado em evento

Produto técnico e educacional - Tipo:

Nome completo do autor:

Matrícula:

Título do trabalho:

RESTRIÇÕES DE ACESSO AO DOCUMENTO

Documento confidencial: Não Sim, justifique:

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

O documento está sujeito a registro de patente? Sim Não


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

DECLARAÇÃO DE DISTRIBUIÇÃO NÃO-EXCLUSIVA

O(a) referido(a) autor(a) declara:

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

/ /

Documento assinado digitalmente
 RENNER BORGES FERREIRA
Data: 05/04/2023 09:11:01-0300
Verifique em <https://validar.iti.gov.br>


Local

Data

Assinatura do autor e/ou detentor dos direitos autorais

Ciente e de acordo:

Assinatura do(a) orientador(a)

Documento assinado digitalmente
 ALEXANDRE CARVALHO SILVA
Data: 05/04/2023 09:13:52-0300
Verifique em <https://validar.iti.gov.br>



Serviço Público Federal
Ministério da Educação
Instituto Federal de Educação, Ciência e Tecnologia Goiano
Campus Morrinhos
Curso Superior de Tecnologia em Sistemas para Internet

ATA DE APRESENTAÇÃO DO TRABALHO DE CURSO

Aos **30** dias do mês de **março** de 2023, às **19** horas e **30** minutos, foi realizada, nas dependências do Instituto Federal Goiano – campus Morrinhos, a apresentação pública do Trabalho de Curso do discente **Renner Borges Ferreira**, intitulado como **Plataforma de Gerenciamento de Artefatos Associados na Construção de Ambientes Virtuais Tridimensionais** como requisito necessário para a conclusão do curso. A Banca Examinadora, constituída pelos professores:

Alexandre Carvalho Silva – Orientador(a)

Reidner Santos Cavalcante – Convidado 1

Marcel da Silva Melo – Convidado 2

emitiram o seguinte resultado:

X	Aprovado
	Aprovado com ressalva (A Banca Examinadora deve definir as exigências a serem cumpridas pelo aluno na revisão, ficando o orientador responsável pela verificação do cumprimento das mesmas.)
	Observações:
	Reprovado com o seguinte parecer:

Por ser verdade firmamos a presente:

Documento assinado digitalmente
gov.br ALEXANDRE CARVALHO SILVA
Data: 05/04/2023 09:01:57-0300
Verifique em <https://validar.iti.gov.br>

Profº Dr. Alexandre Carvalho Silva
(Presidente da Banca)

Documento assinado digitalmente
gov.br REIDNER SANTOS CAVALCANTE
Data: 05/04/2023 17:47:40-0300
Verifique em <https://validar.iti.gov.br>

Documento assinado digitalmente
gov.br MARCEL DA SILVA MELO
Data: 05/04/2023 16:51:12-0300
Verifique em <https://validar.iti.gov.br>

Profº Me. Reidner Santos Cavalcante

Profº Me. Marcel da Silva Melo

Morrinhos **31** de março de 2023

AGRADECIMENTOS

Gostaria de aproveitar este momento para expressar a minha sincera gratidão a todos os que me apoiaram durante esta jornada acadêmica e profissional. Em primeiro lugar, gostaria de agradecer à minha família, que sempre esteve ao meu lado, me incentivando e me dando força nos momentos mais difíceis.

Também gostaria de agradecer aos meus amigos, tanto os antigos quanto os novos, que me acompanharam em todos os momentos, me alegraram nos de descontração e me encorajaram nos desafios. Agradeço em especial aos meus amigos de trabalho, que me ensinaram muito e me deram a oportunidade de crescer profissionalmente.

Não posso deixar de agradecer aos meus professores, que me guiaram durante toda a minha jornada acadêmica, compartilhando seu conhecimento e experiência comigo. Eles me inspiraram a aprender mais e me ajudaram a superar todos os obstáculos.

Também quero expressar meu mais sincero agradecimento à minha esposa. Ela sempre esteve presente em cada etapa desta jornada, me apoiando, encorajando e motivando a superar todos os desafios. Sem a sua dedicação e apoio incondicional, eu não teria conseguido alcançar este objetivo.

Por fim, agradeço a todos que confiaram em mim ao longo desta jornada, incluindo colegas de classe, colegas de trabalho, mentores e pessoas que acreditaram em meu potencial. Cada um de vocês contribuíram para o meu sucesso e sou muito grato por isso.

LISTA DE ILUSTRAÇÕES

FIGURAS

Figura 1 - Diagrama de arquitetura	5
Figura 2 - Casos de uso	6
Figura 3 - Diagrama de classe	8
Figura 4 - Fluxograma de autenticação	14
Figura 5 - Fluxograma de criação de usuário	16
Figura 6 - Fluxograma de criação de login	18
Figura 7 - Fluxograma de criação de objeto e modelos	19
Figura 8 - Página de login da plataforma	21
Figura 9 - Página principal da plataforma	22
Figura 10 - Item da lista de objetos	24
Figura 11 - Detalhes de um objeto na lista	24
Figura 12 - Formulário para criação de um objeto e modelo	25
Figura 13 - Validação do formulário de criação de objetos e modelos	26
Figura 14 - Tela da documentação das rotas	27
Figura 15 - Tela da documentação das rotas com os blocos abertos	27
Figura 16 - Tela da documentação de uma rota aberta	28

TABELAS

Tabela 1 - Descrição UC01	7
Tabela 2 - Descrição UC02	7
Tabela 3 - Descrição UC03	7
Tabela 4 - Descrição UC04	8
Tabela 5 - Teste UC01 e UC02	31
Tabela 6 - Resultado do teste UC01 e UC02	31
Tabela 7 - Teste UC03	32
Tabela 8 - Resultado do teste UC03	33

SIGLAS

Caso de Uso (UC)

Application Programming Interface (API)

JavaScript Object Notation (JSON)

NPM (Node Package Manager)

NoSQL (Not only SQL)

SQL (Structured Query Language)

JWT (JSON Web Token)

REST (Representational State Transfer)

SSR (Server Side Rendering)

NPM (Node Package Manager)

SUMÁRIO

LISTA DE ILUSTRAÇÕES	5
FIGURAS	5
TABELAS	5
SIGLAS	7
SUMÁRIO	8
RESUMO	10
ABSTRACT	11
1. INTRODUÇÃO	1
1.1 PROBLEMÁTICA	1
1.2 ÁREA COM POTENCIAL DE USO NA PROBLEMÁTICA	2
1.3 OBJETIVO GERAL	2
1.4 OBJETIVOS ESPECÍFICOS	2
1.5 JUSTIFICATIVA	2
1.6 ORGANIZAÇÃO DO TRABALHO	3
2. ARQUITETURA DO SISTEMA	4
2.1 CASOS DE USO	6
2.1.1 Descrição dos Atores	6
2.1.2 Descrição dos casos de uso	7
2.2 DIAGRAMA DE CLASSE	8
2.3 TECNOLOGIAS UTILIZADAS	9
2.3.1 Javascript	9
2.3.2 React	9
2.3.3 Next.js	10
2.3.4 NextUI	10
2.3.5 NodeJS	11
2.3.6 Express.js	11
2.3.7 MongoDB	12
2.3.8 YUP	12
2.3.9 Swagger	13
2.4 FLUXOGRAMAS	13
2.4.1 Fluxograma de autenticação	14
2.4.2 Fluxograma de criação de usuário	16
2.4.3 Fluxograma de login	18
2.4.4 Fluxograma de criação de objetos e modelos	19
2.5 OBSERVAÇÕES FINAIS	20
3. FUNCIONAMENTO DO SISTEMA	21
3.1 LOGIN	21
3.2 PÁGINA INICIAL	22

3.2.1 Desconexão	22
3.2.2 Mudar tema	23
3.2.3 Listagem de objetos	23
3.2.4 Pesquisa de objetos	23
3.2.5 Botão de criação de objeto	23
3.2.6 AÇÕES DE CADA OBJETO NA LISTA	24
3.2.7 Objeto expandido	24
3.3 CRIAÇÃO E EDIÇÃO	25
3.4 DOCUMENTAÇÃO	27
3.5 CONSIDERAÇÕES FINAIS	29
4 TESTES DE FUNCIONAMENTO	30
4.1 CONSIDERAÇÕES INICIAIS	30
4.2 TESTE DE CAIXA PRETA	30
4.3 VALIDAÇÃO DOS CASOS DE TESTE	30
4.3.1 Gerência de objetos e modelos	31
4.3.2 Autenticação	32
4.4 CONSIDERAÇÕES FINAIS	34
5. CONCLUSÃO	35
5.1 TRABALHOS FUTUROS	36
REFERENCIAS	37

RESUMO

A demanda por ambientes virtuais tridimensionais está crescendo, requerendo vários artefatos (scripts, modelos 3D, texturas, arquivos multimídia) para criar um ambiente tridimensional imersivo e interativo. Para garantir uma construção sem falhas e evitar retrabalhos, é essencial ter recursos facilitadores que permitam controlar, versionar e gerenciar esses artefatos. Nesse sentido, a criação de sistemas que ofereçam suporte às tecnologias web para gerenciar esses artefatos é altamente atraente para essa área. Com base nessa demanda e potencialidade, foi desenvolvido um sistema que detalha os aspectos tecnológicos, a implementação e a documentação de uso. Além disso, foram realizados testes de funcionalidade para validar os recursos da aplicação.

Palavras-chave: *Ambientes 3D , API Rest, Desenvolvimento React, Banco NoSQL.*

ABSTRACT

The demand for three-dimensional virtual environments continues to grow, requiring a variety of artifacts (scripts, 3D models, textures, media files) to create an immersive and interactive 3D environment. To ensure flawless construction and avoid alterations, it is essential to have facilitation features that allow you to control, version and manage these artifacts. In this sense, the creation of systems that support web technologies to manage these artifacts is highly attractive for this area. On the basis of this request and potential, a system has been developed which details the technological aspects, implementation and documentation of use. Additionally, functional tests were conducted to validate the functionality of the application.

Keywords: 3D Environments, Rest API, React Development, NoSQL Database.

1. INTRODUÇÃO

O crescimento da demanda por aplicações virtuais tridimensionais acontece em variados setores nos últimos anos, desde o entretenimento até a indústria. Para a construção desses cenários são necessários artefatos como modelos 3D, que compõem um cenário virtual.

Nesta vertente, os ambientes virtuais interativos têm provado que possui real potencial, ganhado destaque nas indústrias com diferentes funcionalidades, como treinamento, suporte à decisão, design e revisão, ajudando com a manutenção e simulação de operações industriais (MICHALOS, 2018).

No entanto, a gestão desses artefatos muitas vezes são realizadas manualmente, de forma não padronizada e não integrada, o que acaba levando em falhas no processo construtivo, como modelos incorretos, posicionamentos inadequados e falta de versionamento.

Essas falhas podem impactar no gerenciamento do tempo, aspectos visuais e na qualidade da experiência imersiva desses cenários virtuais. Nesse sentido, a busca por soluções eficientes e integradas para a gestão dos artefatos se torna cada vez mais urgente.

1.1 PROBLEMÁTICA

Com base neste cenário de necessidade surge a seguinte problemática: Como minimizar os impactos causados pela falta de recursos gerenciadores de artefatos para a construção de ambientes tridimensionais?

1.2 ÁREA COM POTENCIAL DE USO NA PROBLEMÁTICA

O desenvolvimento de sistemas computacionais baseados na internet é uma área que está em bastante evolução e oferece cada vez mais soluções eficientes para as necessidades de usuários em geral e empresas.

Nesse contexto, surge a arquitetura baseada em microsserviços que tem se mostrado uma opção promissora para integrar recursos e automatizar tarefas, gerir dados e escalar recursos.

1.3 OBJETIVO GERAL

Este projeto visa promover uma solução computacional com capacidade e facilidade de integrar recursos com potencial de auxiliar a gestão de ativos associados à construção de ambientes virtuais 3D, minimizando os impactos causados pela problemática apresentada.

1.4 OBJETIVOS ESPECÍFICOS

- Disponibilizar recursos de autenticação para controle de acesso;
- Possibilitar acesso às informações (documentação) relacionadas a aspectos de integração da solução desenvolvida com ferramentas externas;
- Fornecer interface gráfica visual (*dashboard*) para visualização clara e concisa de dados.

1.5 JUSTIFICATIVA

A falta de recursos gerenciadores para a construção de ambientes tridimensionais pode levar a diversos problemas como já supracitado.

Nesse cenário surge a possibilidade de uma solução eficiente e integrada para otimizar o processo de construção, garantindo a qualidade, posicionamento e demais características desses artefatos. Além disso, a solução facilita a colaboração e a comunicação entre os membros da equipe, aumentando a eficiência e a produtividade do projeto na totalidade.

Este projeto oferece uma solução completa para o gerenciamento de artefatos em ambientes virtuais, incluindo uma interface visual com recursos modernos, como microsserviços, interface web reativa e documentação de rotas. Além disso, a documentação de rotas facilita a integração com as ferramentas de desenvolvimento de ambientes tridimensionais (normalmente *game engines*), tornando a solução escalável e fácil de integrar.

1.6 ORGANIZAÇÃO DO TRABALHO

Esta monografia foi estruturada a partir da composição de cinco capítulos, descritos brevemente a seguir:

O primeiro capítulo introduz a problemática e a possível solução a partir dos objetivos descritos.

O segundo capítulo apresenta a arquitetura da solução, composto pelos diagramas e casos de uso, permitindo entender as necessidades do projeto, relatando a implementação do sistema, mostrando as tecnologias utilizadas e diagramas de fluxos.

O terceiro capítulo exhibe e descreve a interface do sistema e seus recursos.

O quarto capítulo apresenta os testes de funcionamento, realizados utilizando testes de caixa preta.

Por fim é apresentado as conclusões e trabalhos futuros desta pesquisa.

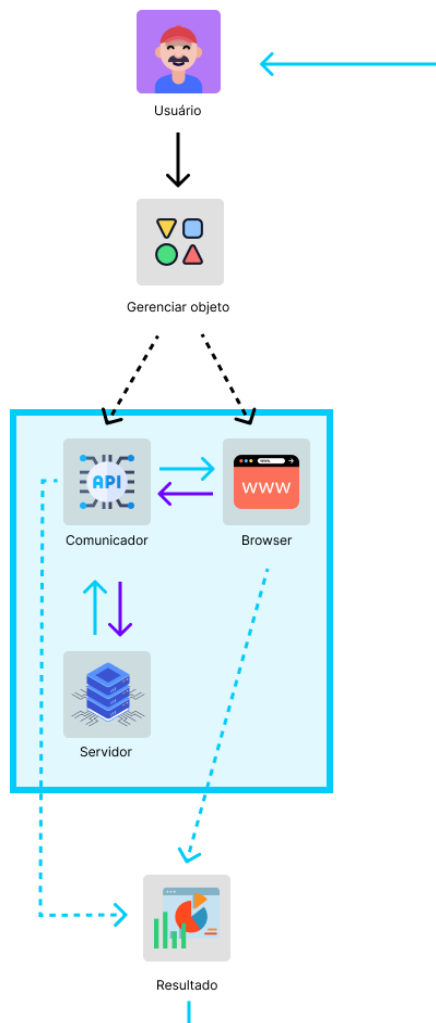
2. ARQUITETURA DO SISTEMA

A arquitetura proposta para o desenvolvimento de sistema é baseada em duas possibilidades de utilização, sendo elas o acesso via navegador e o acesso pela API. Esta arquitetura permite que o sistema seja acessado por uma grande variedade de serviços ou plataformas, expandindo assim seu alcance e funcionalidade.

O acesso pelo navegador permite que os usuários acessem o sistema pelo mesmo, realizando ações como criação, busca e edição de objetos e modelos. Sendo desenvolvido para uma proposta de uma interface web amigável e de fácil utilização, permitindo que até os usuários com menos conhecimento tecnológico possam utilizá-lo sem dificuldades.

Por outra vertente, é possível o acesso pela API permitindo que outras plataformas integrem-se ao sistema, acessando sua base de recursos desde que estejam autorizadas a isso. Para tanto foi utilizado o padrão de projeto REST, permitindo uma integração fácil e segura com outras plataformas. Além disso, a API possui documentação viva, facilitando a integração de terceiros.

A arquitetura do sistema foi planejada para ser escalável. Além disso, durante o desenvolvimento foi considerado questões de segurança, utilizando medidas como autenticação de usuários e criptografia de dados sensíveis. A arquitetura do sistema pode ser analisada resumidamente através da Figura 1.

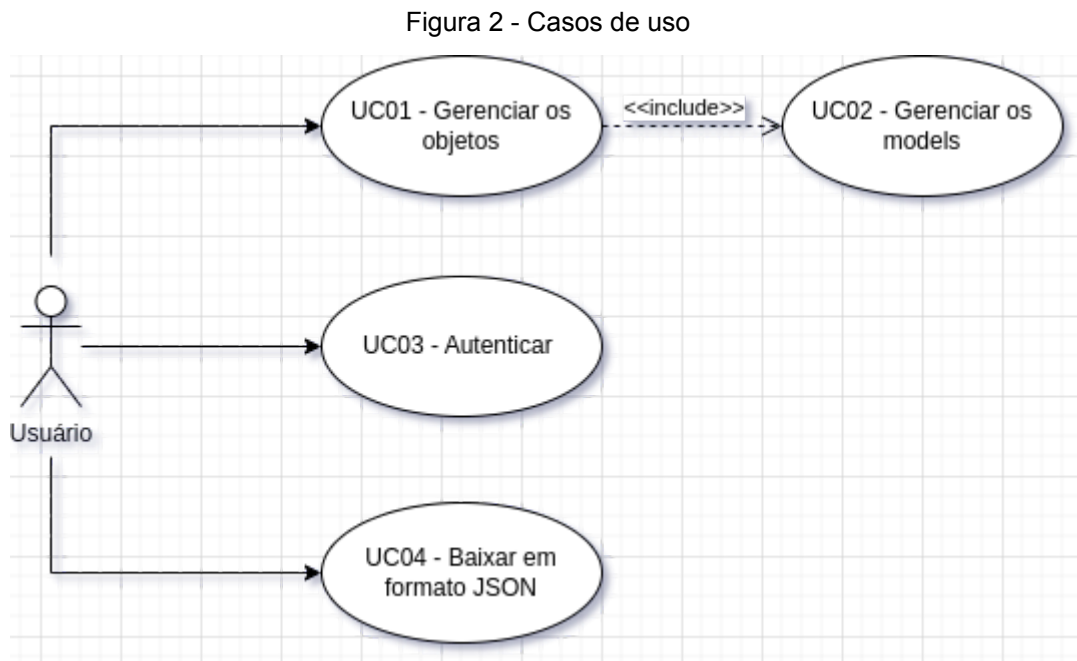


Fonte: Autor.

A Figura 1 demonstra a arquitetura do sistema sintetizadamente, onde se tem algumas etapas caracterizadas por cores diferentes, as setas contínuas correspondem aos fluxos obrigatórios enquanto as tracejadas correspondem aos fluxos opcionais.

Sendo a primeira etapa, representadas pelas setas pretas simboliza o usuário utilizando a plataforma para gerenciar um objeto, na segunda etapa temos as setas roxas representando o caminho dos dados gerenciados até que a ação seja concretizada e na terceira etapa temos as setas azuis que representa o fluxo do resultado até retornar ao usuário.

2.1 CASOS DE USO



Fonte: Autor.

Ao analisar o diagrama de caso de uso é possível observar como o usuário, o ator principal pode interagir com a aplicação, o mesmo pode se autenticar, baixar o objeto e o *modelo* no formato de JSON permitindo assim a integração com outras plataformas, gerenciar os objetos e consequentemente gerenciar os modelos.

2.1.1 Descrição dos Atores

O caso de uso possui apenas um ator, o usuário. O mesmo tem total acesso à aplicação, após sua autenticação, onde poderá via uma API ou navegador gerenciar os objetos e modelos.

2.1.2 Descrição dos casos de uso

Tabela 1 - Descrição UC01

Caso de uso	<i>UC01 - Gerenciar os objetos</i>
Ator Principal	<i>Usuário</i>
Descrição	<i>Possibilitar ao usuário o gerenciamento dos objetos, controlando informações importantes e relevantes sobre os mesmos</i>
Pré-Condição	<i>UC03 - Autenticar</i>
Fluxo Normal	<i>Inexistente</i>
Fluxos Excepcionais	<ol style="list-style-type: none"> 1. <i>Criar um objeto</i> 2. <i>Editar um objeto</i> 3. <i>Buscar um objeto</i> 4. <i>Apagar um objeto</i>
Pós-Condição	<i>UC02 - Gerenciar os modelos</i>

Fonte: Autor.

Tabela 2 - Descrição UC02

Caso de uso	<i>UC02 - Gerenciar os modelos</i>
Ator Principal	<i>Usuário</i>
Descrição	<i>Possibilitar ao usuário o gerenciamento dos modelos, controlando informações importantes e relevantes sobre os mesmos</i>
Pré-Condição	<i>UC01 - Gerenciar os objetos</i>
Fluxo Normal	<i>Inexistente</i>
Fluxos Excepcionais	<ol style="list-style-type: none"> 1. <i>Criar um objeto</i> 2. <i>Editar um objeto</i> 3. <i>Buscar um objeto</i> 4. <i>Apagar um objeto</i>
Pós-Condição	<i>Inexistente</i>

Fonte: Autor.

Tabela 3 - Descrição UC03

Caso de uso	<i>UC03 - Autenticar</i>
Ator Principal	<i>Usuário</i>
Descrição	<i>Possibilitar ao usuário autenticar-se para ter acesso os demais recursos</i>
Pré-Condição	<i>Inexistente</i>

Fluxo Normal	<i>1. Informar as credenciais</i>
Fluxos Excepcionais	<i>Inexistente</i>
Pós-Condição	<i>UC01 - Gerenciar os objetos, UC04 - Baixar em formato JSON</i>

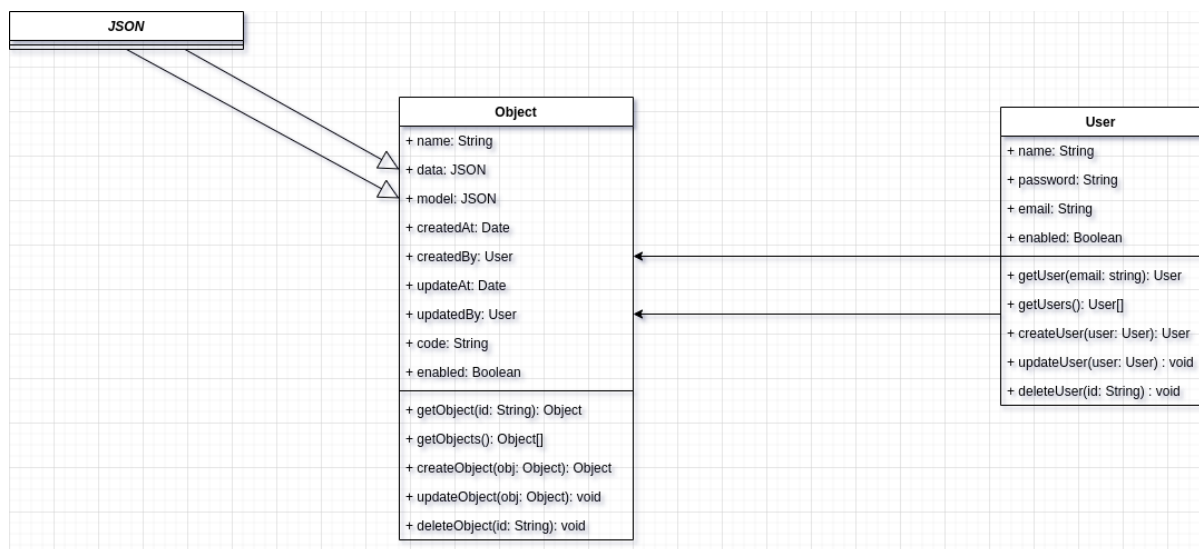
Fonte: Autor.

Tabela 4 - Descrição UC04

Caso de uso	<i>UC04 - Baixar em formato JSON</i>
Ator Principal	<i>Usuário</i>
Descrição	<i>Disponibilizar ao usuário a opção de ter seu object ou model em um padrão JSON onde o mesmo poderá ser manipulado em outras plataformas</i>
Pré-Condição	<i>UC03 - Autenticar</i>
Fluxo Normal	<i>1. Selecionar o objeto ou model que deseja exportar</i>
Fluxos Excepcionais	<i>Inexistente</i>
Pós-Condição	<i>Inexistente</i>

2.2 DIAGRAMA DE CLASSE

Figura 3 - Diagrama de classe



Fonte: Autor.

O diagrama de classes é uma forma de apresentar estaticamente a estrutura de um sistema, apresentando-se classes, atributos, métodos e até mesmo as relações entre objetos.

Sendo assim através do diagrama de classe é possível ter uma noção melhor da arquitetura dos dados onde se tem uma classe principal *Object*, a mesma se associa com outras duas.

Obtendo-se a classe abstrata JSON com o propósito de representar o acrônimo de seu nome, sendo uma classe fundamental para o fluxo e a outra é a classe *User* que serve para termos uma autenticação e um mapeamento dos recursos, como, por exemplo, quem criou ou atualizou um objeto.

2.3 TECNOLOGIAS UTILIZADAS

2.3.1 Javascript

Tendo-se como objetivo do software comunicar com aplicação web e diretamente com outras plataformas via API, foi escolhido o Javascript por fatores como sua alta escalabilidade, desempenho nesses cenários e pela sua comunidade bastante ativa.

Destacando-se pela versatilidade de construir aplicações *front-end* e *back-end* utilizando a mesma linguagem, permitindo utilizar e adaptar o projeto para vários cenários, facilitando a manutenção e o desenvolvimento do projeto (Javascript, 2023).

2.3.2 React

A utilização do React, uma biblioteca em javascript para o *front-end*, se dá pela necessidade de criação de um dashboard para gerenciar os objetos e seus respectivos modelos.

O mesmo foi escolhido pelo seu alto desempenho proporcionado graças à Virtual Dom permitindo que apenas o que foi modificado seja re-renderizado.

Tendo como outro ponto positivo sua flexibilidade e por ser escalável considerando que cada componente é autônomo, possibilitando utilizar o mesmo componente em mais de um lugar, comportamento esse que permite corrigir um erro de um componente e corrigir em todos os lugares que estão utilizando (React, 2023)

2.3.3 Next.js

O Next é um dos *frameworks* mais populares do mercado atualmente, sendo um framework para React permitindo uma pré-configuração, com um aglomerado de funcionalidades com objetivo de agilizar o desenvolvimento de aplicações web.

Sua utilização no projeto é justificada pela agilidade no desenvolvimento proveniente da divisão do código, pois para tornar o carregamento mais rápido o código é quebrado de forma automática.

Outro fator é o SSR que simula um *back-end* dentro do *front-end* e renderiza uma página HTML, permitindo que um usuário com o javascript desabilitado no navegador visualize a página, agilizando o carregamento da página. (Next.js, 2023)

O SSR conta com a vantagem que a página consegue ser identificada pelo motor de pesquisa do Google.

2.3.4 NextUI

Considerando agilizar o desenvolvimento e a eficiência do *dashboard* foi utilizado a biblioteca de componentes NextUI que permite a criação de aplicações modernas e rápidas, ocasionando em menos codificação e mais tempo de

desenvolvimento, facilidade de utilizar e seu principal a compatibilidade com *cross-browser*. (NextUI, 2023)

A escolha dessa biblioteca se dá pela grande compatibilidade com o Next.js, seus componentes bem completos, pela sua documentação bem detalhada e pela possibilidade e facilidade em personalizar os componentes.

2.3.5 NodeJS

O *NodeJS*, utiliza o Javascript para escrever programas server-side, utilizando o interpretador V8 do Google, sendo multiplataforma rodando em Linux ou Windows e possuindo uma comunidade grande e ativa (Node.js,2023).

A utilização deste na aplicação se dá por fatores como sua flexibilidade disponibilizada pelo NPM contendo uma abundância de módulos, pela sua leveza onde não se é exigido muitos recursos de computação e por consequência utilizando pouca memória RAM.

Por ser *single-thread*, traz uma alta escalabilidade e baixo tempo de resposta no cenário dessa aplicação, fatores esses explicados, pois sendo *single-thread* é possível executar milhões de conexões simultâneas.

2.3.6 Express.js

O Express.js é um *framework* que facilita o desenvolvimento das aplicações *back-end*, sendo rápido e leve, podendo até dizer que é um micro-framework por contar apenas com o essencial do seu propósito que é possibilitar a construção de API's Rest. (Express, 2023)

Sua utilização é justificada pela sua facilidade em criar rotas, permitindo gerenciar diferentes tipos de requisições HTTP e tratar suas exceções.

2.3.7 MongoDB

O MongoDB é um banco de dados NoSQL, sendo orientado a documentos, sendo uma das melhores escolhas para armazenar quantidades grandes de dados. (MongoDB, 2023)

O fator da possibilidade de armazenar dados aninhados justifica sua utilização, facilitando assim a estrutura dos dados e podendo até tornar o trabalho de busca mais eficiente quando comparado aos SQL.

2.3.8 YUP

Validar os dados entrados no corpo (*body*) de uma API é crucial para garantir a qualidade e a segurança da API. Algumas das principais razões para validar os dados incluem:

1. Prevenção de erros: Validação de dados garante que os dados recebidos sejam válidos e seguem o esperado pelo sistema, evitando erro e problemas no processamento dessas informações; (YUP, 2023)
2. Garantia da integridade dos dados: Validar os dados garante que as informações recebidas sejam precisas e corretas, o que é importante para garantir a integridade dos dados armazenados pelo sistema;
3. Melhoria da experiência do usuário: Quando os dados são validados, os usuários recebem mensagens claras e precisas sobre erros, tornando a experiência de uso mais suave e eficiente.

Pensando nisso, a biblioteca YUP está sendo utilizada no *back-end*, sendo uma biblioteca de validação de dados, onde todas as rotas da aplicação passam mediante um *middleware* para validar se os mesmos atendem seu propósito, ou seja, a regra de negócio.

2.3.9 Swagger

Sendo uma ferramenta muito útil para documentar e testar APIs. Algumas das principais razões para ter utilizado o Swagger na API incluem:

1. Documentação clara e concisa: O Swagger permite criar uma documentação detalhada e fácil de entender para a API, incluindo exemplos de requisições e respostas, descrições de parâmetros e esquemas de dados;
2. Facilidade de teste: O Swagger oferece uma interface interativa para testar a API, sem a necessidade de escrever código ou usar outras ferramentas de teste; (Swagger, 2023)
3. Integração com ferramentas de desenvolvimento: O Swagger pode ser integrado a ferramentas de desenvolvimento, como o Postman, para tornar o processo de desenvolvimento e teste da API mais eficiente;
4. Melhoria da colaboração: O Swagger ajuda a garantir que todos os membros da equipe tenham uma compreensão clara da API, incluindo suas funcionalidades e limitações.

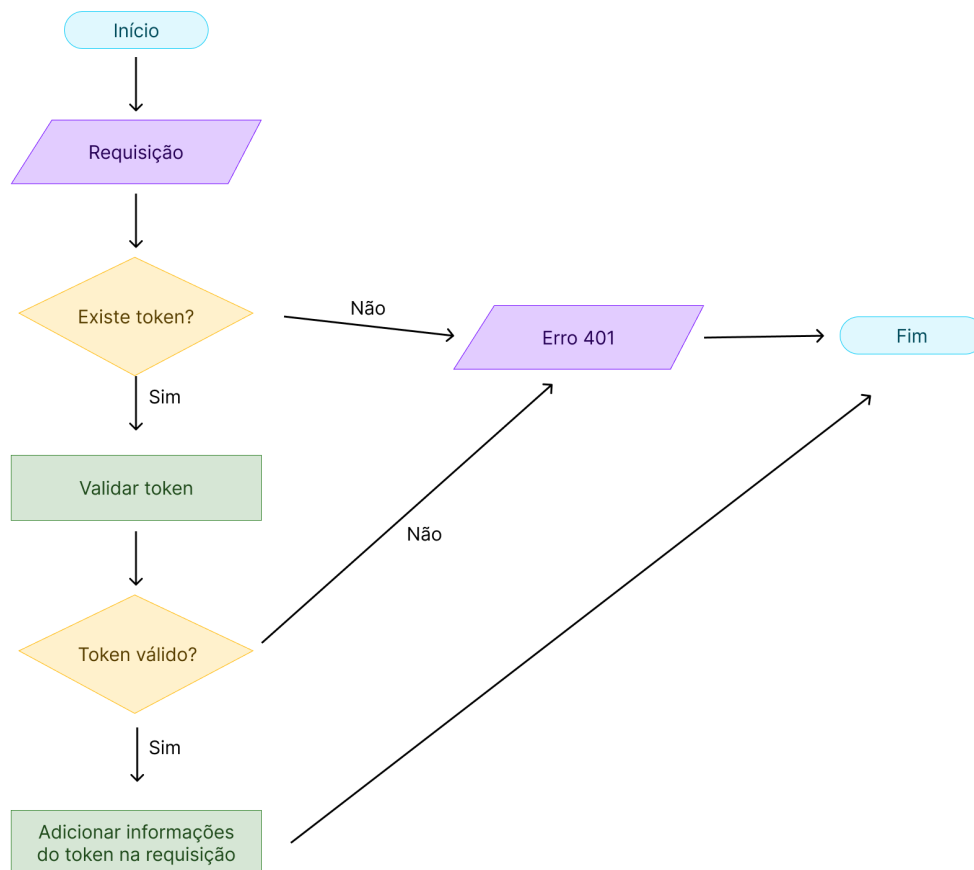
2.4 FLUXOGRAMAS

Nesta seção será apresentado os fluxogramas de algoritmos representando as principais funcionalidades da plataforma proposta.

Esses fluxogramas são recursos de modelagem visual que permite representar a sequência de processos ou ações que possuem um algoritmo, ajudando a comunicar visualmente como o mesmo é executado e onde existem as decisões, com suas possibilidades de execução.

2.4.1 Fluxograma de autenticação

Figura 4 - Fluxograma de autenticação



Fonte: Autor.

A autenticação na plataforma é essencial para seu pleno funcionamento, o algoritmo que executa essa funcionalidade pode-se muitas vezes ser confuso ou de difícil compreensão, para um melhor entendimento a Figura 4 ilustra esta ação.

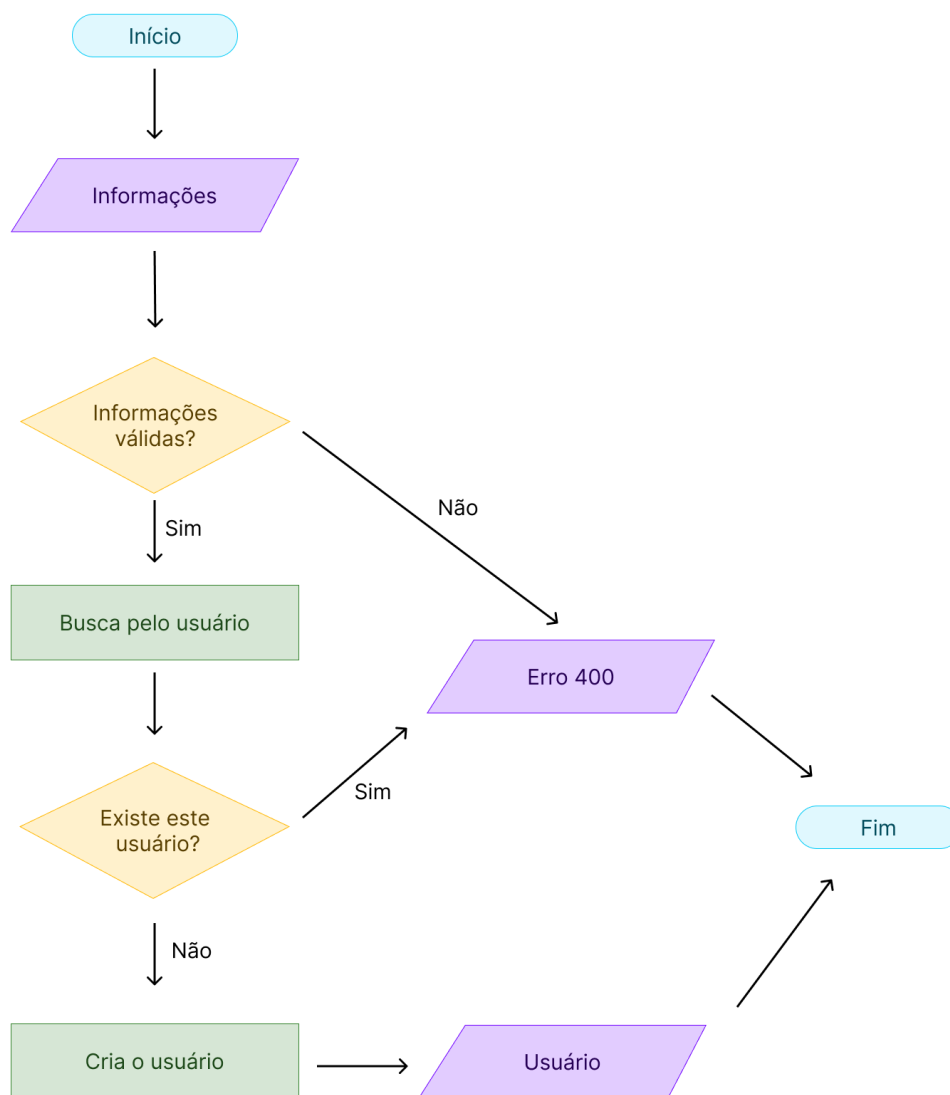
É possível dividir o algoritmo em etapas para facilitar ainda mais seu entendimento, sendo elas:

1. O fluxo ao ser iniciado receberá a requisição que possui as informações necessárias para a autenticação.
2. Após a aquisição dos dados é verificada a existência de um token dentro desses dados.
 - a. Caso tenha token, o fluxo seguirá para a próxima etapa.

- b. Senão é retornado um erro HTTP com o estado 401 de não autorizado e assim chegando ao fim do fluxo.
3. Após saber que o token foi informado, o mesmo será validado com as informações recebidas e a chave secreta presente na configuração do *back-end*.
 - a. Caso o token seja válido, o fluxo seguirá para a próxima etapa.
 - b. Senão é retornado um erro HTTP com o estado 401 de não autorizado e assim chegando ao fim do fluxo.
4. Com o token validado, as informações contidas nele são adicionadas na requisição, onde estará disponível para os próximos fluxos utilizarem e assim o fluxo é finalizado.

2.4.2 Fluxograma de criação de usuário

Figura 5 - Fluxograma de criação de usuário



Fonte: Autor.

A criação de um usuário na plataforma é uma etapa importante para obter o controle de acesso dentro da mesma.

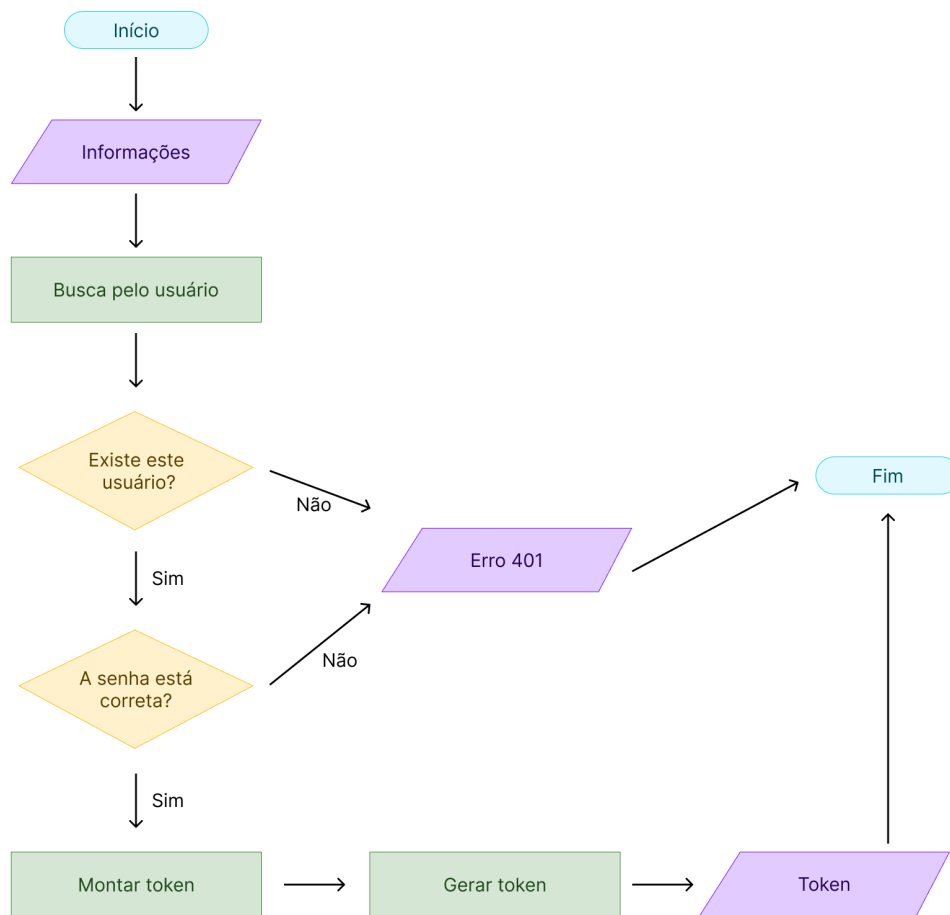
Para entender as sequências de ações dessa operação a Figura 5 representa este processo.

1. O fluxo ao ser iniciado receberá as informações pertinentes à operação.

2. Através das informações passadas é possível realizar uma verificação do formato destas, como, por exemplo, se foi passado o usuário como um texto e a senha no formato definido pela regra de negócio.
 - a. Caso sejam informações com formatos válidos, o fluxo seguirá.
 - b. Senão é enviado um erro com estado 400 de solicitação incorreta e encerrando o fluxo.
3. Agora que sabemos que as informações estão devidamente formatadas corretamente, buscaremos o usuário com as mesmas.
4. A etapa anterior poderá retornar duas situações:
 - a. Caso o usuário exista será retornado um erro com estado 400 de solicitação incorreta finalizando o fluxo
 - b. Senão o fluxo seguirá normalmente.
5. Agora que sabemos que esse usuário não está registrado na base de dados, criaremos o mesmo.
6. As informações desse novo usuário serão retornadas finalizando o fluxo.

2.4.3 Fluxograma de login

Figura 6 - Fluxograma de criação de login



Fonte: Autor.

O login no sistema é o início de toda usabilidade, sendo uma forma eficaz de proteger as informações contidas na plataforma e controlar o acesso de quem consegue acessar as mesmas.

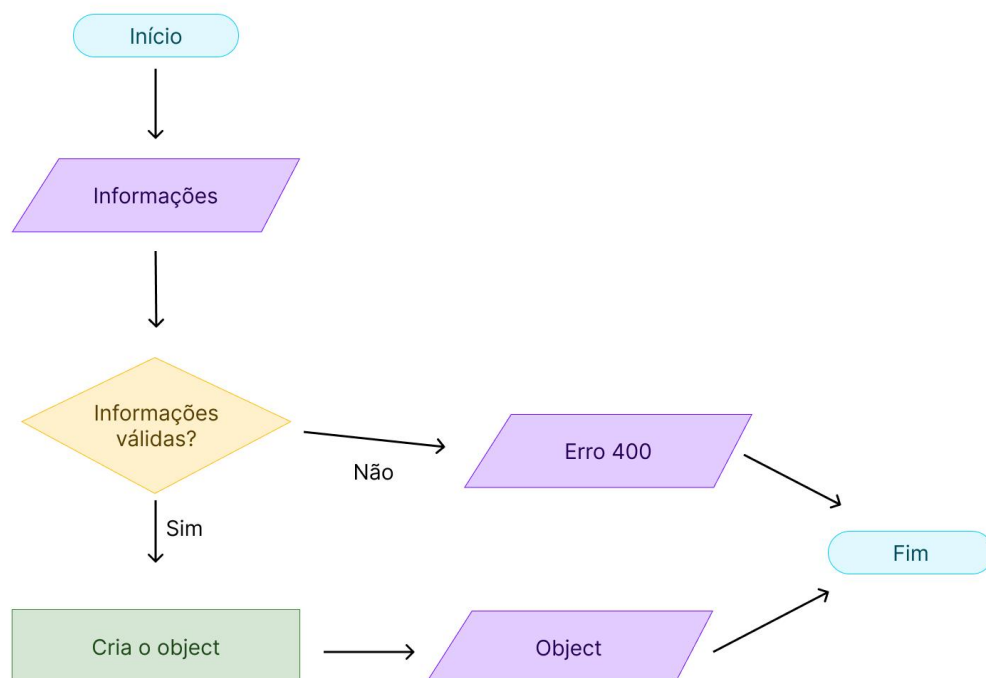
A Figura 6 mostra as sequências de ações para determinada ação. De tal forma as seguintes etapas são:

1. O fluxo ao ser iniciado receberá as informações pertinentes.
2. Com essas informações podemos buscar o usuário na base de dados, atentando-se que a senha não é passada na consulta, por considerar que a mesma é criptografada.
3. Após a busca do usuário podemos ter dois cenários sendo eles:
 - a. Caso exista o usuário, o fluxo seguirá.

- b. Senão ele irá retornar um erro 401 de não autorizado e finalizará o fluxo.
4. Sabendo que o usuário existe, então é necessário ver se a senha informada é a correta, onde:
 - a. Caso a senha seja correta, o mesmo seguirá o fluxo.
 - b. Senão ele irá retornar um erro 401 de não autorizado e finalizará o fluxo.
5. Agora que o usuário está devidamente autorizado será montado o corpo do token que contém as informações importantes para o *front-end*.
6. Com o corpo do token iremos gerá-lo no formato de JWT.
7. O novo token será retornado e assim finalizando o fluxo.

2.4.4 Fluxograma de criação de objetos e modelos

Figura 7 - Fluxograma de criação de objeto e modelos



Fonte: Autor.

Os objetos e modelos são a base de toda a plataforma, sendo a entidade mais importante e onde estarão as informações sobre os modelos tridimensionais que são os artefatos básicos para a construção de ambientes tridimensionais interativos.

Para entender as sequências de ações dessa operação será analisado a Figura 7, onde:

1. O fluxo ao ser iniciado receberá as informações pertinentes.
2. Através dessas informações é possível validar o formato destas, validação essa importante por se tratar de dados que ao serem passados errado perderão sua utilidade na plataforma que for consumir, ponderando padronizar os dados são armazenados em JSON.
 - a. Caso sejam informações com formatos válidos, o fluxo seguirá.
 - b. Senão é enviado um erro com estado 400 de solicitação incorreta e encerrando o fluxo.
3. Após essa validação o objeto é criado e conseqüentemente seu modelo também por ser um subdocumento do mesmo.
4. O objeto criado é retornado e assim encerrando o fluxo.

2.5 OBSERVAÇÕES FINAIS

O propósito desta sessão foi expor a estrutura da plataforma através da exibição de um diagrama de arquitetura, casos de uso e diagrama de classes, bem como apresentar as tecnologias empregadas e os fluxogramas para pontos específicos. É importante ressaltar que outros fluxos não mencionados nesta seção foram omitidos por se tratarem de algoritmos simples ou por serem basicamente semelhantes aos já mencionados.

3. FUNCIONAMENTO DO SISTEMA

Neste capítulo, será explorado os detalhes de como a plataforma desenvolvida neste trabalho opera para realizar suas ações descritas nos casos de uso, simulando um usuário que usará a plataforma por um navegador de internet.

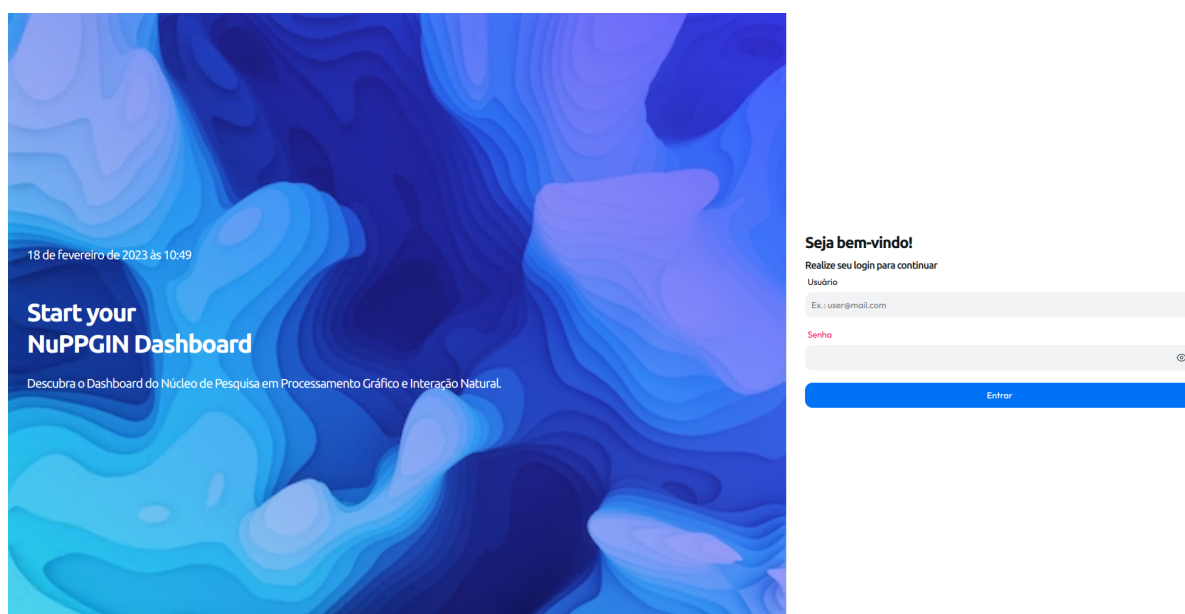
3.1 LOGIN

Na tela de login, existem *inputs* (campos para entrada de textos), sendo eles:

- Usuário: Neste campo espera-se o e-mail da conta cadastrada
- Senha: Enquanto nesse campo espera-se a senha referente a informação colocada no campo acima, onde é possível visualizar ou esconder a senha clicando no ícone de olho.

Após o preenchimento destes é possível clicar no botão escrito “Entrar”, que fará uma requisição e retornará uma mensagem de erro caso exista ou redirecionar o usuário, representado na Figura 8 a seguir:

Figura 8 - Página de login da plataforma



Fonte: Autor.

3.2 PÁGINA INICIAL

Na página principal existem muitas funcionalidades tais como visualizar, editar, criar e apagar um objeto, trocar o tema da aplicação e desconectar-se da mesma, representado na Figura 9 a seguir:

Figura 9 - Página principal da plataforma



Fonte: Autor.

3.2.1 Desconexão

Conforme visto na Figura 9, na parte superior existe um botão escrito “Sair” que ao ser clicado fará o processo de desconexão com a aplicação e após isso o usuário será redirecionado para a página de login.

3.2.2 Mudar tema

Conforme visto na Figura 9, na parte superior existe um botão com um ícone de sol que ao ser clicado altera o tema da aplicação podendo variar em claro ou escuro, sendo um ícone de sol e lua respectivamente.

O tema da aplicação não altera em nada as funcionalidades da plataforma, possuindo um papel estético na mesma.

3.2.3 Listagem de objetos

Conforme visto na Figura 9, abaixo do cabeçalho da página é possível encontrar o texto “Seus objetos” onde abaixo é renderizado a lista de objetos que esse usuário possui acesso.

3.2.4 Pesquisa de objetos

Conforme visto na Figura 9, abaixo do cabeçalho da página existe um *input* onde o usuário poderá digitar a informação que deseja buscar na base de dados ao clicar na tecla enter do teclado ou no ícone de lupa ao lado direito será realizada a pesquisa. O resultado implicará diretamente na listagem de objetos na tela.

3.2.5 Botão de criação de objeto

Conforme visto na Figura 9, abaixo do cabeçalho da página existe um botão "Criar", ao ser clicado o usuário será redirecionado para a página de criação de objetos.

3.2.6 AÇÕES DE CADA OBJETO NA LISTA

Figura 10 - Item da lista de objetos



Fonte: Autor.

Conforme visto na Figura 10, é exibido o nome do objeto, com um código que serve como um atributo de identificação e logo abaixo as informações de criação como o nome do usuário criador e a data.

Existem ações como expansão das informações, abrir a página de edição e apagar o objeto, seguindo respectivamente os ícones da direita para a esquerda.

3.2.7 Objeto expandido

Figura 11 - Detalhes de um objeto na lista



Fonte: Autor.

Conforme visto na Figura 11, ao abrir os detalhes do objeto é possível visualizar informações importantes tais como o verdadeiro conteúdo do objeto e seu

modelo. Não sendo possível editar essas informações apenas escrevendo dentro desse campo que se parece um editor de texto, para isso é necessário clicar no ícone de editar onde o usuário será redirecionado para a página da ação.

Ao lado desses campos que se destacam por conterem os valores é possível identificar um botão com uma nuvem que ao ser clicado baixará a informação referente ao bloco onde foi clicado em formato JSON.

3.3 CRIAÇÃO E EDIÇÃO

A página de criação e edição dos objetos e modelos é a mesma, sendo diferenciado apenas pelo comportamento inicial onde na edição os campos já serão preenchidos com as informações corretas e no final onde será chamado o *endpoint* responsável pela ação executada.

Figura 12 - Formulário para criação de um objeto e modelo



The screenshot displays the NuPPGIN web interface for creating an object and model. At the top left, the logo 'NuPPGIN' is visible. In the top right corner, there is a dark mode toggle switch and a 'Sair' button. The main heading is 'Criar Um Objeto'. Below this, there is a 'Nome' input field with a placeholder text 'Ex.: Mesa de escritório'. Underneath, there are two large, dark-themed text areas labeled 'Objeto' and 'Modelo', each with a '1' icon and a small edit icon. At the bottom right of the form, there is a blue 'Salvar' button.

Fonte: Autor.

Conforme visto na Figura 12, na tela de criação de um objeto e seu modelo temos três *inputs* sendo eles:

- Nome: Campo referente a identificação do objeto através da denominação

- Objeto e modelo: São campos especiais por possuírem uma validação e um funcionamento diferenciado focados em escritas em formato JSON.

Por fim, um botão “Salvar” quando a ação for de criar, ao ser clicado enviará uma requisição ao *back-end* e o mesmo notificará o *front-end* e um de “Editar” para quando realizar a edição que terá o mesmo comportamento supracitado de enviar uma requisição ao *back-end*. Diferenciando apenas em qual requisição será enviada.

A validação desses campos JSON, são importantes por facilitar a correção do erro ao apontar qual linha está o erro e ainda com uma mensagem explicando o mesmo, validação essa observável na Figura 13 logo abaixo:

Figura 13 - Validação do formulário de criação de objetos e modelos

NuPPGIN 🌙 Sair

Criar Um Objeto

Nome

Ex.: Mesa de escritório

Objeto:

```
1 {
2 }
3 }
```

Modelo:

```
1 {
2   "message": "Sou um objeto válido"
3 }
```

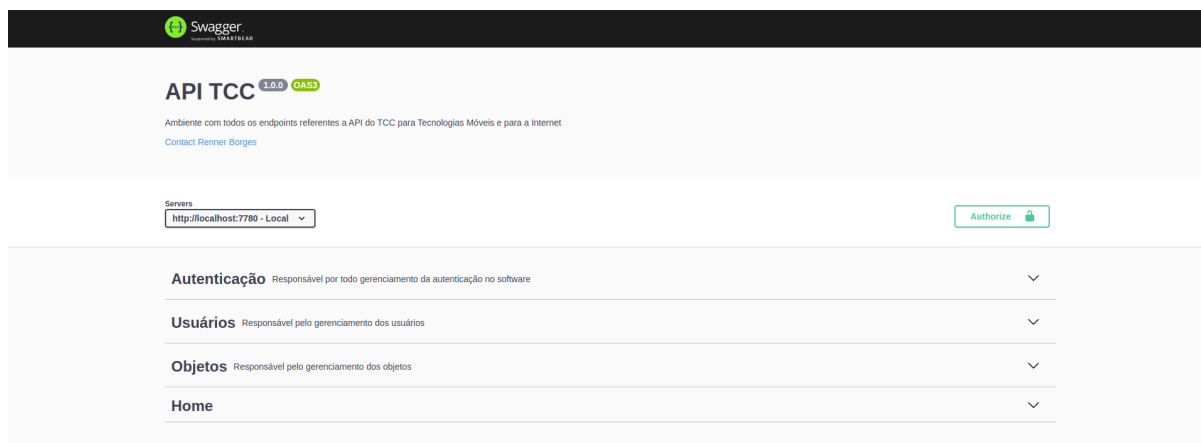
Salvar

Fonte: Autor.

3.4 DOCUMENTAÇÃO

A documentação de rotas facilita a integração com outros sistemas, tornando a solução escalável e fácil de integrar.

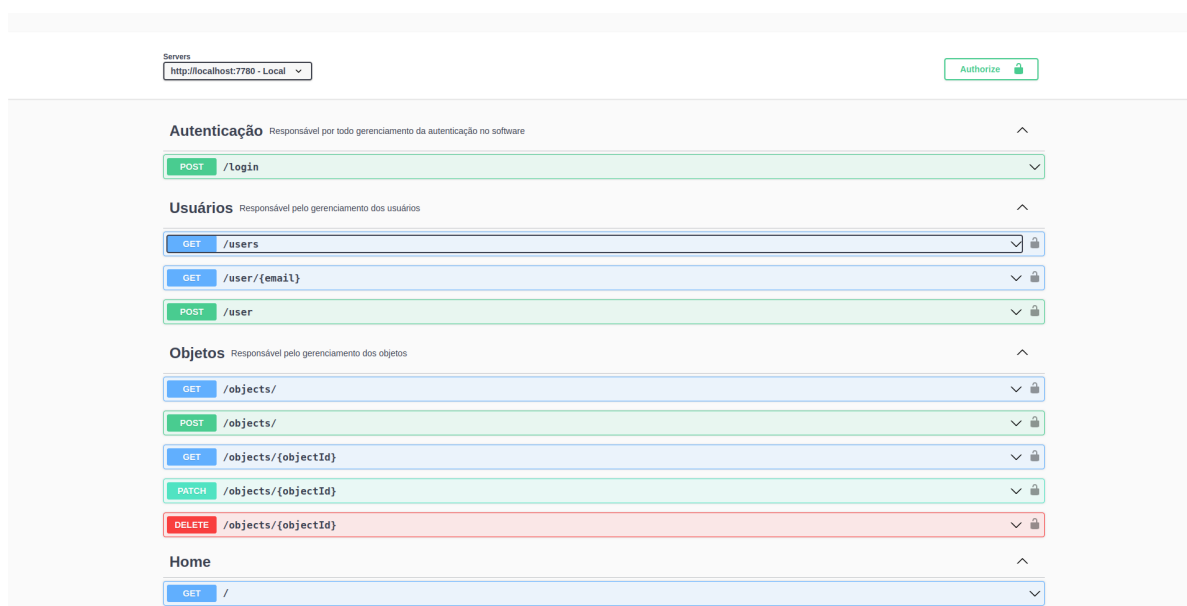
Figura 14 -Tela da documentação das rotas



Fonte: Autor.

Conforme a Figura 14, é possível observar um botão “Authorize” que ao clicar solicitará o *token* e logo abaixo em blocos separados por tópicos os conjuntos de rotas que compartilham entidades, ao clicar em qualquer bloco o mesmo abrirá mostrando as rotas disponíveis conforme a imagem abaixo:

Figura 15 -Tela da documentação das rotas com os blocos abertos



Fonte: Autor.

Conforme a Figura 15, é possível observar que é possível ver que a cor dos itens de cada bloco varia de acordo com seu método REST, permitindo assim uma fácil visualização.

Também é exibido o caminho da rota e se ela é protegida ou não.

Figura 16 - Tela da documentação de uma rota aberta

The screenshot displays a REST client interface for a POST request to the endpoint `/Login`. The request body is a JSON object: `{ "email": "renner@gmail.com", "password": "123456789e" }`. The response is a 200 OK with a JSON body: `{ "auth": true, "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1IjoiIjoiaWVudmVyeiJvcmdicyIsImVtYVlsIjoicmVudmVyeGd0YmlsLmV5bSIsImh0aCI6IjY3OTZlbnRlbnR1eGpaa14LP_oATBbc" }`. The response headers include `access-control-allow-origin: *`, `connection: keep-alive`, `content-length: 219`, `content-type: application/json; charset=utf-8`, `date: Sat, 18 Feb 2023 13:57:05 GMT`, `etag: W/"db-pERApZZU8Zv/BBm1b+qyP0k9VY"`, `keep-alive: timeout=5`, and `x-powered-by: Express`.

Code	Description	Links
200	OK	No links
401	Não autorizado	No links

Fonte: Autor.

Conforme a Figura 16, é possível observar bastante detalhes da rota como: seus parâmetros, corpo esperado e possíveis retornos.

Caso pressionado o botão “Execute” é realizada uma chamada na API com as informações fornecidas.

3.5 CONSIDERAÇÕES FINAIS

Em suma, é incontestável que entender o funcionamento da plataforma é essencial para o uso correto da mesma. Neste capítulo foi apresentado a funcionalidade dos principais fluxos alinhados aos casos de usos já supracitados.

Os recursos apresentados neste capítulo podem ser visualizados em um [vídeo](#) que mostra o comportamento da plataforma. Esse vídeo foi postado em uma plataforma de vídeos online, permitindo que os interessados confirmem os recursos em ação. Assim, assistir ao vídeo pode ser uma excelente forma de se familiarizar com a plataforma e entender como ela pode ser utilizada para atender às necessidades dos usuários (Ferreira, 2023).

4 TESTES DE FUNCIONAMENTO

4.1 CONSIDERAÇÕES INICIAIS

Os testes da plataforma foram realizados utilizando o método de teste de caixa preta, no qual o foco são nas funcionalidades e não em como essas funcionalidades foram desenvolvidas ao nível de código. Este teste será aplicado aos UC01, UC02 e UC03, já que são esses que apresentam maior importância para a plataforma.

4.2 TESTE DE CAIXA PRETA

Os testes de caixa preta também são conhecidos como testes funcionais, sendo um tipo de teste de software que avalia a funcionalidade do sistema sem considerar a estrutura e funcionamento interno do mesmo, onde o testador não possui acesso ao código-fonte ou na arquitetura interna.

4.3 VALIDAÇÃO DOS CASOS DE TESTE

A validação dos casos de teste é uma etapa pertinente do processo, ao garantir que os casos de teste sejam relevantes, precisos e eficazes na detecção dos problemas ou defeitos no sistema.

4.3.1 Gerência de objetos e modelos

Neste teste é avaliado o UC01 e UC02, que dá início ao teste de caixa preta validando o funcionamento das ações de criar e editar um objeto e modelo. No teste realizou-se o preenchimento de todas as informações para validar seu funcionamento, este retornou o estado correto retornando as saídas esperadas.

Tabela 5 - Teste UC01 e UC02

Caso de uso	Condição de entrada	Classes válidas	Classes inválidas
<i>UC01 - Gerenciar os objetos e UC02 - Gerenciar os modelos</i>	<i>Inserir o nome</i>	<i>1. Nome do tipo de texto.</i>	<i>2. Nome sem nada preenchido</i>
	<i>Inserir o modelo</i>	<i>3. Modelo no formato correto do JSON</i>	<i>4. Modelo com JSON incorreto</i>
			<i>5. Modelo preenchido com outros tipos de dados sem ser JSON</i>
			<i>6. Modelo sem nada preenchido</i>
	<i>Inserir o objeto</i>	<i>7. Objeto no formato correto do JSON</i>	<i>8. Objeto com JSON incorreto</i>
			<i>9. Objeto preenchido com outros tipos de dados sem ser JSON</i>
			<i>10. Objeto sem nada preenchido</i>
	<i>Click no botão de envio do formulário</i>	<i>11. Enviar a requisição ao back-end e validar os campos.</i>	<i>12. Enviar a requisição sem os campos devidamente preenchidos</i>
			<i>13. Não retornar nenhuma informação</i>

Fonte: Autor.

Tabela 6 - Resultado do teste UC01 e UC02

Condição de entrada	Classe	Resposta
<i>Inserir o nome</i>	<i>1. Nome do tipo de texto.</i>	<i>O sistema aceitou o nome do tipo texto</i>
	<i>2. Nome sem nada preenchido</i>	<i>O sistema rejeitou o nome vazio</i>

<i>Inserir o modelo</i>	<i>3. Modelo no formato correto do JSON</i>	<i>O sistema aceitou o modelo com o JSON correto</i>
	<i>4. Modelo com JSON incorreto</i>	<i>O sistema rejeitou o modelo com o JSON incorreto</i>
	<i>5. Modelo preenchido com outros tipos de dados sem ser JSON</i>	<i>O sistema rejeitou valores diferentes de um JSON</i>
	<i>6. Modelo sem nada preenchido</i>	<i>O sistema rejeitou o modelo vazio</i>
<i>Inserir o objeto</i>	<i>7. Modelo no formato correto do JSON</i>	<i>O sistema aceitou o objeto com o JSON correto</i>
	<i>8. Modelo com JSON incorreto</i>	<i>O sistema rejeitou o objeto com o JSON incorreto</i>
	<i>9. Modelo preenchido com outros tipos de dados sem ser JSON</i>	<i>O sistema rejeitou valores diferentes de um JSON</i>
	<i>10. Modelo sem nada preenchido</i>	<i>O sistema rejeitou o modelo vazio</i>
<i>Click no botão de envio do formulário</i>	<i>11. Enviar a requisição ao back-end e validar os campos.</i>	<i>O sistema aceitou e validou os campos ao enviar</i>
	<i>12. Enviar a requisição sem os campos devidamente preenchidos</i>	<i>O sistema validou e não enviou a requisição</i>
	<i>13. Não retornar nenhuma informação</i>	<i>O sistema retornou informação referente a requisição</i>

Fonte: Autor.

4.3.2 Autenticação

Neste teste é avaliado o UC03, relacionado ao funcionamento da ação de se autenticar na plataforma. No teste realizou-se o preenchimento de todas as informações para validar seu funcionamento, este retornou o estado correto retornando as saídas esperadas.

Tabela 7 - Teste UC03

Caso de uso	Condição de entrada	Classes válidas	Classes inválidas
<i>UC03 - Autenticar</i>	<i>Inserir o usuário</i>	<i>1. Usuário no formato</i>	<i>2. Usuário no formato</i>

		<i>de um e-mail válido.</i>	<i>de um e-mail inválido</i>
			<i>3 - Usuário vazio</i>
	<i>Inserir a senha</i>	<i>4 - Senha no formato válido</i>	<i>5 - Senha no formato inválido</i>
			<i>6 - Senha vazia</i>
	<i>Click no botão de envio do formulário</i>	<i>7. Enviar a requisição ao back-end e validar os campos.</i>	<i>8. Enviar a requisição sem os campos devidamente preenchidos</i>
			<i>9. Não retornar nenhuma informação</i>
		<i>10 - Retornar o token caso as credenciais sejam válidas</i>	<i>11 - Retornar o token com as credenciais inválidas</i>

Fonte: Autor.

Tabela 8 - Resultado do teste UC03

Condição de entrada	Classe	Resposta
<i>Inserir o usuário</i>	<i>1. Usuário no formato de um e-mail válido.</i>	<i>O sistema aceitou o usuário no formato de e-mail</i>
	<i>2. Usuário no formato de um e-mail inválido</i>	<i>O sistema rejeitou o usuário no formato inválido de e-mail</i>
	<i>3 - Usuário vazio</i>	<i>O sistema rejeitou o usuário vazio</i>
<i>Inserir a senha</i>	<i>4 - Senha no formato válido</i>	<i>O sistema aceitou a senha no formato válido seguindo à regra de negócio da plataforma</i>
	<i>5 - Senha no formato inválido</i>	<i>O sistema rejeitou a senha no formato inválido seguindo à regra de negócio da plataforma</i>
	<i>6 - Senha vazia</i>	<i>O sistema rejeitou a senha vazia</i>
<i>Click no botão de envio do formulário</i>	<i>7. Enviar a requisição ao back-end e validar os campos.</i>	<i>O sistema validou os campos e enviou a requisição</i>
	<i>8. Enviar a requisição sem os campos devidamente preenchidos</i>	<i>O sistema validou e não enviou a requisição</i>
	<i>9. Não retornar nenhuma informação</i>	<i>O sistema retornou informação referente a requisição</i>
	<i>10 - Retornar o token caso as credenciais sejam válidas</i>	<i>O sistema retornou o token</i>

	<i>11 - Retornar o token com as credenciais inválidas</i>	<i>O sistema recusou e não enviou o token</i>
--	---	---

Fonte: Autor.

4.4 CONSIDERAÇÕES FINAIS

Os testes de caixa preta nos casos de uso já supracitado, corresponderam ao esperado, comportando-se corretamente em todos as etapas realizadas, considerando que os casos foram escolhidos dentre os casos de usos com mais relevância na plataforma.

No geral o sistema se comportou corretamente, não tendo erros que impossibilitem o uso do mesmo, passando assim eficazmente pelos testes de caixa preta.

5. CONCLUSÃO

Este projeto apresenta uma plataforma para gerenciamento de artefatos inerentes na construção de ambientes 3D, que ajuda a solucionar problemas como falhas no posicionamento dos modelos em cena e inserção equivocada dos mesmos. Além disso, a plataforma oferece recursos semi-automatizados para ajudar no processo de produção da cena. A proposta surgiu da necessidade da área de Computação Gráfica e das potencialidades da área de Sistemas Gestores de Informações, visando elaborar uma plataforma capaz de gerir e prover recursos na construção de ambientes 3D.

O objetivo proposto neste projeto foi plenamente alcançado através da implementação de uma solução computacional eficaz e de fácil integração, capaz de auxiliar na gestão de artefatos associados à construção de ambientes virtuais 3D. A solução desenvolvida possibilita uma gestão mais eficiente e automatizada dos recursos envolvidos no processo, minimizando os impactos causados pela problemática. Com a utilização desta solução, a equipe responsável pela gestão de artefatos pôde otimizar seus processos de trabalho, reduzir custos e evitar possíveis retrabalhos, contribuindo para a melhoria da qualidade do produto final e para a satisfação dos clientes.

Essa solução também disponibilizou recursos de autenticação para o controle de acessos, possibilitando que apenas usuários autorizados acessem a solução, uma documentação viabilizando acesso às informações necessárias a aspectos de integração com ferramentas externas e por fim uma interface gráfica visual intuitiva, que permite a visualização clara e concisa dos dados.

Ao longo do desenvolvimento, uma das premissas principais foi criar uma ferramenta eficiente. Após os testes de caixa preta, foi possível notar que a ferramenta possui a robustez necessária para ser usada em ambientes de produção reais. Ao resgatar a eficiência nos testes da ferramenta, é possível fornecer aos usuários uma solução que os ajuda a gerenciar artefatos de ambientes virtuais 3D complexos de maneira mais rápida e eficiente.

5.1 TRABALHOS FUTUROS

Um dos principais trabalhos futuros será o de acompanhar a equipe de desenvolvimento de ambientes virtuais tridimensionais. Uma vez que a ferramenta esteja em produção, será necessário manter uma equipe dedicada a monitorar a ferramenta e a implementar melhorias contínuas. Isso garantirá que a ferramenta continue a atender às necessidades dos usuários e permaneça atualizada em relação às tecnologias emergentes.

Outro trabalho futuro será a expansão dos recursos da ferramenta. Isso incluirá a adição de novas funcionalidades. A equipe de desenvolvimento deverá trabalhar em estreita colaboração com os usuários finais para entender suas necessidades e garantir que a ferramenta continue a evoluir de acordo com essas necessidades.

Entre os recursos mapeados para serem adicionados em trabalhos futuros podemos citar: estabelecer um sistema de versionamento para objetos, e criação de espaços de trabalho compartilhados e privados.

Finalmente, também será importante continuar realizando testes de escalabilidade sob pressão da plataforma. Isso garantirá que a ferramenta possa lidar com um número crescente de usuários e permaneça estável mesmo em momentos de pico de uso. Além disso, serão necessários testes de segurança para garantir que a ferramenta permaneça protegida contra possíveis ameaças.

Em resumo, este projeto representou um primeiro passo na criação de uma ferramenta de gestão de artefatos para ambientes 3D. Há ainda trabalho a ser feito para tornar essa ferramenta um produto de alta qualidade que atenda às necessidades dos usuários e permaneça relevante em um ambiente de tecnologia em constante evolução.

REFERENCIAS

MICHALOS, George ; KARVOUNIARI, Anna ; DIMITROPOULOS, Nikolaos ; TOGIAS, Theodoros ; MAKRIS, Sotiris *et al.* **Workplace analysis and design using virtual reality techniques**. Manufacturing Technology, 2018.

Javascript, 2023. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Web/JavaScript>. Acesso em: 06, fevereiro de 2023. sem autor: JavaScript | MDN

React, 2023. Disponível em: <https://reactjs.org/docs/getting-started.html>. Acesso em: 01, janeiro de 2023. sem autor: Getting Started – React

Next.js, 2023. Disponível em: <https://nextjs.org/>. Acesso em: 10, fevereiro de 2023. sem autor: Next.js

NextUI, 2023. Disponível em: <https://nextui.org/docs/guide/getting-started>. Acesso em: 11, janeiro de 2023. sem autor: Getting started | NextUI - Beautiful, fast and modern React UI Library

Node.js, 2023. Disponível em: <https://nodejs.org/pt-br/docs/>. Acesso em: 01, janeiro de 2023. sem autor: Node.js

Express, 2023. Disponível em: <https://expressjs.com/pt-br/>. Acesso em: 05, janeiro de 2023. sem autor: Express

MongoDB, 2023. Disponível em: <https://www.mongodb.com/docs/>. Acesso em: 07, fevereiro de 2023. sem autor: MongoDB Documentation

YUP, 2023. Disponível em: <https://github.com/jquense/yup>. Acesso em: 07, fevereiro de 2023. sem autor: GitHub - jquense/yup: Dead simple Object schema validation

Swagger, 2023. Disponível em: <https://swagger.io/docs/>. Acesso em: 15, fevereiro de 2023. sem autor: Swagger Documentation

Ferreira, Renner Borges Ferreira. NuPPGIN Dashboard. YouTube, 1 de março de 2023. Disponível em: <https://youtu.be/yplyidb7SRQ>. Acesso em: 10, março de 2023.