



**CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E  
DESENVOLVIMENTO DE SISTEMAS**

**PAULO HENRIQUE DA SILVA**

**NEXUS: Sistema de gerenciamento de ativos de TI**

**Guarulhos**

**2024**

**PAULO HENRIQUE DA SILVA**

**NEXUS: Sistema de gerenciamento de ativos de TI**

Trabalho de Graduação do Curso de Análise e Desenvolvimento de Sistemas, apresentado como requisito parcial para obtenção do Título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

**Orientadora:** Prof<sup>ª</sup> Esp. Jane Maria Dos Santos Eberson

**Guarulhos**

**2024**

**PAULO HENRIQUE DA SILVA**

**NEXUS: Sistema de gerenciamento de ativos de TI**

Trabalho de Graduação apresentado ao Curso de Análise e Desenvolvimento de Sistemas como requisito parcial para obtenção do **Título de Tecnólogo em Análise e Desenvolvimento de Sistemas**.

**Banca Examinadora**

**Orientador:** \_\_\_\_\_

Profª Esp. Jane Maria Dos Santos Eberson  
Fatec Guarulhos

**Banca:** \_\_\_\_\_

Titulação + nome completo  
IES de origem

**Banca:** \_\_\_\_\_

Titulação + nome completo  
IES de origem

Guarulhos, 28/06/2024

## RESUMO

SILVA, Paulo Henrique. **NEXUS: Sistema de gerenciamento de ativos de TI**. 2024. X p. Trabalho de Conclusão de Curso – Faculdade de Tecnologia de Guarulhos, Guarulhos.

Este trabalho teve o intuito de desenvolver uma aplicação web chamada NEXUS capaz de administrar o gerenciamento de ativos de uma organização. Para criar uma base teórica, a fim de obter os requisitos de um software do gênero, foi feita uma pesquisa por artigos e trabalhos que comentam sobre os assuntos de gestão e ativos de TI. Após isso, foram estruturados os requisitos da aplicação, os testes e a documentação do trabalho para organizar o desenvolvimento. Logo após, o software foi feito utilizando as tecnologias .NET e Angular, e os resultados foram apresentados conforme um guia de usuário dos serviços existentes. Por fim, foi percebido que o sistema é um protótipo funcional capaz de melhorar a produtividade dos processos de negócio, porém exige maior trabalho caso queira ser implementado em um ambiente produtivo real. Desse modo, o trabalho proporciona essa possibilidade para desenvolvedores que estejam interessados no tema e gostariam de aplicar a solução.

**Palavras-chave:** Ativos de TI, Sistemas de gestão e Monitoramento da infraestrutura de TI

## **ABSTRACT**

This project had the objective to develop a web application called NEXUS capable of managing an organization's asset management. To create a theoretical base to obtain the requirements for the software, a search was conducted for articles and papers discussing the topics of management and IT assets. After that, the application requirements, tests, and work documentation were structured to organize the development. Subsequently, the software was created using .NET and Angular technologies, and the results were presented with a user guide about the existing services. Finally, it was found that the system is a functional prototype capable of improving the productivity of business processes, but it requires further work to be implemented in a real productive environment. Therefore, this work provides this possibility for developers who are interested in the topic and would like to apply the solution.

**Keywords:** IT Assets, Management Systems, IT Infrastructure Monitoring

## LISTA DE FIGURAS

Figura 1: Acessos dos perfis .....	32
Figura 2: Diagrama de negócios de verificação de manutenção .....	34
Figura 3: Diagrama de negócios de análise de requisição .....	35
Figura 4: Casos de uso de permissões gerais .....	36
Figura 5: Casos de uso das funções administrativas .....	37
Figura 6: Casos de uso de verificação de manutenção .....	37
Figura 7: Casos de uso de análise de requisição .....	38
Figura 8: Cliente-Servidor .....	39
Figura 9: Arquitetura MVC .....	41
Figura 10: Exemplo de estrutura de módulo .....	42
Figura 11: Diagrama das classes bases .....	43
Figura 12: Diagrama de classe da estrutura básica .....	45
Figura 13: Diagrama de classes de DTOs .....	47
Figura 14: Diagrama de NexusListaRespostaDTO .....	49
Figura 15: Diagrama de UsuarioPerfil .....	51
Figura 16: Diagrama de ciclos de vida .....	52
Figura 17: Banco de dados de administração .....	53
Figura 18: Banco de dados dos ativos .....	54
Figura 19: Banco de dados de Ciclo de Vida .....	55
Figura 20: Tela de login .....	59
Figura 21: Tela de perfil recém-criado .....	60
Figura 22: Tela de alterar perfil para usuários novos .....	61
Figura 23: Perfil de coordenador selecionado .....	61
Figura 24: Perfil de leitor selecionado .....	62
Figura 25: Tela de Ativos .....	63
Figura 26: Tela de atribuições .....	64
Figura 27: Tela de notificações .....	64
Figura 28: Tela de mudança de perfil .....	65
Figura 29: Tela de mudança de senha .....	65
Figura 30: Tela de editar senha .....	66
Figura 31: Tela de opções de Componentes .....	66
Figura 32: Tela de criação de objeto .....	67
Figura 33: Tela de busca de objetos .....	68
Figura 34: Tela de busca com filtro de nome .....	68
Figura 35: Opções de um objeto .....	69
Figura 36: Tela de edição de objeto .....	70
Figura 37: Caixa de diálogo para excluir objeto .....	71
Figura 38: Tela de detalhes dos objetos .....	71
Figura 39: Ações de adicionar manutenção e software .....	72
Figura 40: Tela de administração para perfil administrador .....	73
Figura 41: Tela de administração para perfil oráculo .....	73
Figura 42: Ação de gerenciar perfis .....	74
Figura 43: Tela de gerenciar perfis .....	75
Figura 44: Menu de requisições .....	75

Figura 45: Tela de criação de requisição .....	76
Figura 46: Tela de atribuições com requisição .....	77
Figura 47: Tela de análise de requisição .....	77
Figura 48: Notificação de requisição reprovada .....	78
Figura 49: Atribuição de concluir requisição .....	78
Figura 50: Tela de concluir requisição .....	79
Figura 51: Notificação de requisição aprovada .....	79
Figura 52: Notificação de conclusão da requisição .....	80
Figura 53: Ação de adicionar manutenção .....	81
Figura 54: Tela de criação de manutenção .....	81
Figura 55: Atribuição de completar manutenção .....	82
Figura 56: Atribuição de completar manutenção sem ações .....	82
Figura 57: Tela de preenchimento de solução .....	83
Figura 58: Ação de completar manutenção .....	83
Figura 59: Notificação de conclusão de manutenção .....	84

## LISTA DE QUADROS

Quadro 1: Lista de sinônimos considerados na pesquisa .....	13
Quadro 2: Critérios de exclusão .....	14
Quadro 3: Descrição dos ativos da aplicação .....	24
Quadro 4: Atributos dos componentes .....	25
Quadro 5: Atributos dos equipamentos .....	26
Quadro 6: Atributos de localizações .....	27
Quadro 7: Atributos de manutenções .....	28
Quadro 8: Atributos de requisições .....	29
Quadro 9: Atributos de softwares .....	29
Quadro 10: Atributos de Usuários .....	30
Quadro 11: Perfis de acesso do sistema .....	31
Quadro 12: Descrição dos módulos do sistema .....	39

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>11</b>
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	<b>13</b>
2.1	Revisão Sistemática	13
2.2	Ativos de TI	15
2.2.1	Visão Geral	15
2.2.2	Ativos no Contexto de TI	15
2.3	Sistemas de gestão	17
2.4	Monitoramento da infraestrutura de TI	18
2.4.1	Gestão de Ativos	18
2.4.2	Ciclo de vida do ativo	19
<b>3</b>	<b>METODOLOGIA</b>	<b>21</b>
3.1	Levantamento de Requisitos	21
3.2	Prototipação e Documentação	21
3.3	Testes	22
<b>4</b>	<b>DESENVOLVIMENTO</b>	<b>23</b>
4.1	Especificação funcional	23
4.1.1	Requisitos	23
4.1.2	Regras de negócio	24
4.1.2.1	Ativos	24
4.1.2.2	Projetos	24
4.1.2.3	Componentes	25
4.1.2.4	Equipamentos	26
4.1.2.5	Localizações	27
4.1.2.6	Manutenções	28
4.1.2.7	Requisições	29
4.1.2.8	Softwares	29
4.1.2.9	Usuários	30
4.1.2.10	Perfis	30
4.1.2.11	Atribuições	33
4.1.3	Ciclos de vida	33
4.1.3.1	Verificação de manutenção	33
4.1.3.2	Análise de requisição	34
4.1.4	Diagramas de caso de uso	35
4.1.4.1	Permissões gerais	35
4.1.4.2	Administração	36

4.1.4.3	Verificação de Manutenção .....	37
4.1.4.4	Análise de Requisição .....	38
4.2	Especificação técnica .....	38
4.2.1	Arquitetura .....	38
4.2.1.1	Módulos .....	39
4.2.1.2	MVC .....	40
4.2.2	Diagramas de classe .....	42
4.2.2.1	<i>Data Transfer Object</i> - DTOs .....	46
4.2.2.2	UsuarioPerfil .....	50
4.2.2.3	Ciclos de vida .....	52
4.2.3	Arquitetura do banco de dados .....	53
4.2.4	HTML, CSS e JavaScript .....	56
4.2.5	Angular .....	56
4.2.6	.NET – C# .....	57
4.2.7	JWT .....	57
4.2.8	SQLServer .....	57
4.2.9	Bibliotecas e frameworks .....	57
<b>5</b>	<b>RESULTADOS E DISCUSSÕES .....</b>	<b>59</b>
5.1	Login .....	59
5.2	Alterar perfil ou projeto .....	60
5.3	Menu inicial .....	62
5.4	Adicionar objetos .....	66
5.5	Buscar objetos .....	67
5.6	Editar objetos .....	69
5.7	Deletar objetos .....	70
5.8	Detalhes de objeto .....	71
5.9	Adicionar manutenções e softwares .....	72
5.10	Administração .....	72
5.11	Gerenciar Perfis .....	74
5.12	Análise de requisição .....	75
5.13	Verificação de manutenção .....	80
<b>6</b>	<b>CONSIDERAÇÕES FINAIS .....</b>	<b>85</b>

## 1 INTRODUÇÃO

Conforme Parente (2021) em um âmbito em que equipamentos de Tecnologia da Informação - TI são utilizados no cotidiano como uma das principais ferramentas de trabalho, se faz necessário o planejamento estratégico por parte da equipe de TI. Conseqüentemente, faz parte do plano a escolha das ferramentas mais adequadas para realizar a gestão das tecnologias na organização, independentemente de ser pública ou privada, uma vez que tecnologia da informação é algo que está presente em todos os ambientes empresariais. Portanto, será que uma aplicação web capaz de administrar os ativos da infraestrutura de TI melhoraria a gestão de uma organização?

Dito isso, este trabalho propõem o desenvolvimento de um sistema web voltado para o gerenciamento de ativos de TI que possuirá o nome **NEXUS**, logo é voltado em administrar o fluxo de transformações do qual um ativo sofre enquanto existir e for relevante para a empresa. Para tal, ativos de TI são objetos de tecnologia da informação, como computadores, switches, *access points*, que possuam valor para a organização que o pertence. Com isso, surgem os softwares de gestão que podem monitorar muitos tipos de ativos, como documentos, projetos, controles de fluxo de informação, mercadorias e, inclusive, ativos de TI. Logo, é possível combinar eficiência dos sistemas de gerenciamento com a devida abstração das políticas de uma companhia para desenvolver softwares para monitorar toda uma infraestrutura de TI.

É nítido que o investimento em ferramentas tecnológicas de cunho informacional é necessário para que os processos de tomada de decisão levem ao alcance dos objetivos (MORAES et al.,2018), e uma vez que um sistema NEXUS será criado com o objetivo de fornecer funções administrativas e recursos informativos que ajudem colaboradores, o software torna-se uma contribuição importante. Dessa forma, justificando a existência e o desenvolvimento desse projeto como um todo.

Assim sendo, o trabalho constituirá na fundamentação teórica da monografia, abordando os temas principais do projeto. Após isso, será comentado a metodologia do trabalho. Em seguida, serão discutidas questões técnicas como a arquitetura do projeto e o desenvolvimento, como as tecnologias usadas e as imagens das

interfaces de usuário. Por fim, serão apresentadas as considerações e resultados do trabalho após a realização.

## 2 FUNDAMENTAÇÃO TEÓRICA

Na fundamentação teórica deste trabalho será abordado brevemente a revisão sistemática da literatura – RSL feita com objetivo de adquirir artigos e dissertações para sustentar a base do projeto. Após isso, os temas principais do trabalho, tal como alguns conceitos chaves, serão discutidos nas seções de Ativos de TI, Sistemas de gestão e monitoramento e infraestrutura de TI a partir dos artigos obtidos na RSL.

### 2.1 Revisão Sistemática

Esta revisão sistemática da literatura tem como propósito obter estudos e artigos científicos que, de alguma forma, corroborem positivamente para o desenvolvimento do sistema em questão. A pesquisa foi feita utilizando base de dados de artigos e teses científicas, filtrados por uma combinação de sinônimos e palavras chaves e por critérios de exclusão. Em seguida, será apresentada a metodologia do RSL, tão como os resultados obtidos.

Para montar a fonte de pesquisa, foram escolhidas base de dados comuns e conhecidas nacionalmente. Para tal, as seguintes fontes foram utilizadas: Biblioteca de dissertações da USP, a biblioteca digital brasileira de dissertações - BDTD e o Google Scholar.

Para determinar como as publicações seriam procuradas, um conjunto de palavras chaves foi feito para filtrar as pesquisas, em uma combinação de sinônimos em inglês e português, conforme o quadro 1:

Quadro 1: Lista de sinônimos considerados na pesquisa

<b>Palavras Chaves</b>	<b>Sinônimos em inglês</b>	<b>Sinônimos em português</b>
<b>Ativos de TI</b>	IT Assets	Ativos de TI
<b>Sistema</b>	System	Sistema
<b>Gerenciamento de TI</b>	TI Management	Gerenciamento de TI

Fonte: Desenvolvido pelo autor, 2024

Logo após, foi preciso realizar a análise do conteúdo de cada resultado encontrado, a fim de verificar se a publicação realmente estava de acordo com o objeto do trabalho. Portanto, foram estabelecidos critérios de exclusão, conforme o quadro 2:

Quadro 2: Critérios de exclusão

Critério	Descrição
CE1	Artigos que não possuem TI ou gestão ou como tema principal
CE2	Artigos publicados antes do ano 2000

Fonte: Desenvolvido pelo autor, 2024

O critério CE1 foi estabelecido pois dificilmente um artigo de um tema muito distinto do trabalho seria aproveitado para o desenvolvimento do software, e o critério CE2 surgiu por conta do foco do projeto, como o objetivo é criar um aplicativo moderno para gestão de TI, dissertações muito antigas podem oferecer uma visão desatualizada da realidade tanto para desenvolvimento de software como para gestão de TI.

Após o filtro, alguns artigos e dissertações foram escolhidos e analisados para contribuir com o projeto. Foi feita uma leitura a fim de obter conclusões sobre os temas com os trabalhos, o que resultou nas considerações a seguir.

Para a palavra-chave **Ativos de TI**, foi percebido o foco na gestão da informação. Por exemplo, Pinheiro (2011) citou que um software do tipo deve ser capaz de otimizar o fluxo de informação permitindo mais agilidade, redução de custos e ganho de produtividade, além de mais segurança de acesso à informação. O que corrobora com a ideia proposta inicialmente de controlar as transformações sofridas pelo ativo por meio de um ciclo de vida.

Já no termo **Sistema**, os resultados apresentaram o que se espera para as características funcionais básicas de um sistema. Para Parente (2021), o planejamento estratégico da equipe de TI pode ser feito a partir da constante verificação dos dispositivos e seus softwares. Assim, é possível concluir que o software deve possuir funcionalidades de buscar, criar, editar e excluir diversos tipos de ativos, não só computadores como também equipamentos (memória RAM, processadores, periféricos etc.), para que a verificação citada seja feita com êxito.

Por fim, para a palavra-chave **Gerenciamento de TI**, são demonstrados os aspectos de um ambiente de TI, como no trabalho de Suckow (2011) do qual o autor afirma que um bom ambiente gerenciável proporciona ao administrador uma grande quantidade de informações fundamentais. Após isso, ainda no trabalho Suckow (2011), foram discutidos os principais tópicos para construir o modelo de

gerenciamento de TI. Desse modo, foi possível coletar as características importantes para modelar as funcionalidades do sistema.

Há mais conclusões e resultados pertinentes encontrados durante a RSL que forneceram uma base conceitual para montar o escopo e entender às capacidades e limitações de um software do mesmo tipo. Contudo, esses artigos serão abordados nas próximas subseções.

## **2.2 Ativos de TI**

### **2.2.1 Visão Geral**

Durante a leitura dos artigos obtidos, foram encontradas discussões interessantes acerca de ativos, tanto no contexto de TI como de forma geral. Sendo assim, podem ser considerados ativos todo e qualquer elemento físico ou não, que seja controlado pela empresa e que a ela proporcione a possibilidade de obtenção de fluxos de caixa (PEREZ; FAMÁ, 2006). Por exemplo, um simples documento que contém o relatório mensal da empresa, seja esse digitalizado ou impresso, já pode ser considerado um ativo, pois as informações contidas no recurso podem ser utilizadas para melhorar as operações diárias e, conseqüentemente, o lucro.

Além do mais Perez e Famá (2006) complementam abordando no contexto de contabilidade básica, em que são considerados ativos os bens e os direitos de uma entidade, expressos em moeda e à disposição da organização. Por conseqüência, dependendo do contexto, uma exata definição de ativo pode variar, considerando a distinta natureza de cada empresa gerir seus negócios diários.

Porém, é possível dar uma descrição comum para ativo que funcione para qualquer situação. Logo, um ativo é todo recurso (tangível ou intangível) que esteja sob o controle de uma organização e possa ser utilizado para produzir produtos ou serviços aos seus clientes, visando gerar benefícios econômicos (PEREZ; FAMÁ, 2006). Em outros termos, é tudo que possui valor para uma companhia.

### **2.2.2 Ativos no Contexto de TI**

Agora alterando o contexto do caso, é possível descrever ativos em uma visão voltada a TI, como relata Silva e Pinto (2019), em uma rede de computadores de uma empresa, os ativos de rede integram um conjunto de dispositivos e suas

informações relacionadas que permitem operar os negócios da organização. Dessa forma, um ativo de TI não é apenas um computador e sim qualquer recurso da tecnologia de informação capaz de se comunicar e otimizar os negócios.

Dentre os ativos de TI mais comuns e conhecidos, é possível citar os computadores e periféricos (mouses, teclados, monitores). Há também os menos conhecidos pelo público, porém não menos importantes, como switches e roteadores. Em um contexto ideal de gerenciamento de TI, todo ativo do tipo deve ser rastreado e monitorado.

Contudo, durante a pesquisa, foi observado uma característica interessante na gestão de ativos: o monitoramento da informação. Como afirma Silva e Pinto (2019, p.57), “gerenciar uma rede de computadores é uma tarefa de monitoramento dos elementos da rede, sejam eles hardware ou software que façam parte da mesma”. Inclusive, para a tecnologia da informação, conhecer os recursos informáticos é a base para manter um ambiente onde a informação possa fluir de maneira segura e confiável (SILVA; PINTO, 2019).

O que implica em além de rastrear o ativo, é necessário monitorar seus atributos, como nome, descrição, data de cadastro, usuário responsável etc. Tão como seus relacionamentos com outros ativos, exemplificando, quais componentes estão conectados via cabo ou quais equipamentos estão acoplados fisicamente.

Logo, como indica Suckow (2011), o principal recurso administrado na TI não são os ativos em si, e sim a informação. Assim, é correto afirmar que o mais importante no controle dos ativos é garantir que suas informações estejam de acordo com as necessidades das empresas. Dessa forma:

A informação se tornou um recurso essencial para o gerenciamento de organizações modernas. Isso porque o ambiente de negócios atual é volátil, dinâmico, turbulento e necessita da crescente demanda por informação precisa, relevante, completa, oportuna e econômica necessária para guiar a tomada de decisões com o objetivo de melhorar as habilidades da organização de gerir oportunidades e ameaças. (Ada; Ghaffarzadeh, 2015, p.1)

Para cada companhia, as oportunidades e ameaças geridas podem variar. Contudo, geralmente em uma empresa privada o uso da TI busca encontrar o máximo de benefícios que a tecnologia pode oferecer para obter mais vantagem em relação à concorrência (SUCKOW, 2011).

Por fim, foi observado durante a pesquisa a necessidade de mudança das empresas ocasionada pelo uso da tecnologia da informação, dado que é preciso adaptar ou alterar por completo processos de negócios para conseguirem implantar ferramentas tecnológicas melhores, como os próprios sistemas de gestão. Dessa forma, muitas vezes uma organização necessita mudar regras tradicionais e não condizentes que com a realidade atual para adotar processos novos e atualizados, o que implica em “adaptar os ativos já existentes utilizando de ferramentas e metodologias que já estão consolidadas no mercado...” (SUCKOW, 2011, p.15).

### **2.3 Sistemas de gestão**

No contexto mais voltado para a gestão, durante a pesquisa de artigos, foi visto conforme Moraes et al. (2018) que para desenvolverem e atuarem ativamente frente à concorrência, ampliando clientes e mercados, necessitam de informações qualificadas. Apesar da gestão de informação já ter sido discutida anteriormente nesse trabalho, novos conceitos puderam ser obtidos e serão discutidos nessa seção.

Dentre os assuntos observados, o fluxo de informação, outro nome para ciclo de vida, mostrou certo destaque, pois o fluxo de informações funciona como o elemento unificador entre o uso ativos e sistemas de informações, facilitando tomadas de decisão e manuseio da tecnologia e das pessoas em nível operacional e estratégico (MORAES et al., 2018).

Além do mais, é possível completar o raciocínio ao dizer que:

Para gerenciar fluxos informacionais, quer formais ou informais, é necessário realizar algumas ações integradas objetivando tratar e disseminar todo o ativo informacional e intelectual da organização, incluindo desde documentos, bancos, bases de dados, produzidos interna e externamente à organização até o conhecimento individual dos diferentes atores existentes na organização. (Pinheiro, 2012, p.7)

Portanto, é correto afirmar que controlar as transformações dos ativos por meio de fluxos informacionais exige uma abordagem concreta. Uma vez que um fluxo de informação pouco claro ou inconsistente não cumprirá o principal objetivo que é disseminar a informação.

Dentre as alternativas para garantir um bom monitoramento da informação em uma empresa está a implantação de sistemas de gestão, nesse caso, Moraes (2018)

discute a ideia de um sistema de informação gerencial - SIG. São basicamente sistemas com funcionalidades de apoio a decisão para gestores. Os SIGs auxiliam na coleta, processamento, armazenamento, análise e disseminação de informações de forma a atender propósitos específicos do negócio (MORAES, 2018).

Contudo, apesar de encurtar as distâncias entre compradores e vendedores, por outro lado, demanda investimentos contínuos em atualização e manutenção. (BALARINE, 2002, Apud MORAES et al, 2018). Um sistema de qualidade, de qualquer tipo, é extremamente custoso de criar e manter, porém o custo pode ser econômico, já que “O emprego da Tecnologia da Informação (TI) envolve o uso de computadores, tecnologias de comunicações, hardwares e serviços para aprimorar operações” (LAURINDO et al, 2001, Apud MORAES et al, 2018, p. 40), o que demanda um investimento. Além do mais, o custo necessário também pode ser tempo, uma vez que há uma equipe por trás dedicando horas de trabalho para garantir o pleno funcionamento da aplicação.

Com isso, é preciso evitar que problemas no desenvolvimento do sistema prejudiquem os negócios da empresa, já que um sistema que piora a eficiência perde completamente o objetivo. Assim, é primordial o estabelecimento de um sistema de informação integrado, que facilite a inovação e a difusão do conhecimento ao longo da cadeia de suprimentos para uma melhoria da produtividade e qualidade (RIBEIRO et al., 2019).

## **2.4 Monitoramento da infraestrutura de TI**

### **2.4.1 Gestão de Ativos**

Outro ponto importante a ser citado com mais aprofundamento é gestão (ou monitoramento) da infraestrutura de TI. Nesse ponto será dado ênfase à gestão de ativos. Como foi observado no trabalho de Silva e Pinto (2019), a gestão de ativos de rede compreende além do registro das informações dos dispositivos, mas como também outros dados pertinentes ao seu ciclo de vida.

Logo, o ideal para um gestor de TI é ter uma completa noção do que acontece na rede, que pode ser feito utilizando informações oriundas do monitoramento da rede que “permitem ao administrador ter uma visão estratégica do parque de

computadores podendo assim prever possíveis atualizações e manutenções.” (SILVA; PINTO, 2019, p. 59).

Além do mais:

Nesse sentido, a gestão de ativos tem um papel relevante para o administrador de rede. Sendo este processo, portanto, o primeiro a ser dado para a implantação de uma rede organizada, onde todos os dispositivos são registrados e sua utilização é acompanhada desde a aquisição até o descarte, reduzindo desperdícios e maximizando a utilização dos recursos existentes. (Silva; Pinto, 2019, p. 59)

Por fim, é correto afirmar que a gestão de ativos é uma atividade essencial, por isso, é preciso conhecer os recursos de TI e utilizá-los corretamente (SILVA; PINTO, 2019). Desse modo, é possível dizer que ferramentas de apoio à gestão de ativos devem apresentar os pontos de qualidades citados.

#### **2.4.2 Ciclo de vida do ativo**

Dentre os temas mais relevantes do monitoramento de TI está o controle dos ciclos de vida (ou fluxos de informação). Em um ciclo de vida típico, um ativo passa pelas fases de aquisição, operação e fim de vida, com a aquisição envolvendo atividades manuais como a identificação e a inclusão no banco de dados (STONE et al., 2018). Portanto, qualquer transformação pré-estabelecida pela organização detentora do ativo já pode ser considerada como uma atividade do ciclo de vida.

Ciclos de vida podem ser formais, bem documentados, concretos, claros para os colaboradores, o que acaba sendo mais fácil de gerir, porém podem ser também informais, pouco claros e não concretos, dificultando a administração. Ciclos não concretos tendem a piorar a qualidade geral do serviço ou produto feito pela companhia, uma vez que é muito difícil manter um padrão das transformações dos ativos.

Suckow (2011) argumenta que saber identificar a origem de um problema é o primeiro e mais importante passo para solucioná-lo. Nesse caso, a falta de organização da equipe ou controle das atividades por parte da administração podem ser os principais causadores. Logo, Suckow (2011, p. 22) completa, “é fundamental que o gestor de TI invista em métodos que diminuam o tempo de detecção e localização da origem dos problemas.”

Neste ponto, entra um conceito interessante chamado ambientes gerenciáveis (de TI). Basicamente são cenários em que o processo de administração é extremamente facilitado por conta da boa estrutura. Um ambiente de TI gerenciável proporciona a capacidade de dispor ao gestor uma grande quantidade de informações que são de fundamental ajuda para a detecção precoce dos problemas e a origem deles (SUCKOW, 2011).

Com isso, Suckow (2011) complementa ao dizer que em um ambiente de TI gerenciável é possível realizar auditorias nos equipamentos sem se deslocar até eles, concentrando as informações em um único local. Por exemplo, os dados podem estar armazenados em um sistema integrado que possibilita esse controle por parte dos administradores, destacando a importância de softwares informacionais na construção de um ambiente gerenciável.

Entretanto, Suckow (2011, p. 27) evidencia as limitações dos sistemas de gestão, pois “com a grande diversidade de fabricantes e fornecedores de equipamentos fica impossível dispor de uma única ferramenta que gerencie tudo e da maneira que o gestor necessita”. Isso é mais evidente se considerar a realidade de cada empresa, pois cada organização possui regras de negócio, oportunidades e fraquezas distintas, desse modo, um sistema nunca englobará por completo os requisitos de todas as camadas. Todavia, é viável criar uma boa ferramenta que consiga, no mínimo, melhorar a produtividade e a administração dos ativos.

Assim sendo, o investimento em TI torna-se uma maneira de atribuir mais valor aos serviços e produtos de uma empresa, tão como uma forma de superar a concorrência. Dentre os investimentos, o controle dos ativos dos departamentos da empresa faz parte das aquisições fundamentais.

### **3 METODOLOGIA**

Após a revisão teórica, foi possível determinar grande parte das necessidades de um software de gestão de ativos. Sendo assim, foi viável montar o planejamento da metodologia do trabalho. A seção metodologia do projeto abordará o processo de desenvolvimento da aplicação, iniciando ao tratar do levantamento de requisitos, em seguida a prototipação e engenharia do software e por fim os testes no sistema.

#### **3.1 Levantamento de Requisitos**

O levantamento de requisitos é uma das partes fundamentais no desenvolvimento de software. Como afirma Sommerville (2019), requisitos são as descrições das funcionalidades que o sistema deve prestar e suas restrições, o que reflete as necessidades do cliente para atender um determinado propósito. Nesse projeto, o levantamento de requisitos foi feito conforme os dados obtidos durante a pesquisa de fundamentação teórica.

É notório que esse método não é capaz de obter por completo as necessidades para criar um sistema comparável a um software empresarial. No entanto, considerando o objetivo do projeto, o intuito não é englobar regras de negócio de uma empresa e sim criar uma aplicação genérica capaz de otimizar a produtividade. Para tal, o procedimento torna-se o mais adequado considerando a realidade do trabalho.

#### **3.2 Prototipação e Documentação**

Após o levantamento de requisitos, junto ao desenvolvimento do próprio sistema, serão feitas a prototipação e a documentação do projeto. A prototipação será utilizada ocasionalmente com o objetivo de testar opções para o projeto antes de implementá-las por definitivo no sistema. Por outro lado, a documentação do projeto, além de textos, utilizará diagramas UML – *Unified Modeling Language*, que conforme Sommerville (2019) é uma linguagem gráfica que inclui diversos tipos de diagramas de sistema que fornecem visões diferentes do software. Assim, com adaptações nos diagramas para adequá-los à realidade do trabalho, a estrutura do sistema será explicada visualmente.

### **3.3 Testes**

Enquanto o software estiver sendo desenvolvido e após o término da implementação das funcionalidades, serão feitos testes no sistema, que são verificações para confirmar se a aplicação cumpre os requisitos. Por conta da limitação de tempo de desenvolvimento e falta de outros recursos, os testes foram reduzidos para testes simples de caixa preta. Em que o teste é feito na perspectiva do usuário final, em que somente a entrada e a saída de dados são observadas. Logo, são testes rápidos que verificam se uma funcionalidade está correta na visão do usuário.

A escolha de adotar somente testes de caixa preta mostra-se como uma limitação na metodologia do projeto. Por consequência, isso dá oportunidade de aprimorar o trabalho no futuro, em que há mais tempo disponível para dedicar ao software. Dito isso, é correto afirmar que os testes são um ponto de melhoria a ser discutido.

## 4 DESENVOLVIMENTO

### 4.1 Especificação funcional

No desenvolvimento foi construída a especificação funcional do projeto, isso é, a parte da documentação responsável pela descrição das funcionalidades e regras de negócios do sistema. Nas próximas subseções, serão apresentados os conceitos fundamentais e as premissas do software.

#### 4.1.1 Requisitos

Após a análise dos trabalhos na revisão teórica, foi possível determinar que o software, no mínimo, deve ser capaz de:

- Manter<sup>1</sup> ativos de TI básicos (computadores, impressoras, switches etc.)
- Manter equipamentos e componentes dos ativos (mouses, teclados, processadores).
- Manter softwares e licenças dos itens físicos (Sistema Operacional, Servidores de aplicação, Bancos de Dados etc.);
- Manter o cadastro e perfis de acesso dos usuários.
- Manter a configuração de projetos (plantas, instalações etc.) para segregar dados.
- Administrar o ciclo de vida de alguns ativos para controlar as transformações que ocorrem com o objeto enquanto existir.
- Gerar notificações para mudanças no ciclo de vida.

Dado a natureza um tanto acadêmica do projeto, a fim de deixar o sistema e os requisitos menos genérico, algumas regras de negócio foram inseridas para demonstrar a capacidade da aplicação de abstrair e otimizar processos similares aos de uma organização real. As normas e suas características específicas serão discutidas nas próximas seções.

---

<sup>1</sup> Manter é uma palavra utilizada para substituir a junção dos termos buscar, cadastrar, atualizar e deletar. É uma forma de resumir as operações básicas de manipulação de objetos.

## 4.1.2 Regras de negócio

### 4.1.2.1 Ativos

Há seis tipos de ativos que devem ser administrados na aplicação, conforme o quadro 3:

Quadro 3: Descrição dos ativos da aplicação

Ativo	Descrição
<b>Componentes</b>	Principais ativos de TI utilizados em uma corporação, como computadores, switches, pontos de acesso etc.
<b>Equipamentos</b>	Ativos que necessitam de ativos do tipo componente para serem utilizados, como processadores, mouses etc.
<b>Localizações</b>	Localização do ativo dentro da planta industrial.
<b>Manutenções</b>	Manutenções realizadas em componentes.
<b>Requisições</b>	Solicitações realizadas para a equipe de TI relacionadas aos ativos, como compra de novo computador, troca de SSD etc.
<b>Softwares</b>	Sistemas instalados nos ativos componentes, como sistemas operacionais e outros serviços.

Fonte: Desenvolvido pelo autor, 2024

**RN-AT-01<sup>2</sup>:** Qualquer ativo deve estar vinculado a uma das categorias do quadro 3.

**RN-AT-02:** Todo tipo de ativo deve possuir, pelo menos, as propriedades de nome, descrição, atualizado por, data de última atualização, usuário criador e data de criação.

### 4.1.2.2 Projetos

Projetos são o modo utilizado para segregar dados e separar os objetos do sistema. Em outros termos, um ativo contido no projeto 1, não deve aparecer na lista de objetos do projeto 2. Logo, é possível separar os ativos por instalações, assim um funcionário que trabalha em uma instalação não tem acesso aos itens de instalações distintas.

<sup>2</sup> RN é a sigla para regra de negócio. AT refere-se ao tipo de RN, que pode variar, nesse caso significa ativo. A sequência de números é o contador de RNs para aquele tipo.

**RN-PJ-01:** Todo objeto de componente, equipamento, localização, manutenção, requisição ou software deve estar vinculado a um projeto.

**RN-PJ-02:** Um projeto deve possuir nome e descrição.

**RN-PJ-03:** Um objeto existente em um projeto, só pode ser visualizado e atualizado por um usuário configurado com perfil naquele projeto.

#### 4.1.2.3 Componentes

Componentes possuem os seguintes atributos, conforme o quadro 4:

Quadro 4: Atributos dos componentes

<b>Atributo</b>	<b>Descrição</b>	<b>Observação</b>	<b>Obrigatório?</b>
<b>Nome</b>	Identificador do produto.		X
<b>Descrição</b>	Descrição do objeto.		
<b>Número de série</b>	Número para identificar o patrimônio.		X
<b>Status</b>	Representa o estado atual do componente.	Tem os valores de <b>Em Manutenção</b> ou <b>Regular</b>	X
<b>Marca</b>	Empresa responsável pela fabricação do ativo.		X
<b>Modelo</b>	Modelo do produto.		X
<b>Tipo</b>	Tipo de componente.	Tem os valores de <b>Computador, Roteador, Access Point, Impressora, Switch, Servidor, Modem</b> ou <b>Firewall.</b>	X
<b>Data de aquisição</b>	Quando o ativo foi adquirido.		X
<b>Link da nota fiscal</b>	Link para acesso à nota fiscal eletrônica do ativo		X
<b>Atualizado por</b>	Último usuário que atualizou o objeto.		
<b>Data última atualização</b>	Dia e hora em que o objeto foi atualizado.		
<b>Usuário criador</b>	Usuário que criou o objeto.		X

<b>Data criação</b>	Dia e hora em que o objeto foi criado.		X
---------------------	--	--	---

Fonte: Desenvolvido pelo autor, 2024

**RN-CP-01:** Todo componente inicia com o status com valor **Regular**.

**RN-CP-02:** Quando um componente está em manutenção, seu status deve ser alterado para **Em Manutenção**.

**RN-CP-03:** Após a manutenção ser concluída, o status deve retornar para **Regular**.

**RN-CP-04:** O número de série deve ser único para cada componente e equipamento.

**RN-CP-05:** O número de série não pode ser alterado após o cadastro do objeto no sistema.

**RN-CP-06:** Todo componente deve estar relacionado com uma localização.

#### 4.1.2.4 Equipamentos

Equipamentos possuem os seguintes atributos, conforme o quadro 5:

Quadro 5: Atributos dos equipamentos

<b>Atributo</b>	<b>Descrição</b>	<b>Observação</b>	<b>Obrigatório</b>
<b>Nome</b>	Identificador do produto.		X
<b>Descrição</b>	Descrição do objeto.		
<b>Número de série</b>	Número para identificar o patrimônio.		X
<b>Marca</b>	Empresa responsável pela fabricação do ativo.		X
<b>Modelo</b>	Modelo do produto.		X
<b>Tipo</b>	Tipo de componente.	Tem os valores de <b>Memória RAM, Mouse, Teclado, Processador, Disco Rígido, Placa de Vídeo, Fonte de Alimentação, Gabinete, Webcam</b>	X

		ou Headset	
<b>Data de aquisição</b>	Quando o ativo foi adquirido.		X
<b>Link da nota fiscal</b>	Link para acesso à nota fiscal eletrônica do ativo		X
<b>Atualizado por</b>	Último usuário que atualizou o objeto.		
<b>Data última atualização</b>	Dia e hora em que o objeto foi atualizado.		
<b>Usuário criador</b>	Usuário que criou o objeto.		X
<b>Data criação</b>	Dia e hora em que o objeto foi criado.		X

Fonte: Desenvolvido pelo autor, 2024

**RN-EQ-01:** O número de série deve ser único para cada componente e equipamento.

**RN-EQ-02:** Todo equipamento deve estar relacionado com um componente.

#### 4.1.2.5 Localizações

Localizações possuem os seguintes atributos, conforme o quadro 6:

Quadro 6: Atributos de localizações

Atributo	Descrição	Observação	Obrigatório
<b>Nome</b>	Identificador do produto.		X
<b>Descrição</b>	Descrição do objeto.		
<b>Atualizado por</b>	Último usuário que atualizou o objeto.		
<b>Data última atualização</b>	Dia e hora em que o objeto foi atualizado.		
<b>Usuário criador</b>	Usuário que criou o objeto.		X
<b>Data criação</b>	Dia e hora em que o objeto foi criado.		X

Fonte: Desenvolvido pelo autor, 2024

**RN-LC-01:** Qualquer componente deve estar relacionado com pelo menos uma localização.

#### 4.1.2.6 Manutenções

Manutenções possuem os seguintes atributos, conforme o quadro 7:

Quadro 7: Atributos de manutenções

Atributo	Descrição	Observação	Obrigatório
<b>Nome</b>	Identificador do produto.		X
<b>Descrição</b>	Descrição do objeto.		
<b>Data de início</b>	Data de início da manutenção.		X
<b>Data de término</b>	Data de término da manutenção.		X
<b>Solução</b>	Solução apresentada para concluir a manutenção.		X
<b>Atualizado por</b>	Último usuário que atualizou o objeto.		
<b>Data última atualização</b>	Dia e hora em que o objeto foi atualizado.		
<b>Usuário criador</b>	Usuário que criou o objeto.		X
<b>Data criação</b>	Dia e hora em que o objeto foi criado.		X

Fonte: Desenvolvido pelo autor, 2024

**RN-MN-01:** Toda manutenção deve estar vinculada a um componente.

**RN-MN-02:** Uma manutenção deve estar relacionada com um usuário responsável por realizá-la.

**RN-MN-03:** Após criada, a manutenção já deverá iniciar seu ciclo de vida.

**RN-MN-04:** Uma manutenção só pode ser editada enquanto estiver em ciclo de vida e somente pelo usuário responsável.

**RN-MN-05:** A data de início deve ser atribuída assim que a manutenção for criada e não pode ser alterada.

**RN-MN-06:** A data de término deve ser atribuída assim que a manutenção for completada e não pode ser alterada.

#### 4.1.2.7 Requisições

Requisições possuem os seguintes atributos, conforme o quadro 8:

Quadro 8: Atributos de requisições

Atributo	Descrição	Observação	Obrigatório
Nome	Identificador do produto.		X
Descrição	Descrição do objeto.		
Atualizado por	Último usuário que atualizou o objeto.		
Data última atualização	Dia e hora em que o objeto foi atualizado.		
Usuário criador	Usuário que criou o objeto.		X
Data criação	Dia e hora em que o objeto foi criado.		X

Fonte: Desenvolvido pelo autor, 2024

**RN-RQ-01:** Toda requisição deve estar vinculada a um usuário do perfil coordenador, responsável por analisá-la.

**RN-RQ-02:** Após criada, a requisição deve iniciar seu ciclo de vida.

**RN-RQ-03:** Requisições não podem ser editadas.

#### 4.1.2.8 Softwares

Softwares possuem os seguintes atributos, conforme o quadro 9:

Quadro 9: Atributos de softwares

Atributo	Descrição	Observação	Obrigatório
Nome	Identificador do produto.		X
Descrição	Descrição do objeto.		
Chave de licença	Chave para ativar o software		
Data de vencimento	Data de expiração da licença software		
Atualizado por	Último usuário que atualizou o objeto.		
Data última atualização	Dia e hora em que o		

	objeto foi atualizado.		
<b>Usuário criador</b>	Usuário que criou o objeto.		X
<b>Data criação</b>	Dia e hora em que o objeto foi criado.		X

Fonte: Desenvolvido pelo autor, 2024

**RN-SW-01:** Todo software deve estar vinculado a um componente.

#### 4.1.2.9 Usuários

Usuários possuem os seguintes atributos, conforme o quadro 10:

Quadro 10: Atributos de Usuários

<b>Atributo</b>	<b>Descrição</b>	<b>Observação</b>	<b>Obrigatório</b>
<b>Nome</b>	Identificador do produto.		X
<b>Descrição</b>	Descrição do objeto.		
<b>Nome de acesso</b>	Nome de acesso para login no software		X
<b>Senha</b>	Senha de segurança da conta		X
<b>Atualizado por</b>	Último usuário que atualizou o objeto.		
<b>Data última atualização</b>	Dia e hora em que o objeto foi atualizado.		
<b>Usuário criador</b>	Usuário que criou o objeto.		X
<b>Data criação</b>	Dia e hora em que o objeto foi criado.		X

Fonte: Desenvolvido pelo autor, 2024

**RN-US-01:** O nome de acesso deve ser único para usuário.

**RN-US-02:** Usuários não podem editar as próprias informações, somente a senha.

#### 4.1.2.10 Perfis

Os seguintes perfis de acesso devem estar disponíveis na aplicação para os usuários, conforme o quadro 11:

Quadro 11: Perfis de acesso do sistema

<b>Perfil</b>	<b>Descrição</b>
<b>Leitor</b>	Apenas pode consultar informações.
<b>Suporte</b>	Não consegue excluir objetos e acessar funções administrativas.
<b>Coordenador</b>	Não consegue excluir objetos e acessar funções administrativas. Porém pode analisar requisições.
<b>Administrador</b>	Pode realizar todas as ações relacionadas a negócios, inclusive as administrativas.
<b>Oráculo</b>	Conta crítica. Consegue fazer tudo.

Fonte: Desenvolvido pelo autor, 2024

As permissões de cada perfil estão listadas com mais detalhes na Figura 1:

Figura 1: Acessos dos perfis

Seção	Funcionalidade	Leitor	Suporte	Coordenador	Administrador	Oráculo
Componente	Buscar	Sim	Sim	Sim	Sim	Sim
	Criar	Não	Sim	Sim	Sim	Sim
	Atualizar	Não	Sim	Sim	Sim	Sim
	Excluir	Não	Não	Não	Sim	Sim
Equipamento	Buscar	Sim	Sim	Sim	Sim	Sim
	Criar	Não	Sim	Sim	Sim	Sim
	Atualizar	Não	Sim	Sim	Sim	Sim
	Excluir	Não	Não	Não	Sim	Sim
Localizações	Buscar	Sim	Sim	Sim	Sim	Sim
	Criar	Não	Sim	Sim	Sim	Sim
	Atualizar	Não	Sim	Sim	Sim	Sim
	Excluir	Não	Não	Não	Sim	Sim
Requisições	Buscar	Sim	Sim	Sim	Sim	Sim
	Criar	Não	Sim	Sim	Sim	Sim
	Atualizar	Não	Sim	Sim	Sim	Sim
	Excluir	Não	Não	Não	Sim	Sim
Softwares	Buscar	Sim	Sim	Sim	Sim	Sim
	Criar	Não	Sim	Sim	Sim	Sim
	Atualizar	Não	Sim	Sim	Sim	Sim
	Excluir	Não	Não	Não	Sim	Sim
Atribuições	Visualizar	Não	Sim	Sim	Sim	Sim
Administração	Buscar usuários	Não	Não	Não	Sim	Sim
	Criar usuários	Não	Não	Não	Sim	Sim
	Atualizar usuários	Não	Não	Não	Sim	Sim
	Alterar perfis de usuário	Não	Não	Não	Sim	Sim
	Excluir usuários	Não	Não	Não	Sim	Sim
	Buscar projetos	Não	Não	Não	Não	Sim
	Criar projetos	Não	Não	Não	Não	Sim
	Atualizar projetos	Não	Não	Não	Não	Sim
Excluir projetos	Não	Não	Não	Não	Sim	

Fonte: Desenvolvido pelo autor, 2024

**RN-PR-01:** Os perfis de um usuário devem estar vinculados aos projetos. Logo, um usuário pode ter diferentes permissões dependendo do projeto selecionado.

**RN-PR-02:** Um usuário pode ter mais de um perfil por projeto.

**RN-PR-03:** Deve existir um usuário padrão do sistema chamado **Oráculo** que contenha o perfil oráculo para um projeto base, que é também padrão.

**RN-PR-04:** Apenas o usuário oráculo pode atribuir o perfil oráculo a outros usuários.

#### 4.1.2.11 Atribuições

As atribuições são tarefas geradas aos usuários participantes de ciclos de vida que contém as ações de intervenção possíveis para determinado passo.

**RN-AT-01:** Uma atribuição só pode ser designada para um usuário por vez.

**RN-AT-02:** Um usuário só pode atuar em uma atribuição caso tenha acesso ao projeto do objeto que está sendo analisado no ciclo.

**RN-AT-03:** Uma atribuição deve possuir um nome para identificá-la e uma descrição do que deve ser feito pelo usuário.

**RN-AT-04:** As atribuições devem possuir prazos para serem concluídas, conforme as regras dos ciclos de vida.

#### 4.1.3 Ciclos de vida

Os ciclos de vida são compostos por uma sequência de passos organizados de maneira lógica. Os passos podem necessitar da ação, gerada através de uma atribuição, de um usuário para serem concluídos. A ação sempre possui um prazo para ser completada, do qual considera apenas dias úteis (segunda a sexta-feira desconsiderando feriados). Inclusive, a ação feita pelo usuário pode alterar o curso do ciclo de vida, o que pode gerar variações na sequência.

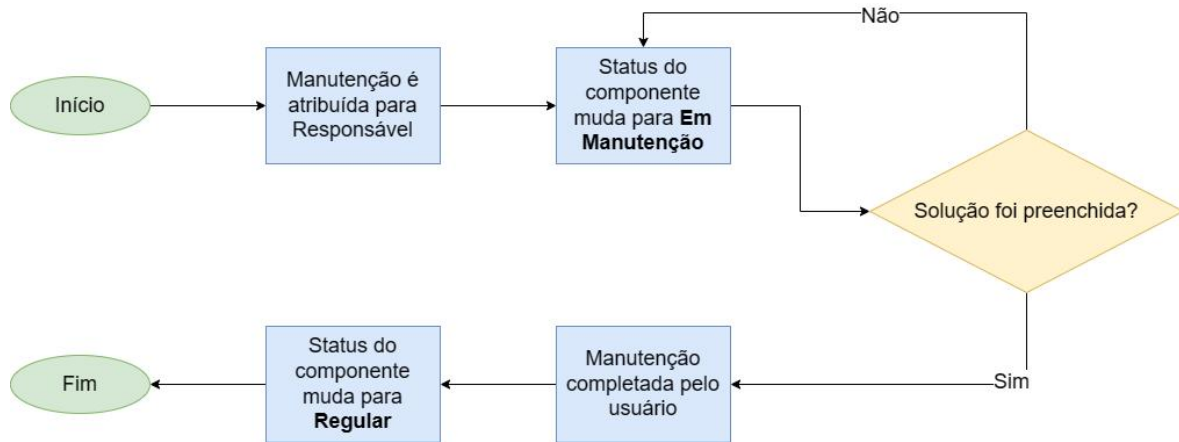
Para demonstrar o funcionamento dos ciclos de vida, serão utilizados diagramas de negócio, que são representações gráficas feitas para revelar os caminhos de execução lógica de processos de negócio (YUZRAAN, 2021). Assim, é possível mostrar os caminhos e as sequências de passos existentes de modo visual. Dado isso, os ciclos de vida do software serão descritos e apresentados nas subseções seguintes.

##### 4.1.3.1 Verificação de manutenção

O ciclo inicia com a manutenção sendo atribuída para o usuário responsável, que a recebe em uma atribuição na caixa de entrada. Em seguida, o status do componente mudará para **Em Manutenção**. O usuário só poderá concluir a manutenção assim que o campo de solução for concluído, nesse caso, o status do componente voltará a ser **Regular** e o ciclo encerra.

A sequência de passos pode ser visualizada na Figura 2:

Figura 2: Diagrama de negócios de verificação de manutenção



Fonte: Desenvolvido pelo autor, 2024

**RN-CVM-01:** O responsável possui o prazo de 3 dias úteis para completar a manutenção.

**RN-CVM-02:** Uma notificação de aviso deve ser gerada para o usuário criador da manutenção após o responsável completar a manutenção.

**RN-CVM-03:** A notificação de aviso não deve ser gerada caso o próprio usuário criador seja o responsável pela manutenção.

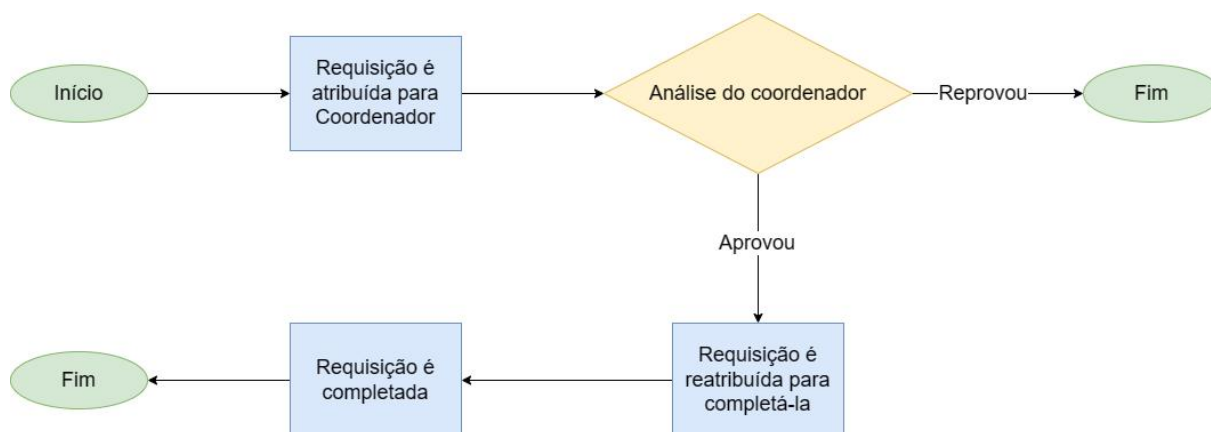
**RN-CVM-04:** A manutenção só poderá ser completada caso o campo de solução estiver preenchido.

#### 4.1.3.2 Análise de requisição

Uma requisição é criada e o coordenador a recebe como uma atribuição. Ele deve aprovar ou reprová-la. Se reprová-la, o ciclo termina, caso contrário o coordenador será responsável por completar a requisição. Após isso, o usuário deverá marcar a requisição como **completada** para encerrar a sequência.

Os passos podem ser visualizados na Figura 3:

Figura 3: Diagrama de negócios de análise de requisição



Fonte: Desenvolvido pelo autor, 2024

**RN-CAR-01:** O coordenador possui o prazo de 3 dias úteis para aprovar ou rejeitar a requisição e 5 dias úteis para completá-la.

**RN-CAR-02:** Uma notificação deve ser gerada para o usuário criador da requisição após o coordenador aprovar ou rejeitar a requisição.

**RN-CAR-03:** Outra notificação será enviada ao usuário após o coordenador.

#### 4.1.4 Diagramas de caso de uso

Para representar as funcionalidades do sistema em alto nível, serão utilizados diagramas de caso de uso. Podem ser definidos como:

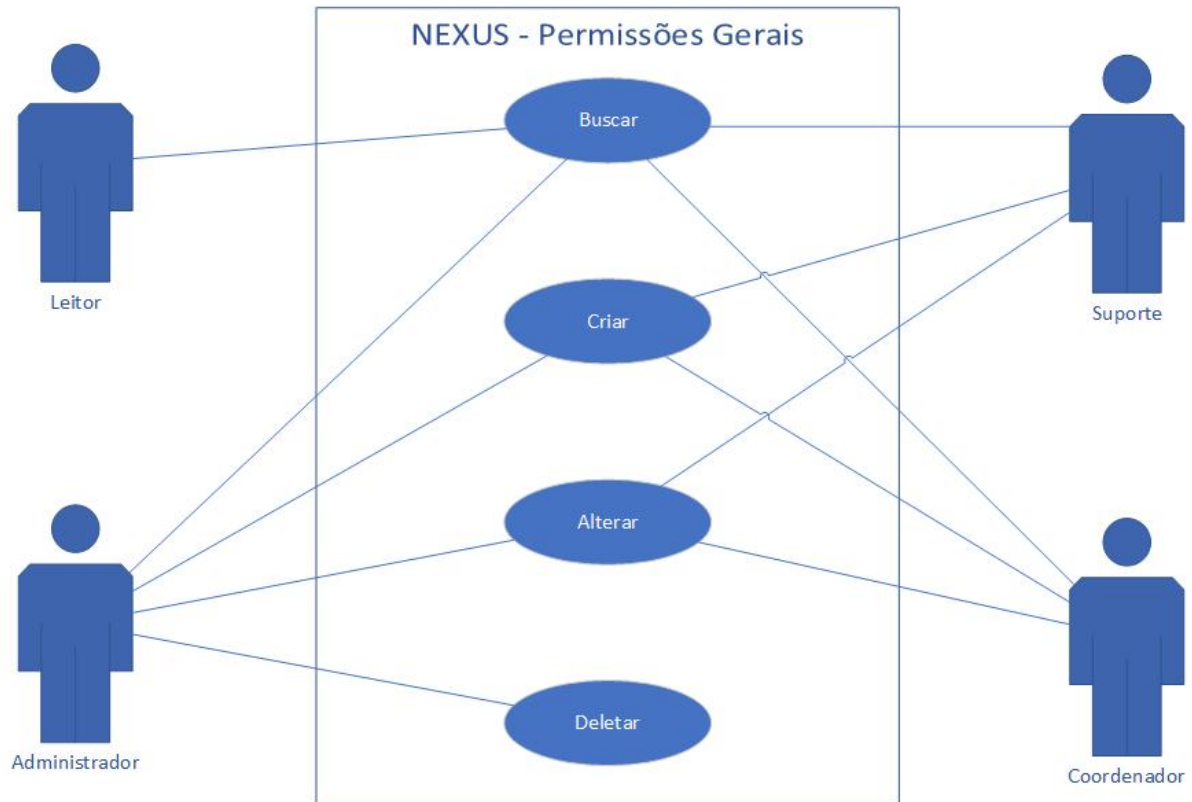
O conjunto de casos de uso representa todas as interações possíveis que serão descritas nos requisitos de sistema. Os atores no processo, que podem ser seres humanos ou outros sistemas, são representados como 'bonecos palito'. Cada classe de interação é representada como uma elipse nomeada. Linhas fazem a ligação entre os atores e a interação. Opcionalmente, pontas de seta podem ser acrescentadas às linhas para mostrar como a interação começa. (SOMMERVILLE, 2019, p.108)

Dado isso, as seguintes subseções apresentarão uma sequência de casos de uso para demonstrar as características fundamentais do sistema.

##### 4.1.4.1 Permissões gerais

Diagrama de casos de uso das permissões gerais da manipulação de todos os tipos de ativos pode ser observado na Figura 4:

Figura 4: Casos de uso de permissões gerais



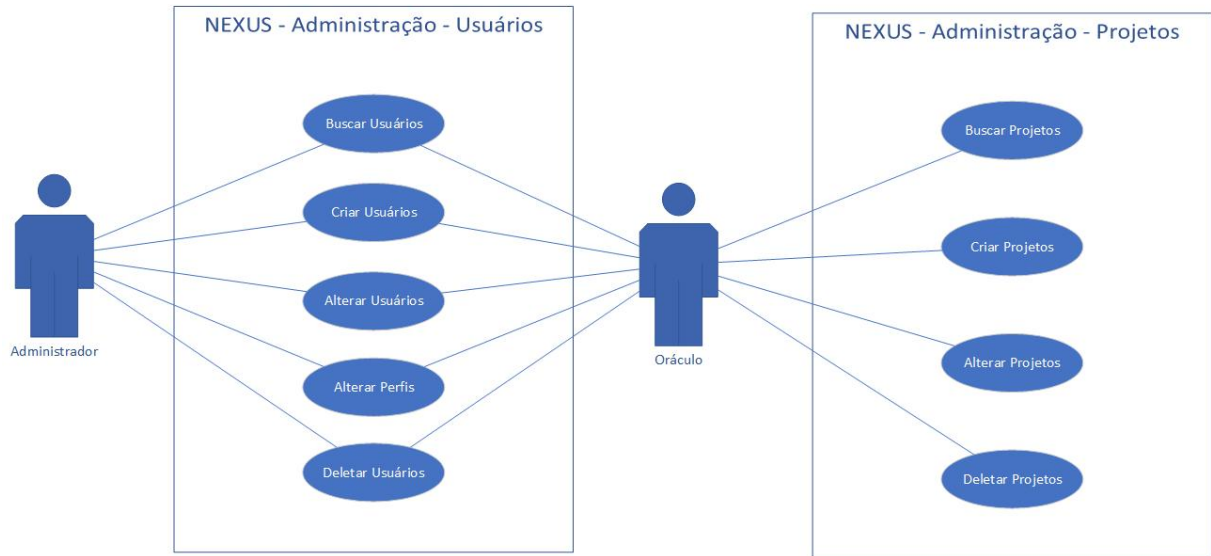
Fonte: Desenvolvido pelo autor, 2024

O perfil de oráculo foi desconsiderado neste diagrama pois não se trata de um grupo de acesso de negócios, e sim um perfil administrativo da aplicação.

#### 4.1.4.2 Administração

Diagrama de casos de uso das funções administrativas na Figura 5:

Figura 5: Casos de uso das funções administrativas

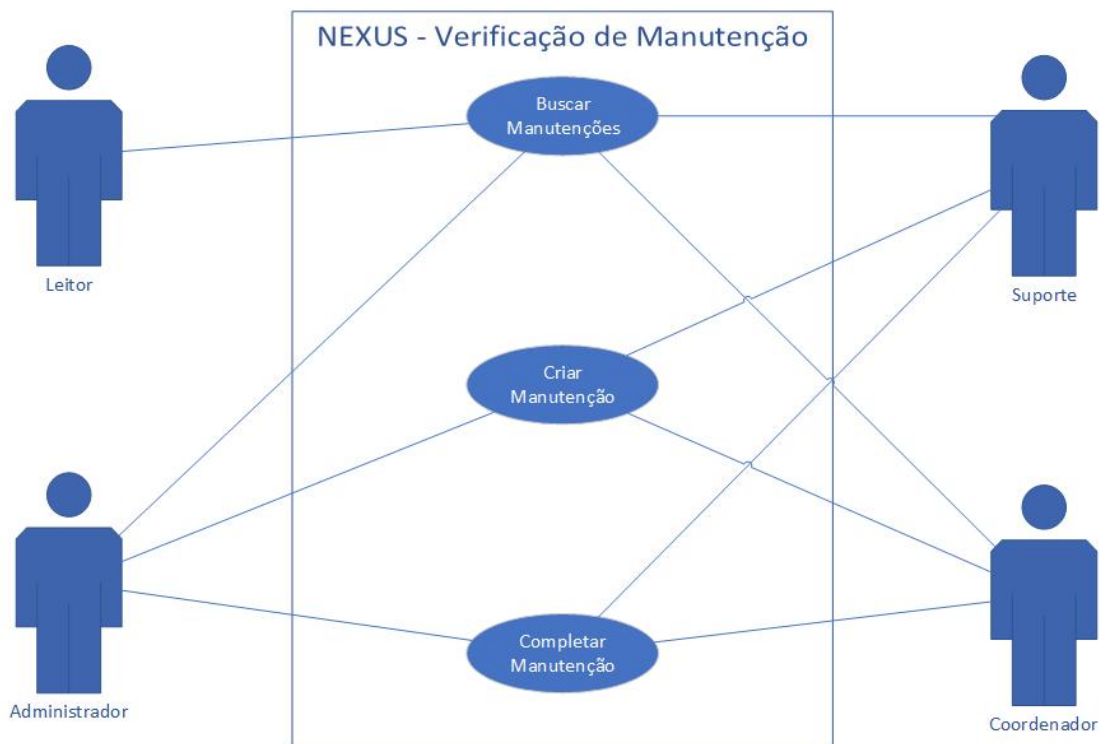


Fonte: Desenvolvido pelo autor, 2024

#### 4.1.4.3 Verificação de Manutenção

Na Figura 6, é possível observar o diagrama de casos de uso do ciclo de vida de verificação de manutenção:

Figura 6: Casos de uso de verificação de manutenção

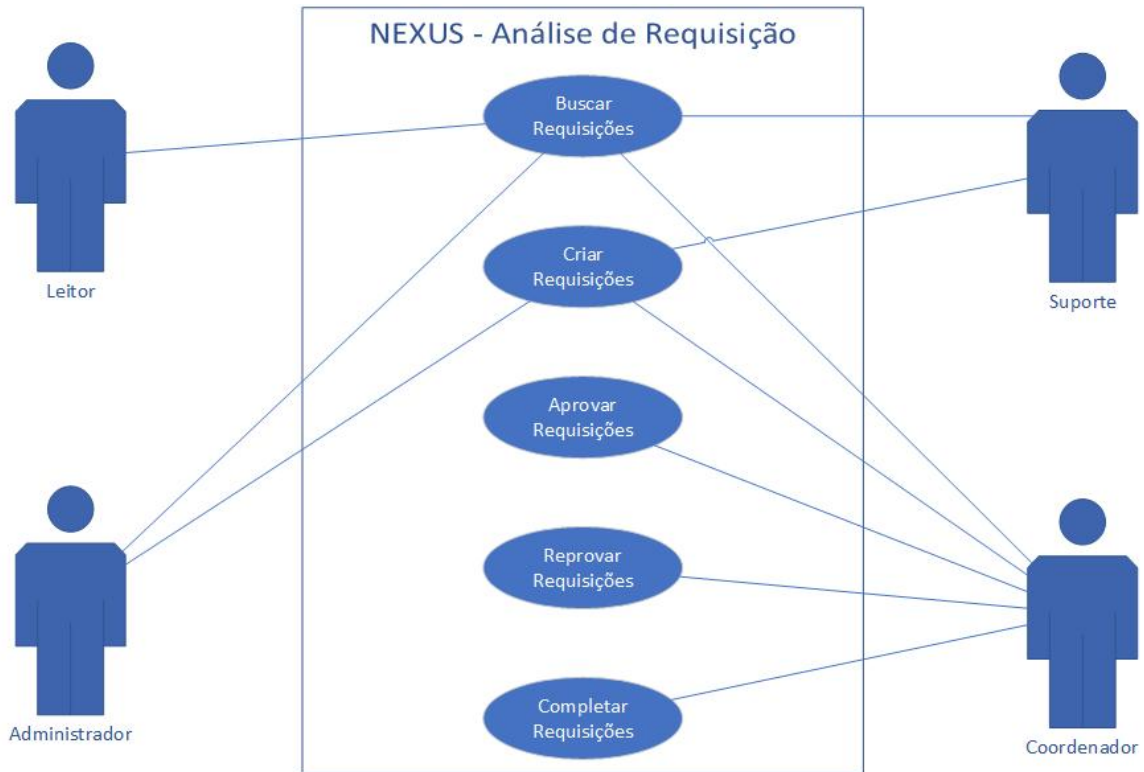


Fonte: Desenvolvido pelo autor, 2024

#### 4.1.4.4 Análise de Requisição

Diagrama de casos de uso do ciclo de vida de análise de requisição pode ser observado na Figura 7:

Figura 7: Casos de uso de análise de requisição



Fonte: Desenvolvido pelo autor, 2024

## 4.2 Especificação técnica

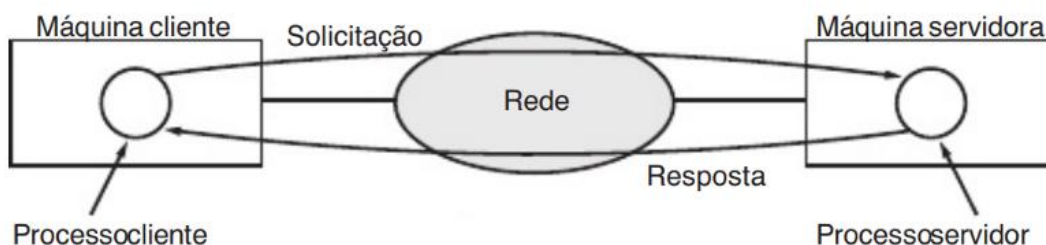
Assim como a especificação funcional, foi feita a especificação técnica, que é a parte da documentação responsável por explicar os aspectos técnicos de implementação do software. Nas próximas subseções, será apresentado a estrutura técnica do projeto.

### 4.2.1 Arquitetura

O software é um sistema web, logo, o sistema funciona com base no conceito de **cliente-servidor**. Basicamente, há um cliente (normalmente um navegador web do usuário final), responsável por fazer requisições a um servidor. O cliente aguarda por uma resposta e o servidor, ao receber a solicitação, executa o trabalho solicitado ou procura pelos dados e envia a resposta de volta (TANEMBAUM; WETHERALL,

2011 Apud PARENTE, 2021). Os detalhes desse conceito podem ser observados na Figura 8:

Figura 8: Cliente-Servidor



Fonte: PARENTE, 2021

Neste sentido, para interagir diretamente com o navegador do usuário há uma aplicação Front-end, feita com o uso da ferramenta Angular, responsável por exibir a interface de usuário, controlar as requisições do cliente e solicitar e retornar as respostas do servidor. Por outro lado, para administrar o servidor, existe uma aplicação Back-end, criada com a ferramenta .NET, feita para gerir as regras de negócio, atualizar o banco de dados e enviar os dados corretos para cada situação. Mais detalhes sobre as tecnologias utilizadas serão fornecidos nas próximas seções.

#### 4.2.1.1 Módulos

Para facilitar o desenvolvimento, o Back-end do sistema foi projetado na arquitetura monolítica, isso é, o programa em si do servidor funciona isoladamente em um único computador (SOMMERVILLE, 2019). Contudo, apesar de ser apenas uma solução, o código é dividido em módulos em que cada um responsável por administrar um conjunto de entidades. Os detalhes de cada módulo podem ser vistos no quadro 12:

Quadro 12: Descrição dos módulos do sistema

Módulos	Descrição	Entidades
Administração	Gere entidades que existem por fins administrativos	Notificacao Perfil Projeto Usuário UsuárioPerfil

<b>CicloVidaAtivo</b>	Manipula os ciclos de vida	Atribuicao
<b>Dados</b>	Administra os ativos	Componente Equipamento Localização Manutenção Requisição Software
<b>Compartilhado</b>	Possui entidades úteis aos três outros módulos	NexusCicloVidaController NexusCicloVidaService NexusEnumDTO NexusEnvioDTO NexusListaRespostaDTO NexusRespostaDTO NexusController NexusRepository NexusService NexusManipulacaoEnum NexusObjeto NexusReferenciaObjeto NexusRelacionamento

Fonte: Desenvolvido pelo autor, 2024

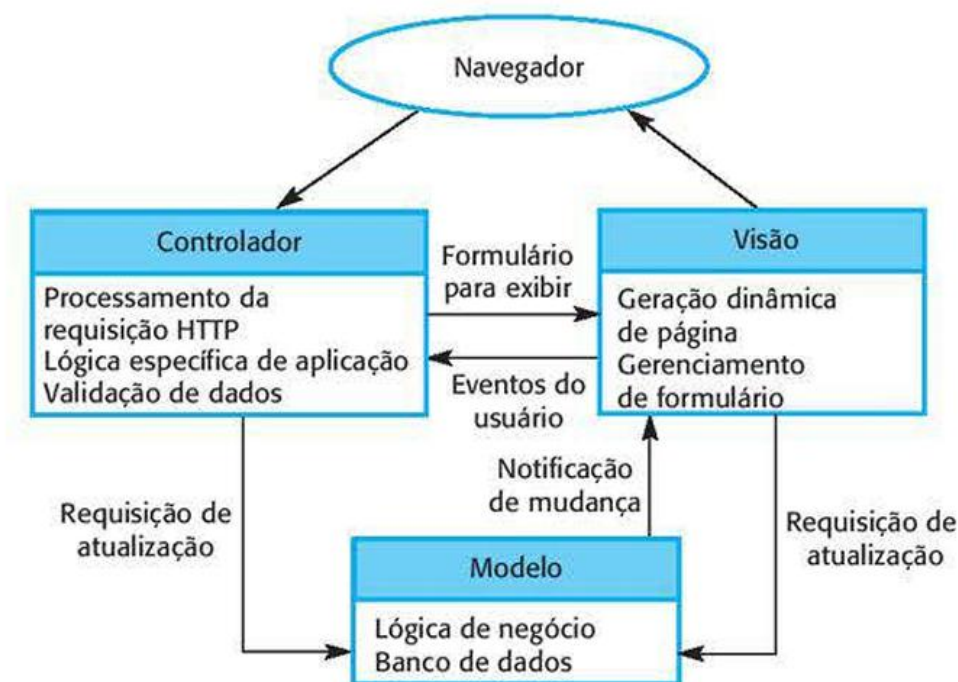
#### 4.2.1.2 MVC

Para organizar os módulos, todos seguem a arquitetura MVC (*Model-View-Controller*), isso é, segundo Sommerville (2019), há três componentes principais em cada model:

- **Model:** Entidade responsável por encapsular os dados enviados pela *View* e o *Controller* e as operações relacionadas a esses dados.
- **View:** Interface de usuário responsável por exibir os dados do *Model* enviados pelo *Controller*.
- **Controller:** Processa as requisições do cliente e retorna um *Model* apropriado para cada solicitação.

É possível entender visualmente a arquitetura MVC na Figura 9:

Figura 9: Arquitetura MVC



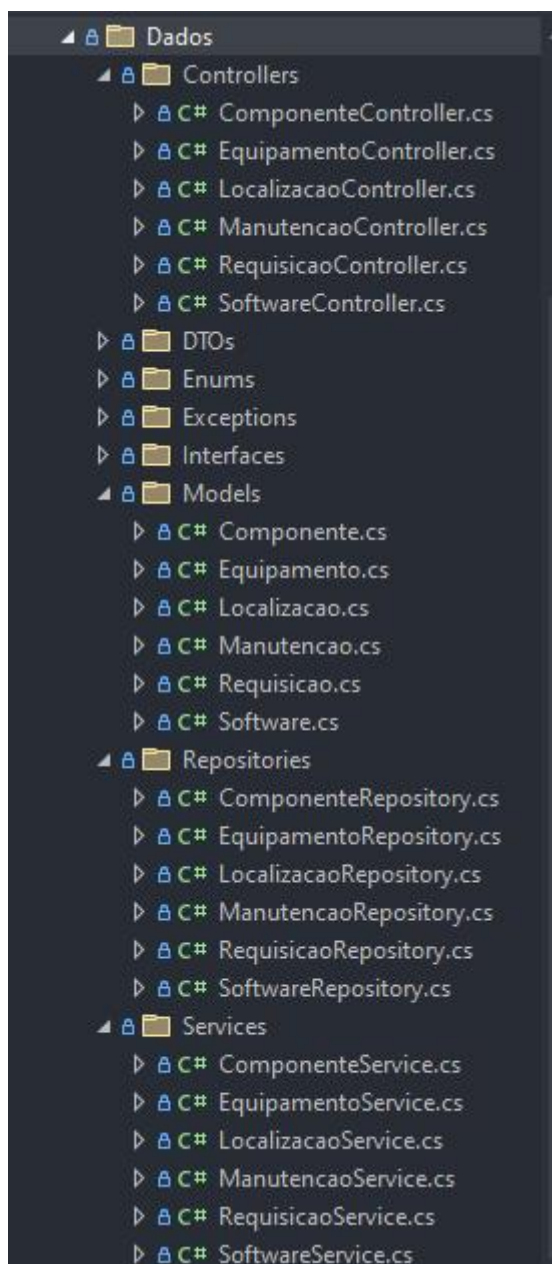
Fonte: SOMMERVILLE, 2019

Para o contexto do NEXUS, foi feita uma adaptação para o modelo MVC, para facilitar o desenvolvimento. Assim, para cada entidade, haverá 4 componentes:

- **Model:** Contém todas as propriedades da entidade e encapsula os dados para facilitar o envio entre cada componente.
- **Repository:** Realiza as operações no banco de dados respectivo a entidade.
- **Service:** Contém as operações relacionadas a manipulação dos dados da entidade e as regras de negócio do sistema.
- **Controller:** Administra as requisições para a entidade.

Com isso, os módulos do projeto ficam com uma estrutura similar a Figura 10:

Figura 10: Exemplo de estrutura de módulo



Fonte: Desenvolvido pelo autor, 2024

Não existem *Views* nos módulos do servidor, pois a *View* é separada da aplicação Back-end. Logo, esse componente existe apenas no *Front-end*.

#### 4.2.2 Diagramas de classe

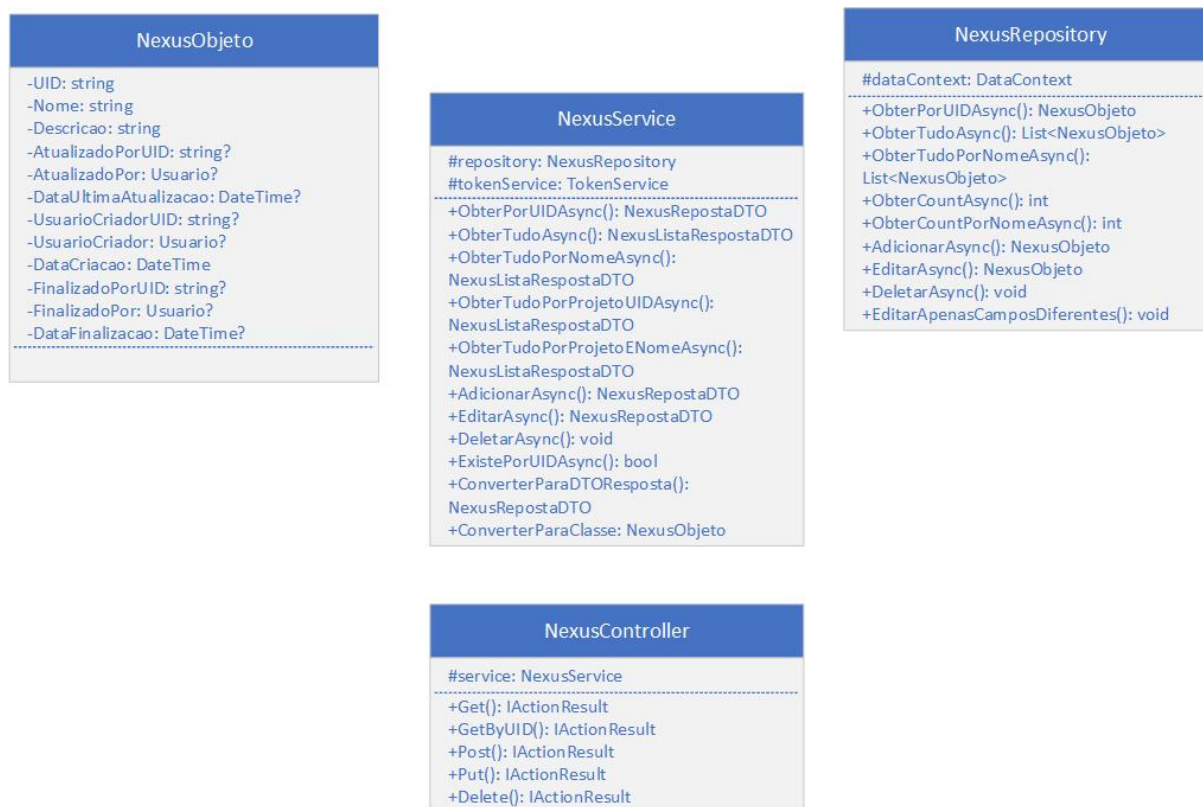
O sistema é composto por diversas classes. Uma classe é a representação de uma entidade, porém dentro do software. As classes são compostas por características (propriedades) e comportamentos (ações). Em outros termos, uma

classe fictícia de cachorro teria como propriedades o nome, raça e cor, e como ações latir, pular e comer.

Assim, utilizam-se os diagramas de classe para representá-las. Para tanto, são um conjunto de modelos visuais contendo as classes, suas respectivas propriedades e ações, e os relacionamentos entre essas classes (SOMMERVILLE, 2019). Dito isso, as principais classes do sistema serão apresentadas nas próximas subseções:

O sistema possui algumas classes bases feitas para reutilizar código por meio da herança, ou seja, uma classe herda as características e métodos da classe base, sem a necessidade de replicar o código. Isso significa que as informações comuns serão mantidas em um único lugar e não é preciso ver todas as classes no sistema para ver se são afetadas por mudanças (SOMMERVILLE, 2019). As classes bases podem ser observadas na Figura 11 abaixo:

Figura 11: Diagrama das classes bases



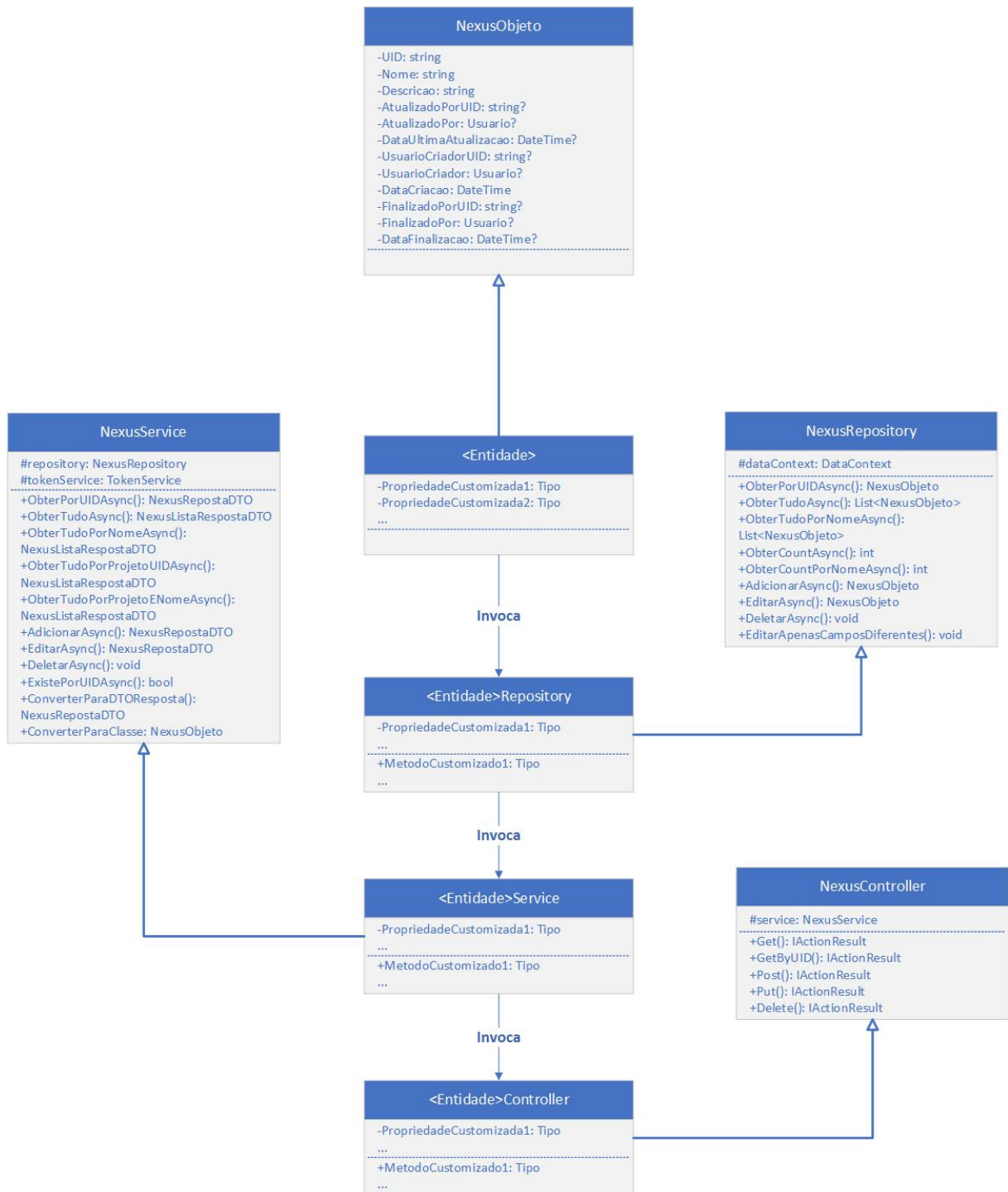
Fonte: Desenvolvido pelo autor, 2024

As classes presentes são:

- **NexusObjeto:** É a base para as classes do tipo *Model*. Possui todas as propriedades básicas utilizadas por todas os ativos rastreados pela aplicação.
- **NexusRepository:** Possui os métodos básicos de buscar, adicionar, editar e deletar no banco de dados já implementados para serem utilizados pelas classes de *Repository*.
- **NexusService:** É base dos classes do tipo *Service*, pois contém os métodos básicos de manipulação de regras de negócio do sistema.
- **NexusController:** Tem os métodos básicos de buscar, adicionar, editar e deletar para se conectar e responder as requisições da aplicação front-end.

A estrutura básica dos módulos de dados, ciclo de vida e administração pode ser descrita na Figura 12:

Figura 12: Diagrama de classe da estrutura básica



Fonte: Desenvolvido pelo autor, 2024

Para quase todas as entidades do projeto, a estrutura e relacionamento entre as classes são iguais. As classes bases são utilizadas para formar um padrão de propriedades e métodos entre os quatro componentes de manipulação de entidades. Inclusive, caso seja necessário criar propriedades e métodos customizados para a

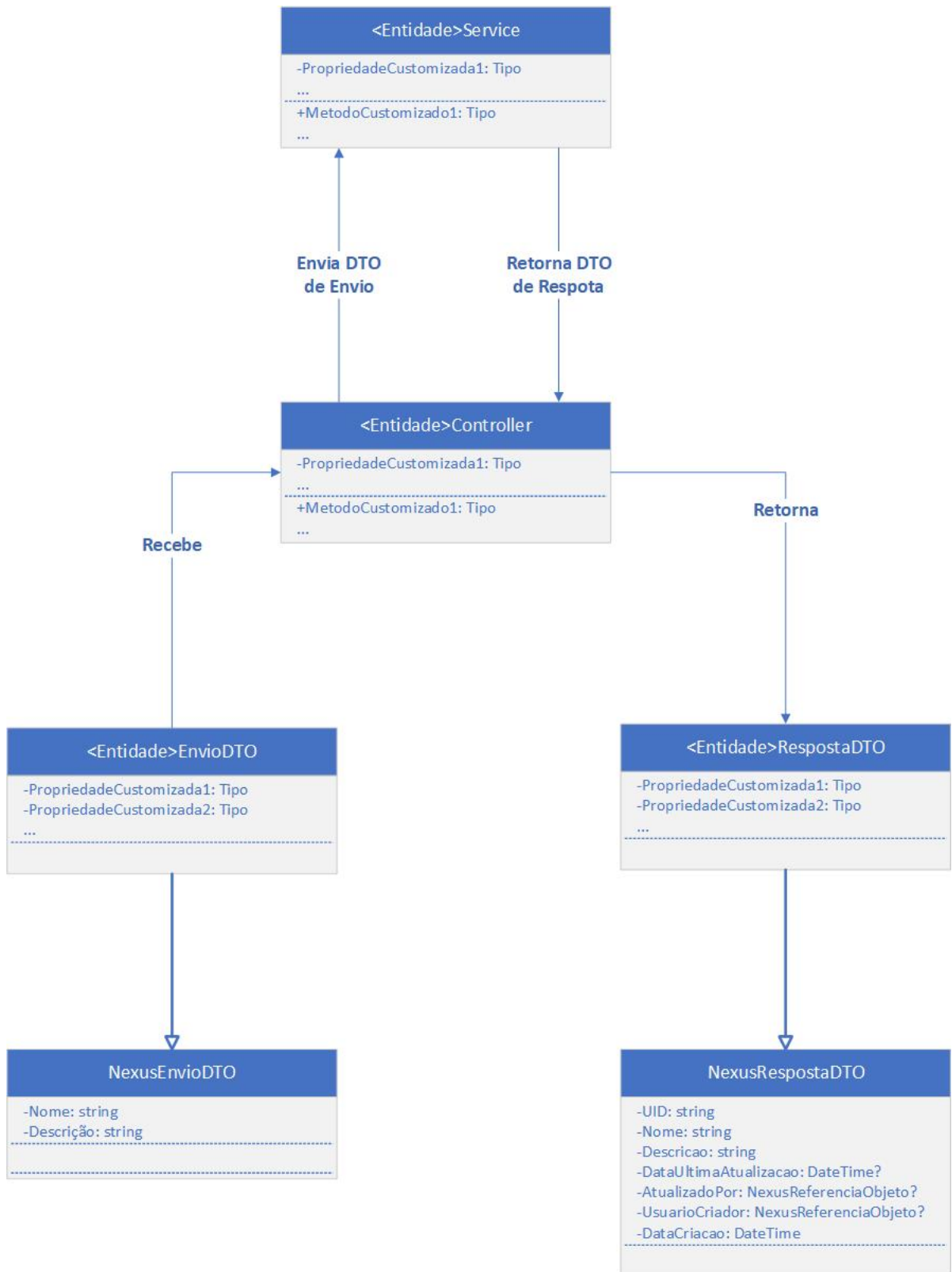
entidade específica, pois as características básicas não são o suficiente, basta adicioná-los à classe derivada.

#### **4.2.2.1 *Data Transfer Object* - DTOs**

No sistema, há classes de apoio para cada entidade chamadas de DTOs (*Data Transfer Object*). Elas são utilizadas para transferir dados entre classes e aplicações distintas e evitar que informações desnecessárias sejam enviadas em requisições, em vez de expor classes de *Model* que poderiam conter informações sensíveis.

No projeto, as DTOs foram utilizadas para transferir dados do back-end para o front-end. Assim, as classes derivadas de *Service* são responsáveis pela manipulação e conversão de *Model* para DTO e vice-versa. Com isso, é possível observar a seguinte estrutura para as DTOs, conforme a Figura 13:

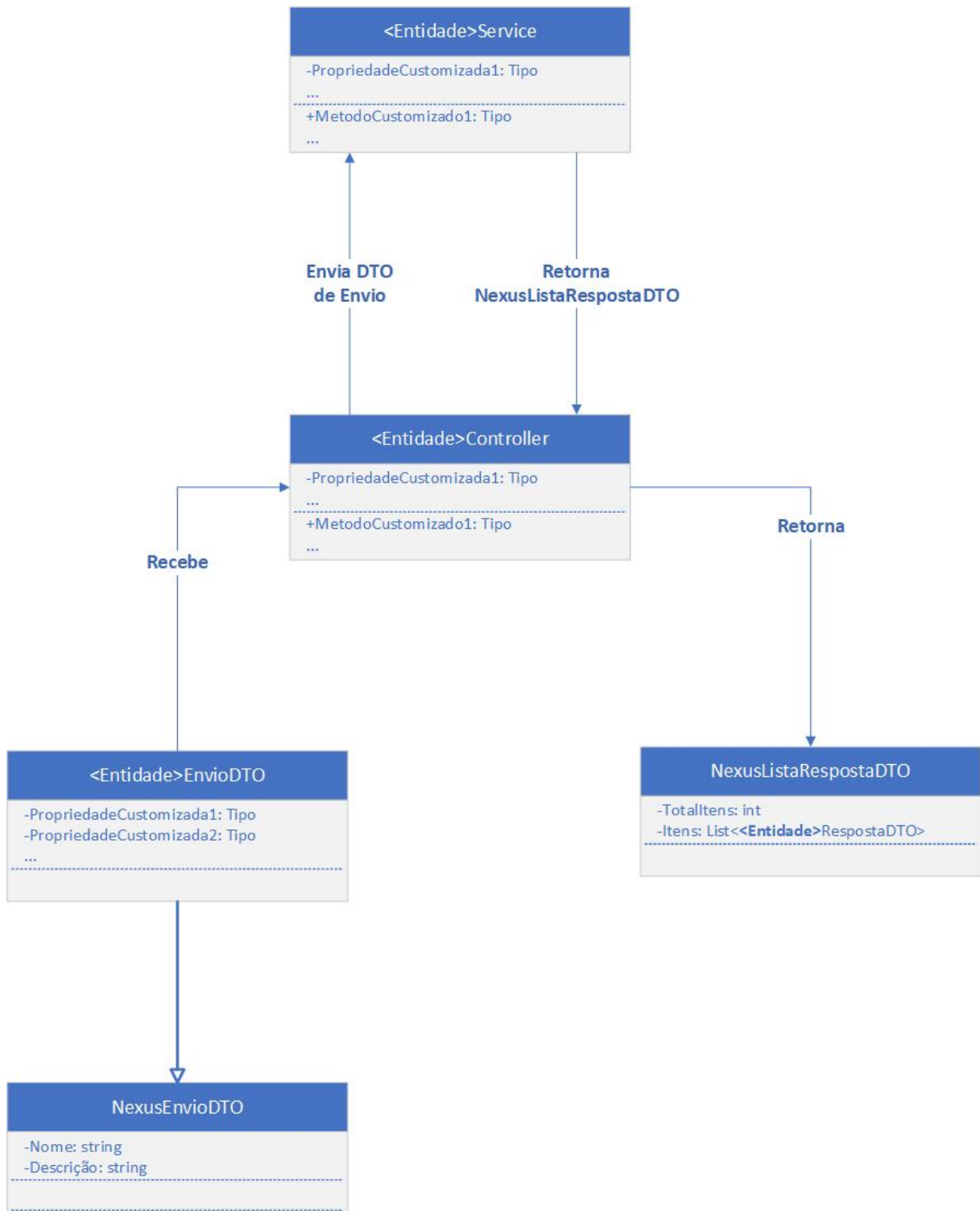
Figura 13: Diagrama de classes de DTOs



Fonte: Desenvolvido pelo autor, 2024

Assim, cada entidade possui suas respectivas DTOs de envio e resposta, que herdam as classes bases de **NexusEnvioDTO** e **NexusRespostaDTO**, responsáveis por respectivamente receber os dados enviados pelo front-end e retornar as informações pertinentes nas requisições. Contudo, o diagrama da última Figura apenas se aplica em casos de retorno de um único objeto. Para situações em que é necessário retornar lista de objetos, há a classe **NexusListaRespostaDTO**, que segue a estrutura descrita na Figura 14 abaixo:

Figura 14: Diagrama de NexusListaRespostaDTO



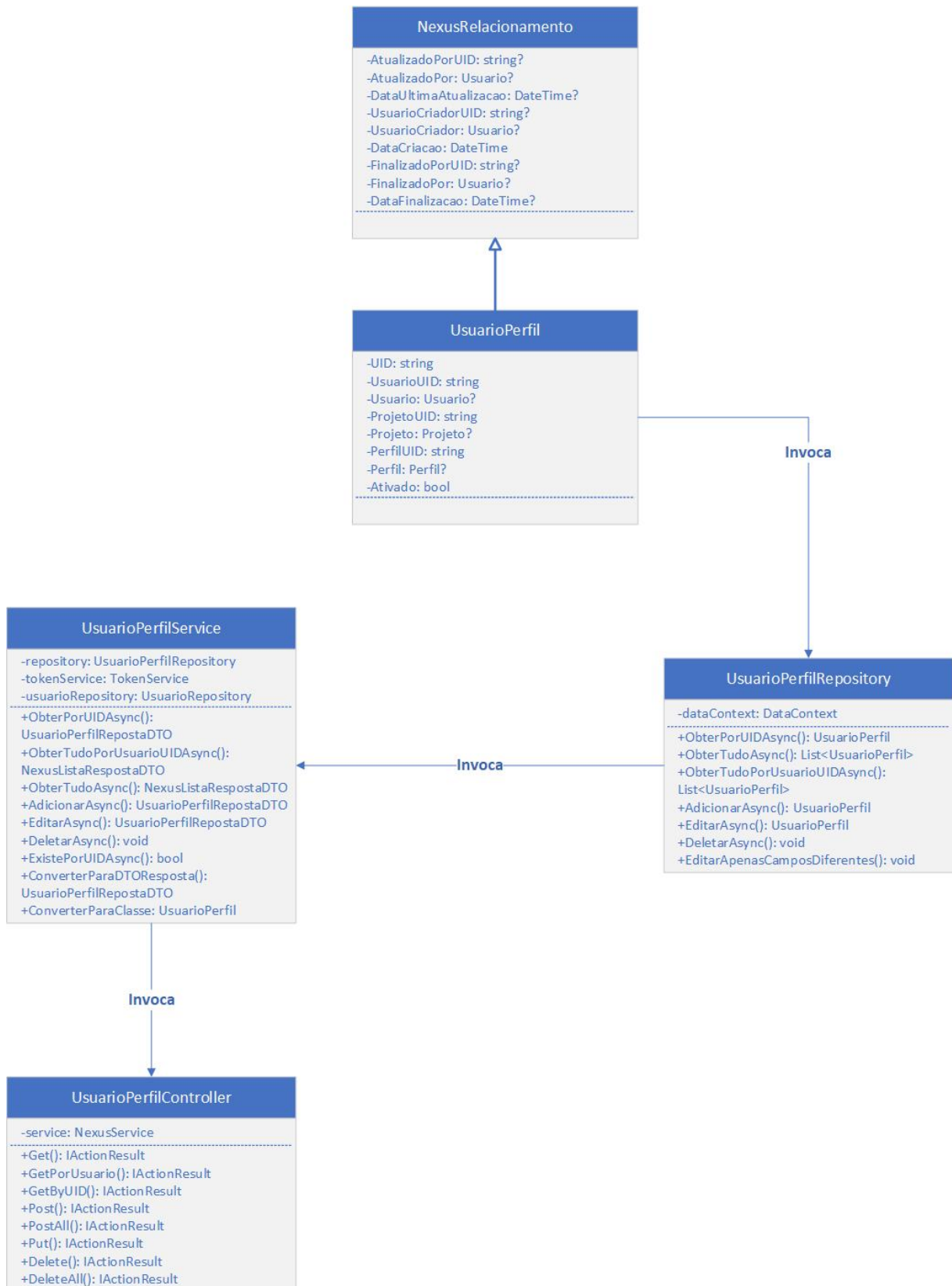
Fonte: Desenvolvido pelo autor, 2024

Logo, **NexusListaRespostaDTO** é uma classe composta pelo total de itens de uma determinada entidade e a lista de itens pesquisados, em formato de DTO de resposta da entidade.

#### 4.2.2.2 **UsuarioPerfil**

A única entidade da aplicação que segue uma estrutura diferente das demais é **UsuarioPerfil**. Isso porque a classe foi criada com o intuito de normalizar o relacionamento entre usuário, projeto e perfil, em que um usuário. Desse modo, seus atributos acabam sendo distintos dos demais, seguindo a estrutura descrita na Figura 15:

Figura 15: Diagrama de UsuarioPerfil



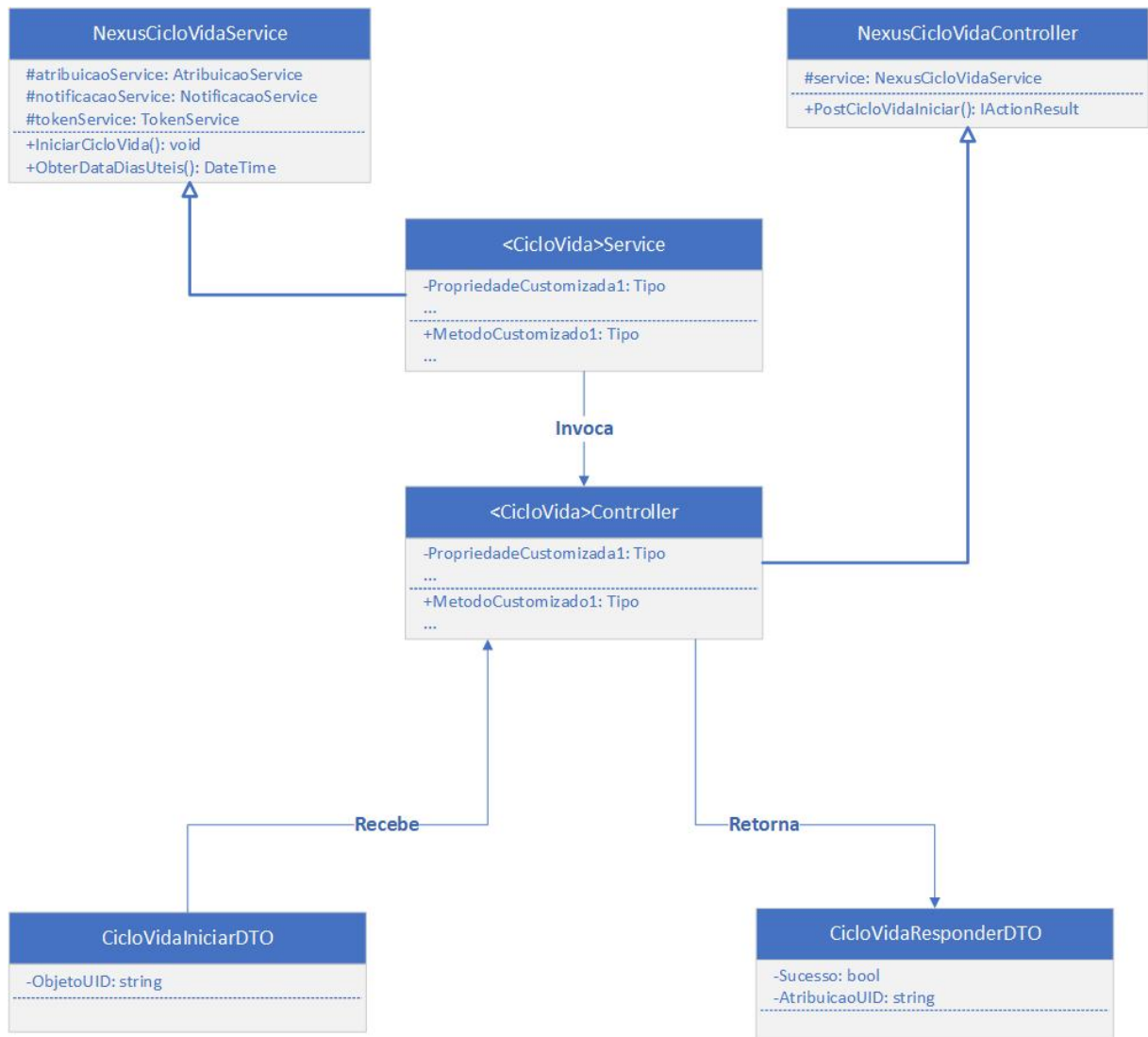
Fonte: Desenvolvido pelo autor, 2024

Assim, usuário perfil não herda nenhuma classe base, apenas utiliza a classe **NexusRelacionamento** para formar seu modelo.

#### 4.2.2.3 Ciclos de vida

Os ciclos de vida, por sua vez, possuem uma estrutura diferente das entidades comuns. Para manipular os ciclos de vida, são feitas requisições a um *Controller* específico do ciclo, responsável por invocar a classe de *Service* criada para executar as ações dos passos do ciclo de vida. A estrutura é descrita na Figura 16:

Figura 16: Diagrama de ciclos de vida



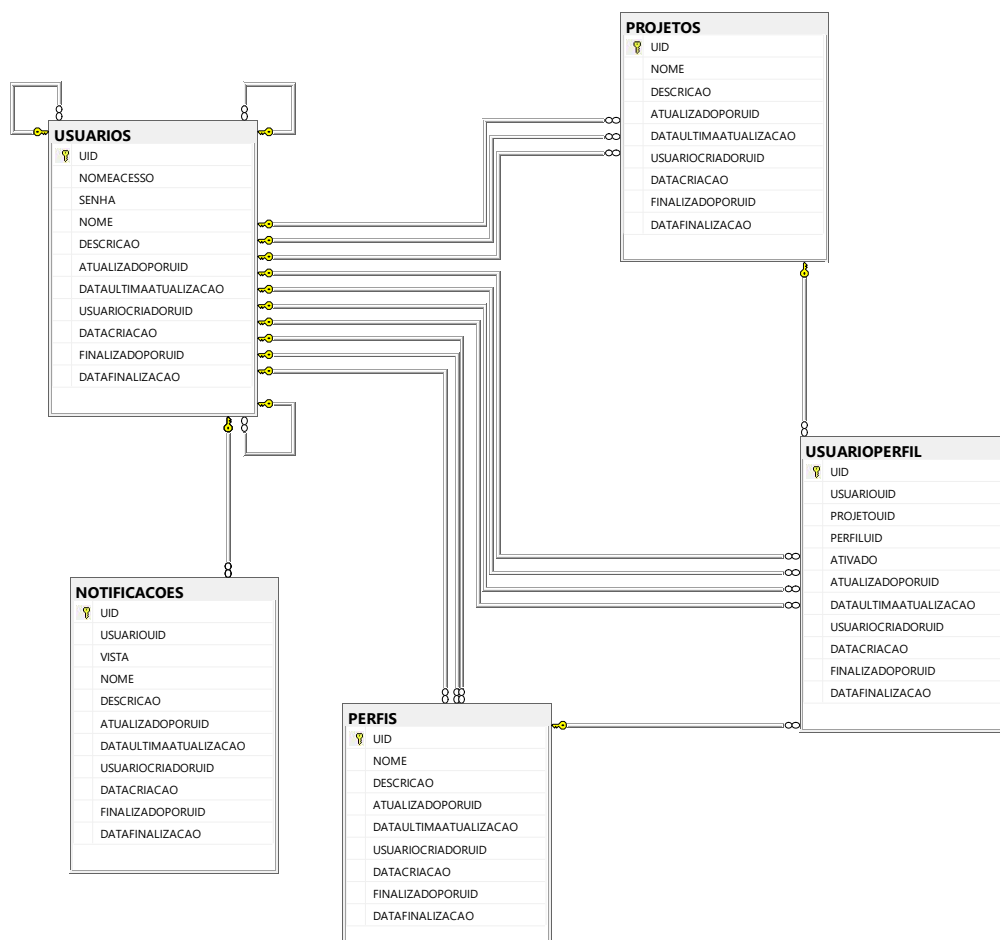
Fonte: Desenvolvido pelo autor, 2024

Sendo assim, cada ciclo de vida possui um *Service* e um *Controller* próprio, que herdam respectivamente a classe **NexusCicloVidaService** e **NexusCicloVidaController**, porém as DTOs que se comunicam com a aplicação front-end são iguais para todos. Desse modo, é possível manter um padrão de comportamento, enquanto há código customizado para se adequar as regras de negócio para cada ciclo de vida.

### 4.2.3 Arquitetura do banco de dados

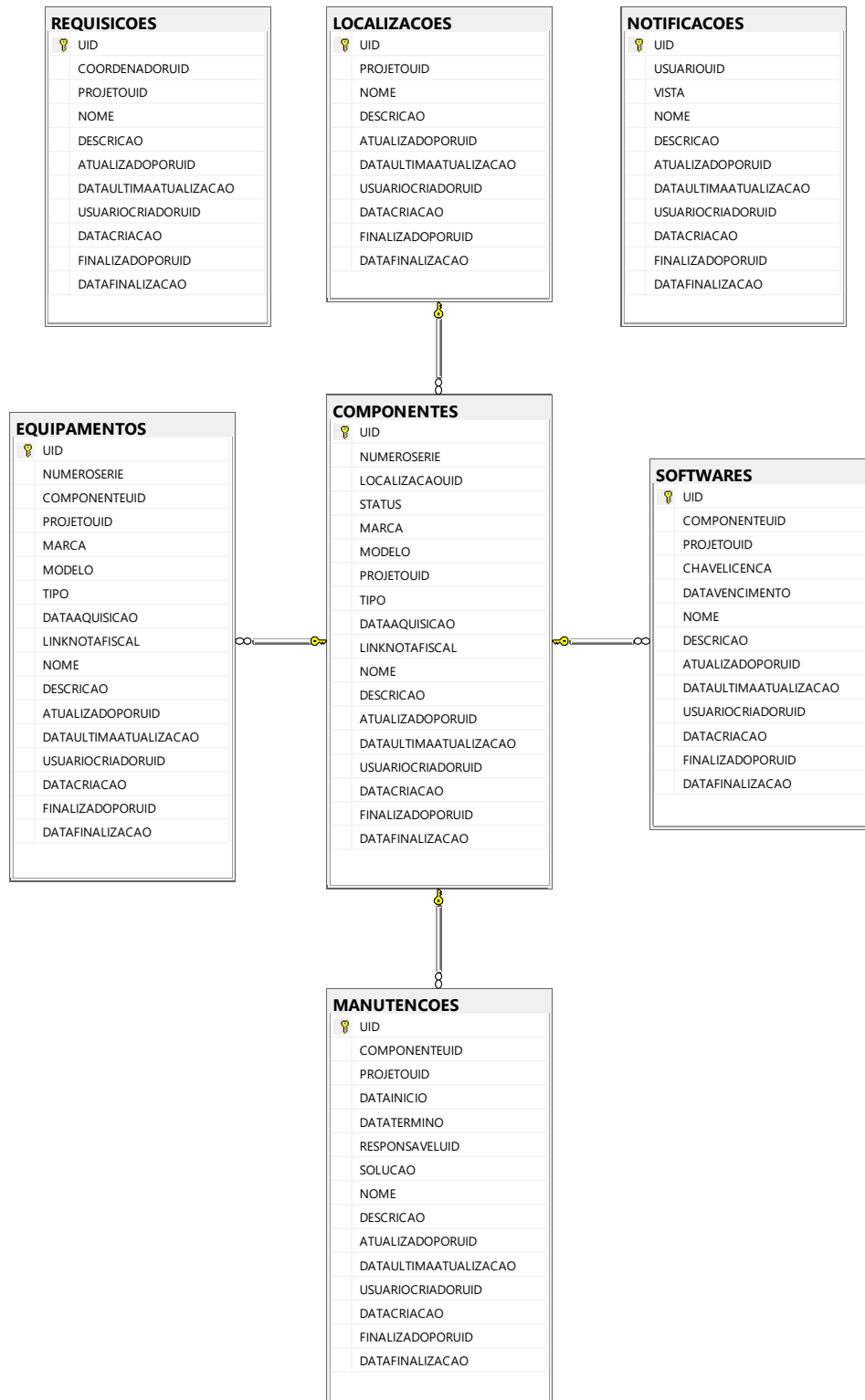
O banco de dados, assim como a aplicação, pode ser dividido em módulos distintos, apesar de na prática isso não ocorrer e todas as tabelas estarem relacionadas entre si, a divisão facilita a organização e visualização do banco. Isso pode ser visto nas figuras 17, 18 e 19:

Figura 17: Banco de dados de administração



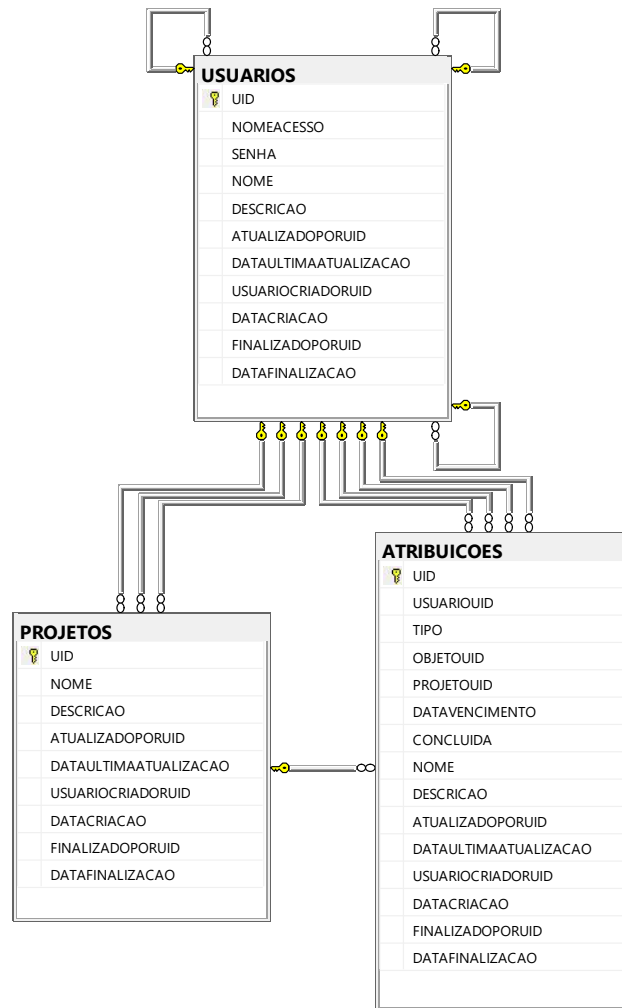
Fonte: Desenvolvido pelo autor, 2024

Figura 18: Banco de dados dos ativos



Fonte: Desenvolvido pelo autor, 2024

Figura 19: Banco de dados de Ciclo de Vida



Fonte: Desenvolvido pelo autor, 2024

A característica mais marcante do banco, de modo geral, é a existência dos campos **DATAFINALIZACAO** e **FINALIZADOPORUID**, e similares, apesar de não estarem documentados na especificação funcional. No sistema, nenhum registro é apagado do banco de dados, apenas do software. Isso significa que um registro ativo possui valores nulos para ambos os campos e após ser deletado no sistema as colunas receberão os dados respectivos. Conseqüentemente, o *back-end* é programado para ignorar registros que possuem valores em ambas as colunas, o que os faz não aparecer em consultas para os usuários.

A funcionalidade de não apagar dados do sistema foi feita com o intuito de promover maior segurança enquanto o software estiver funcionando, uma vez que, por engano, dados importantes podem ser apagados por usuários. Logo, utilizando o método descrito, é possível restaurar os dados e recuperar as perdas causadas.

#### 4.2.4 HTML, CSS e JavaScript

Para construir a aplicação front-end, algumas tecnologias webs comuns foram utilizadas. Todas possuem a característica de serem amplamente usadas, e algumas, inclusive, são consideradas fundamentais para o desenvolvimento de sites para a internet. Nos próximos parágrafos, serão descritas as tecnologias **HTML**, **CSS**, **SCSS**, **JavaScript** e **TypeScript** com maior aprofundamento.

**HTML** (*Hyper Text Markup Language*) é uma linguagem de marcação de texto que ajuda a construir a estrutura, formato, estilo e a marcação das páginas web (UBAH, 2021). É base para visualização de informações e define a estrutura primordial de todos os sites modernos.

**CSS** (*Cascading Style Sheets*) é uma linguagem de estilo utilizada para alterar o estilo e a aparência de uma página web, sendo possível criar sites mais atrativos para o usuário final (UBAH, 2021). No caso do projeto, foi utilizada uma variação de CSS chamada **SCSS** (*Sassy Cascading Style Sheets*) da qual adiciona novas funcionalidades que não existem no CSS padrão, e ajudam no desenvolvimento do projeto.

**JavaScript (JS)** é uma linguagem de script feita para dar movimento a página web. Assim, JS é capaz de manipular os elementos e o estilo da página web, realizar cálculos, fazer requisições e outras ações conforme necessário (UBAH, 2021). No sistema, foi utilizada uma variação de JS chamada **TypeScript (TS)**, que oferece uma variedade maior de funcionalidades úteis para o desenvolvimento da aplicação comparado a linguagem padrão.

#### 4.2.5 Angular

Para desenvolver a aplicação front-end, foi utilizado um framework web chamado *Angular*, que é utilizado para criar aplicações de página única eficientes e sofisticadas (Angular, 2019). Assim, os sites em angular são feitos em um único documento HTML que é alterado conforme a necessidade, a partir da manipulação da estrutura da página com *TypeScript*, que dá impressão da mudança de telas para o usuário final.

#### 4.2.6 .NET – C#

Por outro lado, a aplicação *back-end* foi desenvolvida utilizando o framework **.NET** e a linguagem de programação **C#**:

- **.NET**: É um framework muito vasto usado para criar aplicações modernas, robustas e escalonáveis (SOMMERVILLE, 2019)
- **C#**: É uma linguagem de programação orientada a objetos desenvolvida pela Microsoft e muito utilizada com **.NET** (SOMMERVILLE, 2019.).

#### 4.2.7 JWT

Para garantir maior segurança das requisições das duas aplicações, *front-end* e *back-end*, foram utilizados *tokens JWT (JSON Web Tokens)*. São tokens assinados digitalmente feitos para transmitir informações entre diferentes partes. (JWT.IO - JSON Web Tokens Introduction, [s.d.]). Funcionam de modo similar a uma chave de acesso.

Assim, para a aplicação *front-end* comunicar-se com *back-end* é preciso informar um *token* válido para que a requisição seja autorizada. Esse *token* é gerado e enviado ao *front-end* sempre que um usuário realizar login na aplicação, o que por consequência permite ao usuário realizar suas ações normalmente e garante que aplicativos terceiros não utilizem o servidor *back-end*.

#### 4.2.8 SQLServer

Para o banco de dados, no projeto, foi utilizado o **SQLServer**, um sistema de gerenciamento de banco de dados relacional que permite criar, armazenar e editar instâncias de banco de dados (RAY; WEST, 2023). O *back-end* é responsável por enviar os dados necessários para a instância do sistema para que as instruções sejam executadas corretamente.

#### 4.2.9 Bibliotecas e frameworks

Para o projeto, foi utilizada uma coletânea de bibliotecas e frameworks extras para apoiar no desenvolvimento. A seguir serão listados e descritos cada um:

- **Entity Framework:** É um framework de ORM (Object Relation Mapper) que permite o mapeamento das classes do projeto para integrá-las a instância do banco de dados e facilita a manipulação dos dados.
- **JWTBearer:** Permite a utilização de tokens JWT em requisições entre o servidor e o cliente.
- **BCrypt.Net-Next:** Biblioteca para encriptar textos utilizada no projeto para codificar senhas dos usuários no banco de dados.
- **AutoMapper:** Biblioteca que facilita a conversão entre objetos de diferentes classes.
- **Angular Material:** Biblioteca de componentes de interface de usuário feita para facilitar a criação de aplicações modernas em angular.

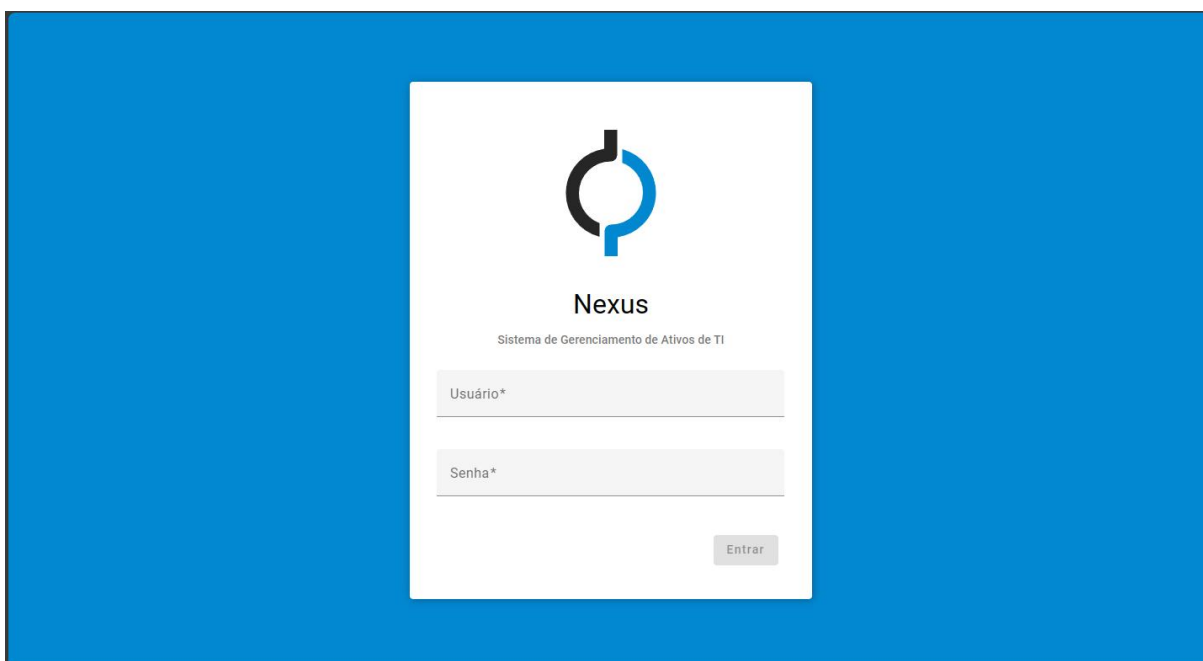
## 5 RESULTADOS E DISCUSSÕES

Após o desenvolvimento do projeto, foram obtidos resultados que ocasionaram na criação do software **NEXUS**. Para discutir o que foi feito, nas próximas seções será descrito o funcionamento do software e apresentado telas da aplicação, na perspectiva do usuário final. Conforme o perfil selecionado do usuário, as opções disponíveis podem variar, logo, para facilitar a explicação, as funcionalidades serão apresentadas de modo geral considerando o perfil **Administrador**.

### 5.1 Login

Para conectar-se na aplicação, é necessário possuir um nome de acesso e senha válidos, conforme a Figura 20:

Figura 20: Tela de login



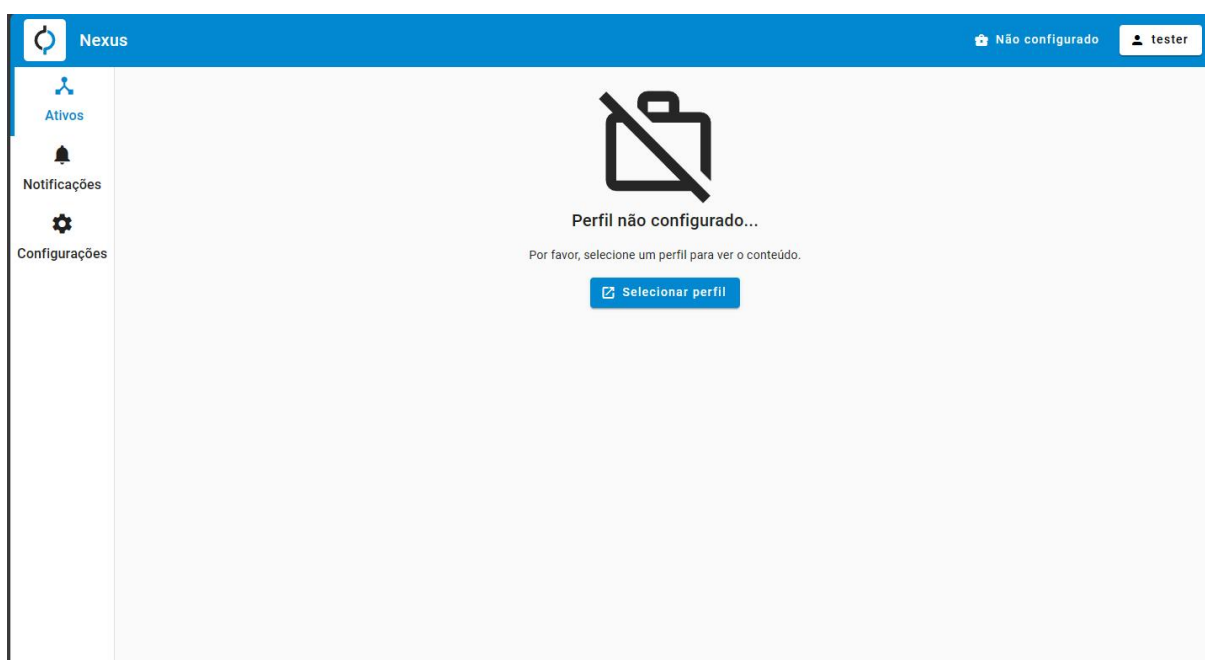
Fonte: Desenvolvido pelo autor, 2024

Basta inserir as credenciais que será possível ter acesso ao menu inicial e utilizar o sistema.

## 5.2 Alterar perfil ou projeto

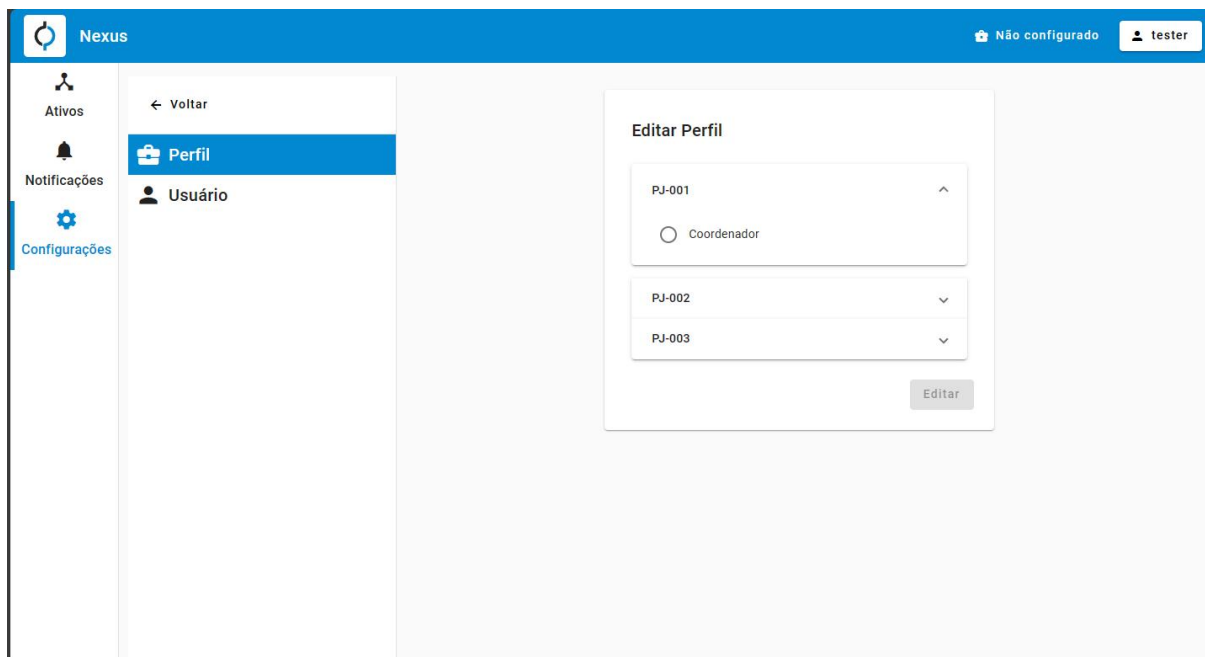
Para utilizar as funcionalidades do sistema, é preciso possuir um perfil ativo na aplicação. Na tela inicial, haverá um botão que indicará onde é possível observar os perfis disponíveis para o usuário. É necessário contatar um administrador para ter acesso a mais projetos e perfis. Conforme o perfil selecionado, as telas do software serão ligeiramente diferentes. Isso pode ser observado nas imagens 21, 22, 23 e 24:

Figura 21: Tela de perfil recém-criado



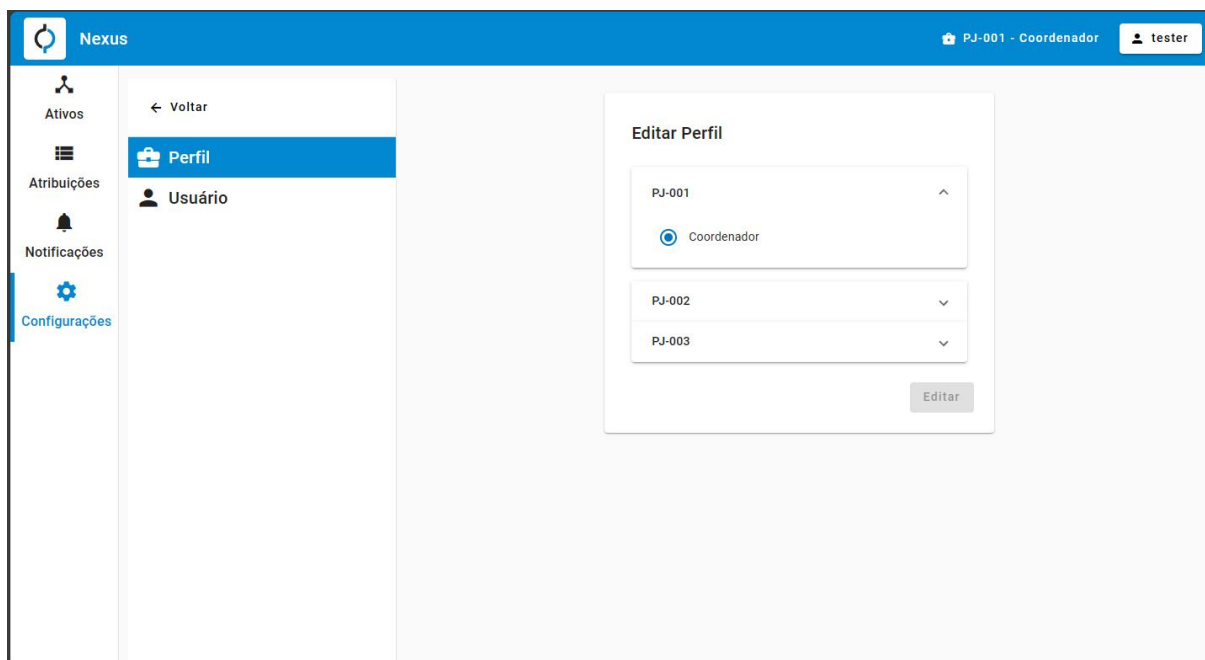
Fonte: Desenvolvido pelo autor, 2024

Figura 22: Tela de alterar perfil para usuários novos



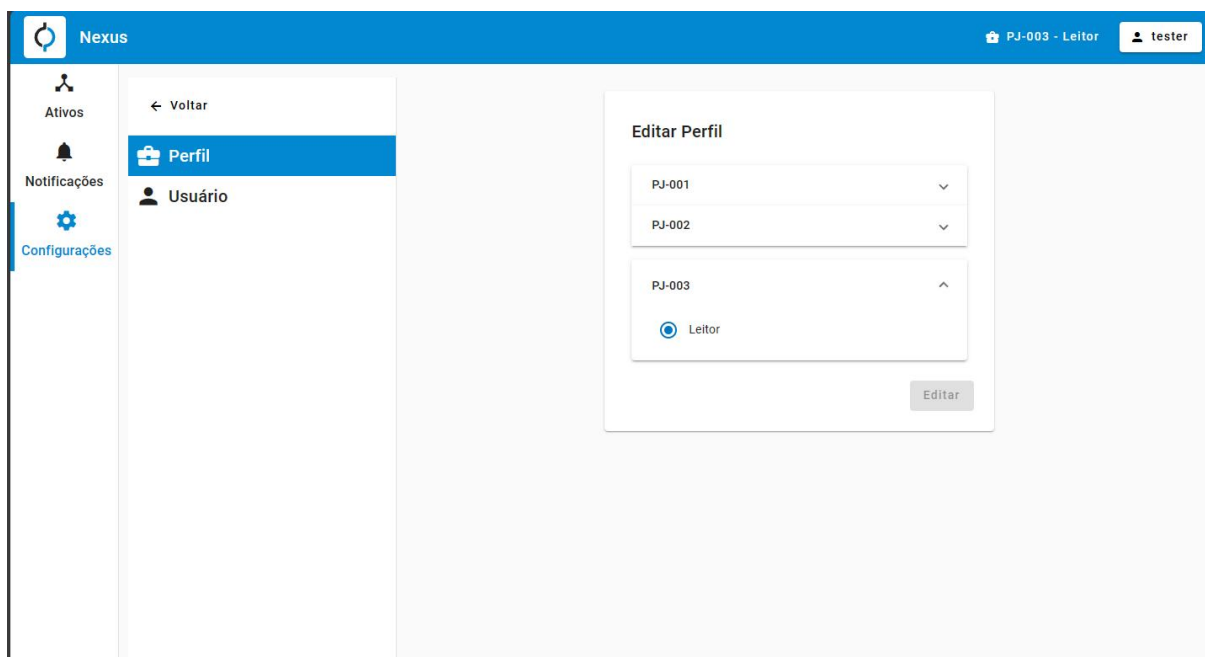
Fonte: Desenvolvido pelo autor, 2024

Figura 23: Perfil de coordenador selecionado



Fonte: Desenvolvido pelo autor, 2024

Figura 24: Perfil de leitor selecionado



Fonte: Desenvolvido pelo autor, 2024

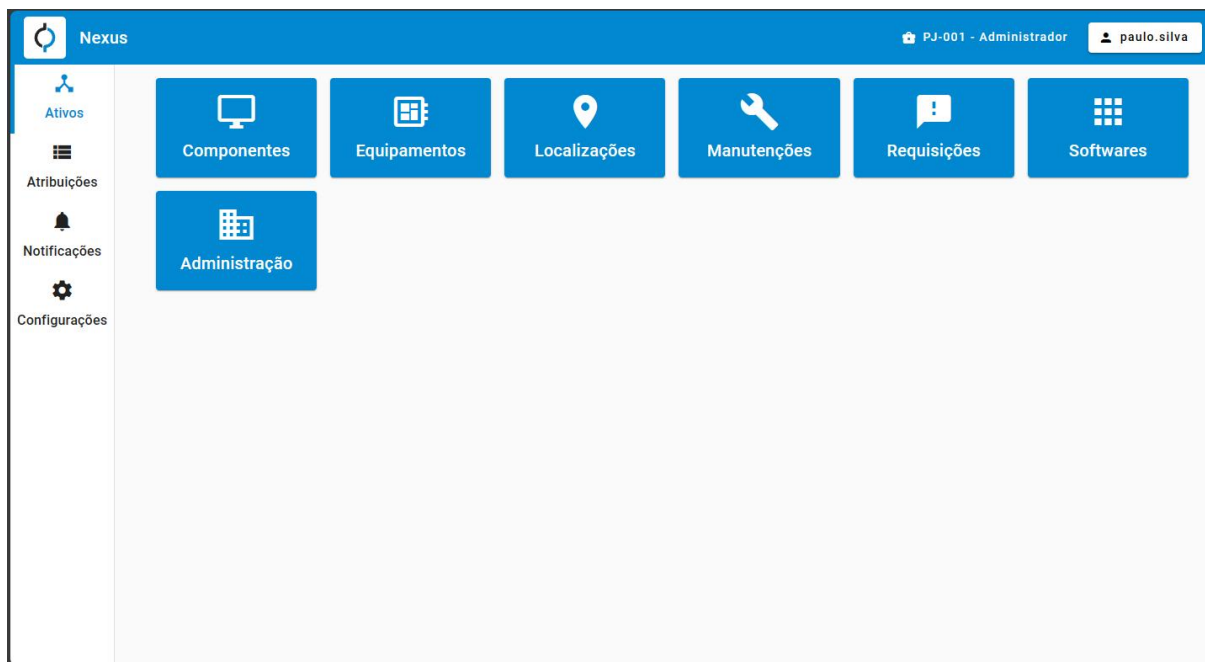
Por consequência, só será possível realizar ações em projetos do qual o usuário tem um perfil ativo. Por sua vez, esse perfil deverá ter as permissões necessárias para acessar as funções, caso contrário o botão da funcionalidade **não** aparecerá na interface de usuário.

### 5.3 Menu inicial

O menu inicial possui quatro seções, que serão descritas a seguir.

**Ativos** é onde se realiza as ações de manipulação de objetos, buscar, criar, editar e deletar. A tela pode ser vista na Figura 25:

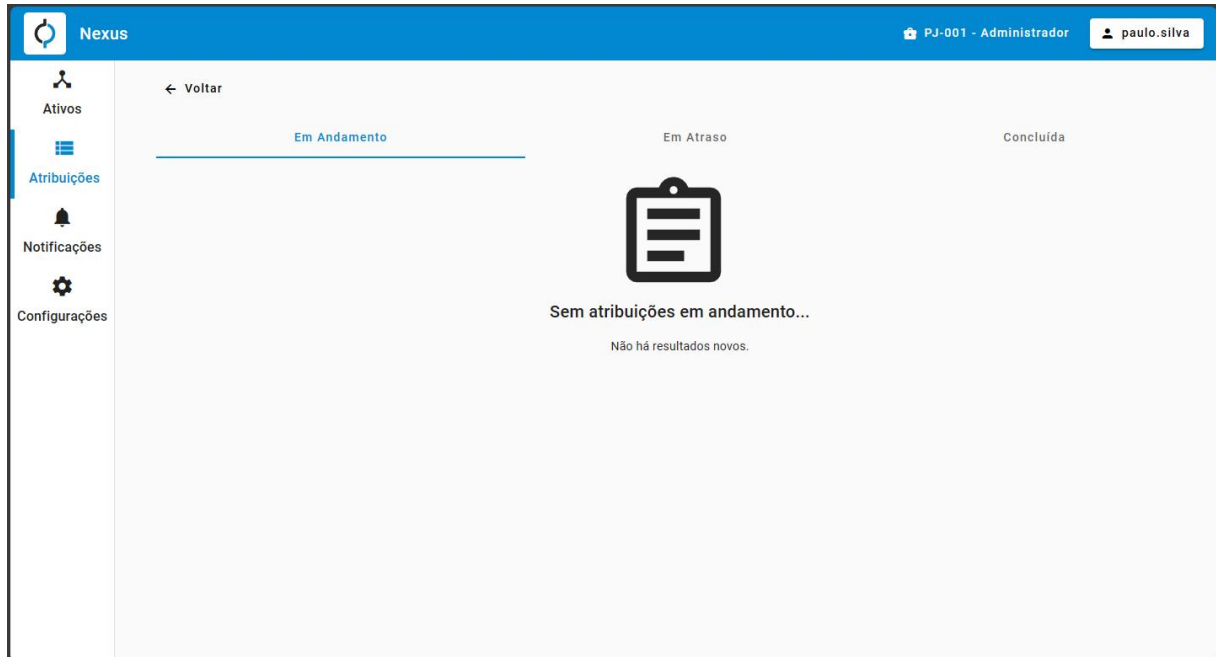
Figura 25: Tela de Ativos



Fonte: Desenvolvido pelo autor, 2024

**Atribuições** é a seção responsável por controlar as atribuições do ciclo de vida e permitir ações por parte do usuário nos processos de negócio dos ativos. É possível observar as atribuições em andamento (não concluídas), em atraso e concluídas, tal como indicado na Figura 26:

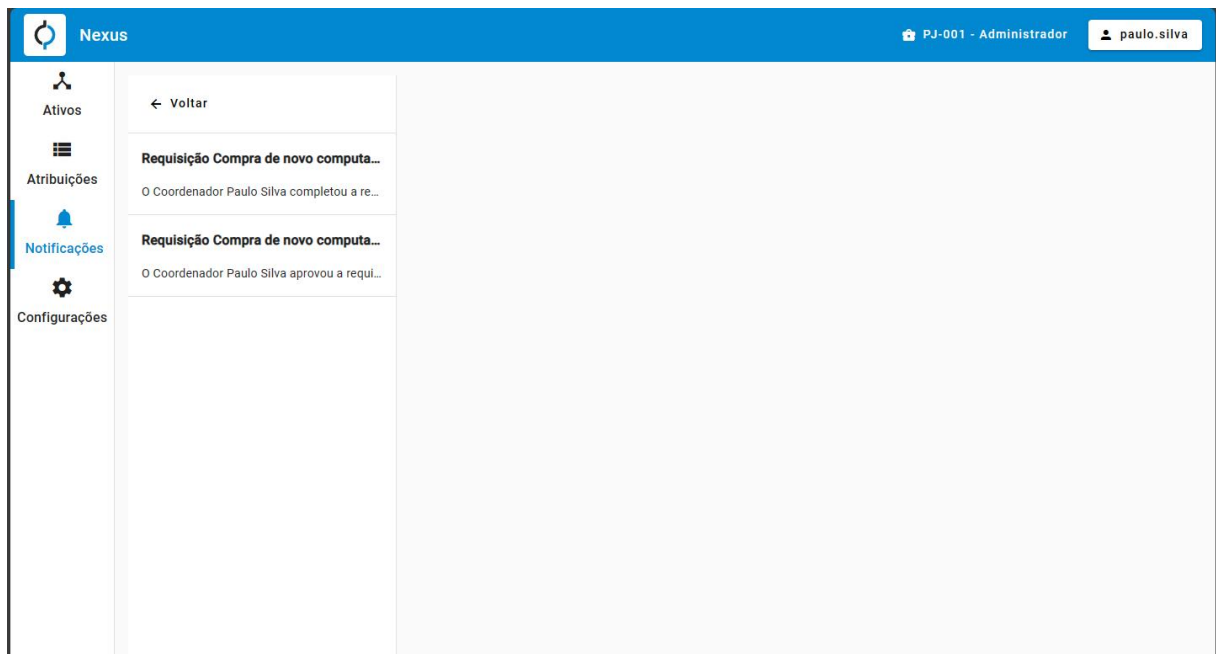
Figura 26: Tela de atribuições



Fonte: Desenvolvido pelo autor, 2024

**Notificações** é a área de notificações sobre mudanças no ciclo de vida de ativos, segundo a Figura 27:

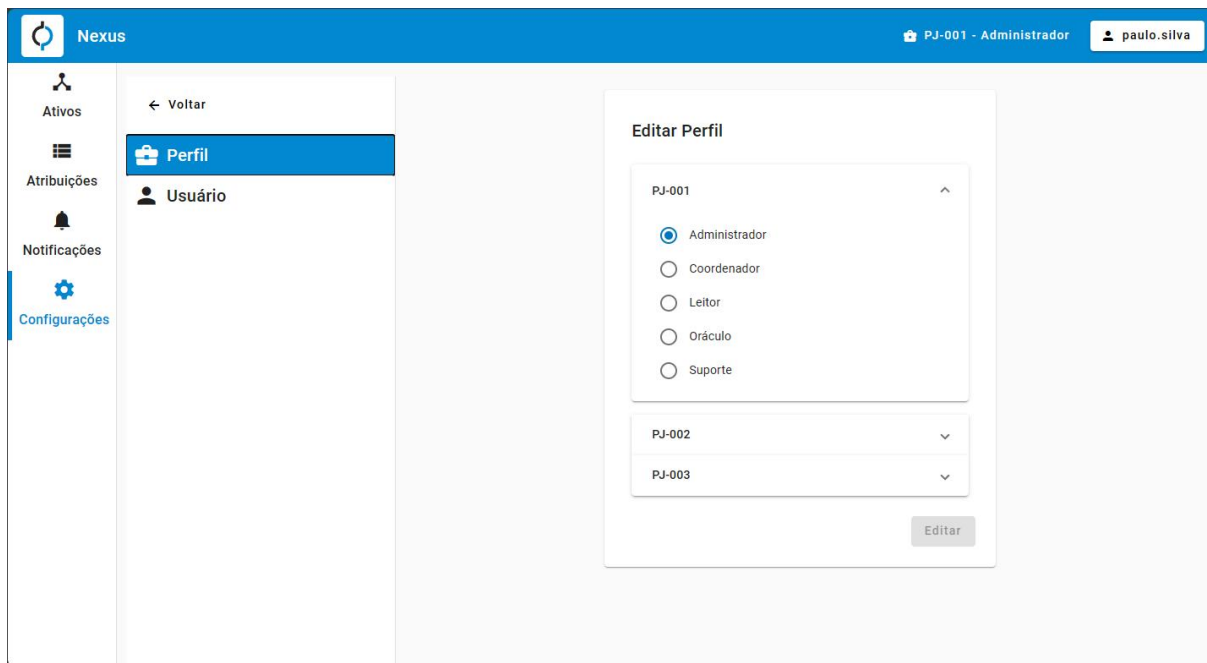
Figura 27: Tela de notificações



Fonte: Desenvolvido pelo autor, 2024

**Configurações** é área de preferências do usuário. Nessa seção, é possível alterar o perfil e projeto ativos para o usuário, dado a Figura 28:

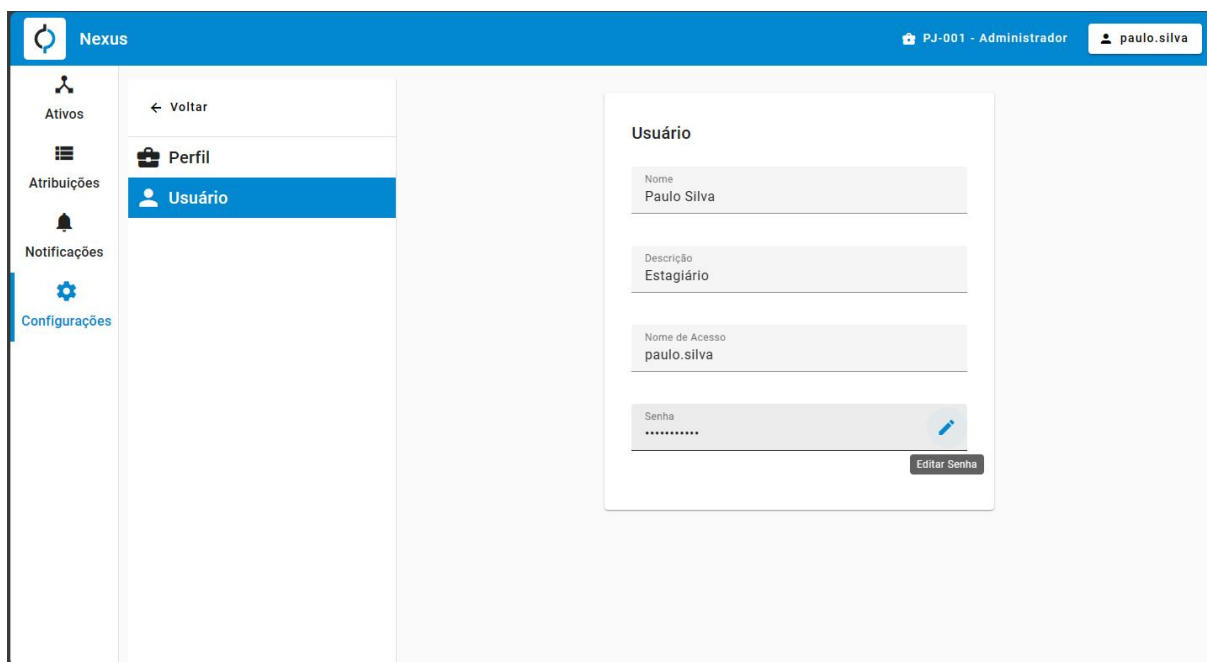
Figura 28: Tela de mudança de perfil



Fonte: Desenvolvido pelo autor, 2024

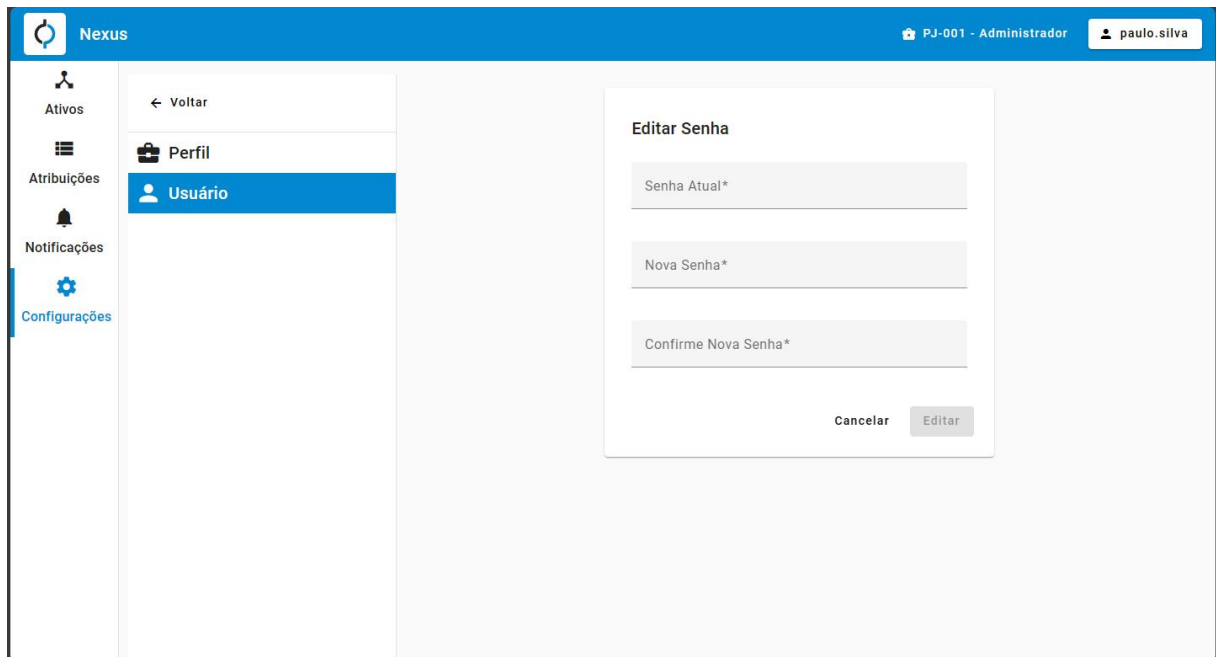
Além do mais, é possível ver as informações do usuário e alterar a senha da conta atual, conforme ilustrado nas figuras 29 e 30:

Figura 29: Tela de mudança de senha



Fonte: Desenvolvido pelo autor, 2024

Figura 30: Tela de editar senha

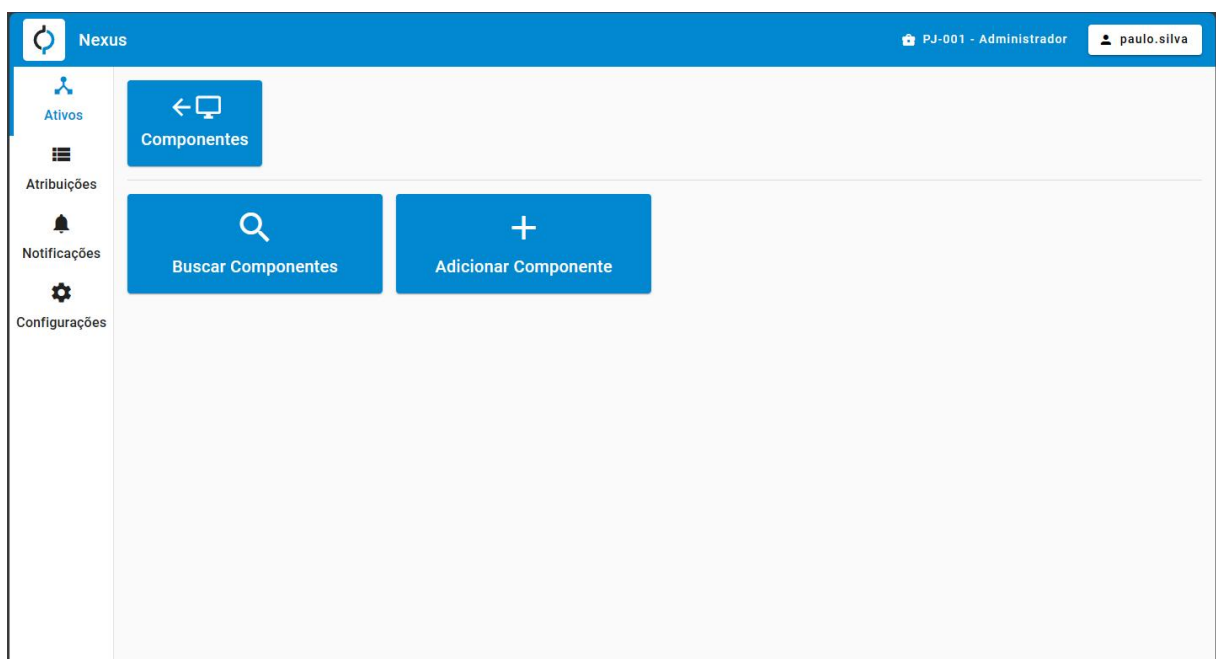


Fonte: Desenvolvido pelo autor, 2024

## 5.4 Adicionar objetos

Para adicionar um projeto a aplicação, selecione um botão no menu inicial de ativos e será possível observar as opções disponíveis. Neste exemplo, será utilizado o ativo **Componente**, como mostra a Figura 31:

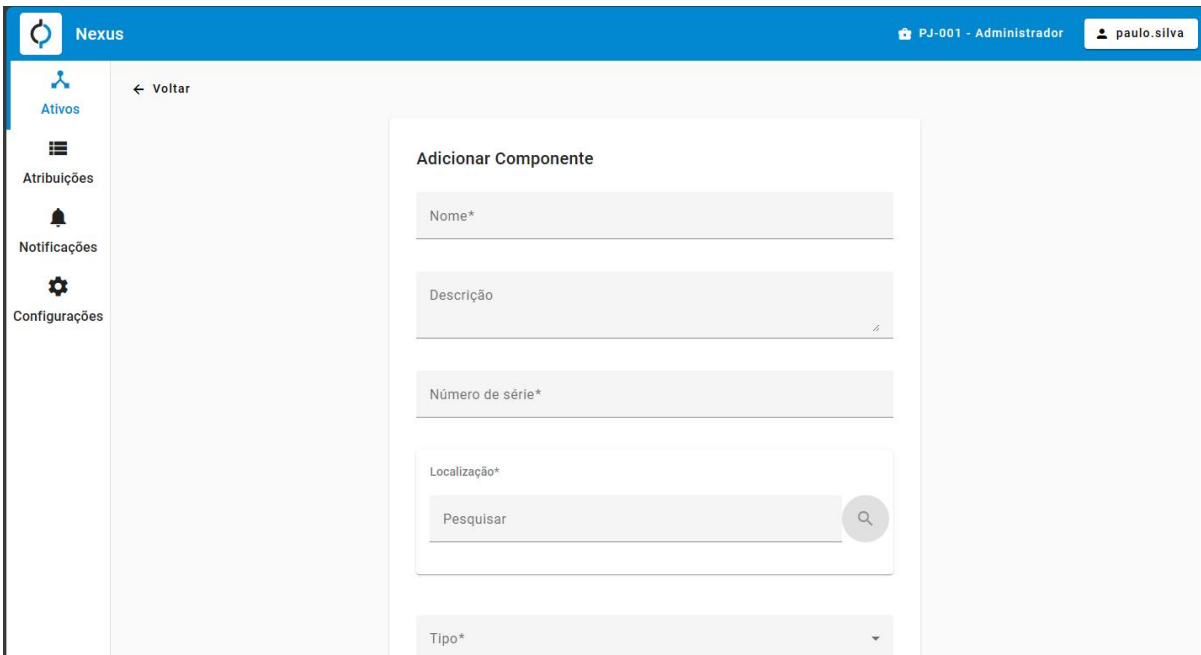
Figura 31: Tela de opções de Componentes



Fonte: Desenvolvido pelo autor, 2024

Após isso, selecione a opção de adicionar, e um formulário de criação estará disponível, dado a Figura 32. Basta preencher as opções obrigatórias, marcadas com \*, que será possível adicionar o objeto à aplicação.

Figura 32: Tela de criação de objeto



The screenshot shows the 'Adicionar Componente' (Add Component) form within the Nexus application. The interface includes a top navigation bar with the 'Nexus' logo, a user profile 'paulo.silva', and a role 'PJ-001 - Administrador'. A left sidebar contains navigation options: 'Ativos', 'Atribuições', 'Notificações', and 'Configurações'. The main content area features a 'Voltar' (Back) button and a form with the following fields: 'Nome\*' (Name), 'Descrição' (Description), 'Número de série\*' (Serial Number), 'Localização\*' (Location) with a search input labeled 'Pesquisar', and 'Tipo\*' (Type) as a dropdown menu.

Fonte: Desenvolvido pelo autor, 2024

## 5.5 Buscar objetos

Para pesquisar um objeto, selecione a opção de busca equivalente e será possível observar todos os objetos no projeto atual e seus principais atributos. Além do mais, existe a opção de filtrar a pesquisa por nome. Tais funcionalidades podem ser vistas nas figuras 33 e 34:

Figura 33: Tela de busca de objetos

The screenshot shows the Nexus application interface. The top header displays the logo, the name 'Nexus', and the user information 'PJ-001 - Administrador' and 'paulo.silva'. The sidebar on the left contains navigation icons for 'Ativos', 'Atribuições', 'Notificações', and 'Configurações'. The main content area is titled 'Localizações' and features a search bar with the placeholder text 'Pesquisar por nome...'. Below the search bar is a table with the following columns: 'Nome', 'Atualizado por', 'Data Última Atualização', 'Usuário Criador', and 'Data Criação'. The table contains seven rows of data. At the bottom of the table, there is a pagination control showing 'Itens por página: 25' and '1 - 7 de 7'.

Nome	Atualizado por	Data Última Atualização	Usuário Criador	Data Criação
LAB-02			Paulo Silva	04/05/2024 14:54:09
LAB-01			Paulo Silva	04/05/2024 14:53:53
ETQ-01	Paulo Silva	04/05/2024 14:54:23	Paulo Silva	04/05/2024 14:53:32
SEE			Paulo Silva	04/05/2024 14:53:09
SAL-03			Paulo Silva	13/04/2024 15:12:12
SAL-02			Paulo Silva	13/04/2024 15:12:05
SAL-01	Paulo Silva	13/04/2024 15:12:18	Paulo Silva	13/04/2024 15:11:54

Fonte: Desenvolvido pelo autor, 2024

Figura 34: Tela de busca com filtro de nome

The screenshot shows the Nexus application interface with a search filter applied. The search bar now contains the text 'LAB'. Below the search bar, there is a filter indicator that says 'Resultados para "LAB"'. The table below shows only two rows of data, corresponding to the filtered results. The pagination control at the bottom shows 'Itens por página: 25' and '1 - 2 de 2'.

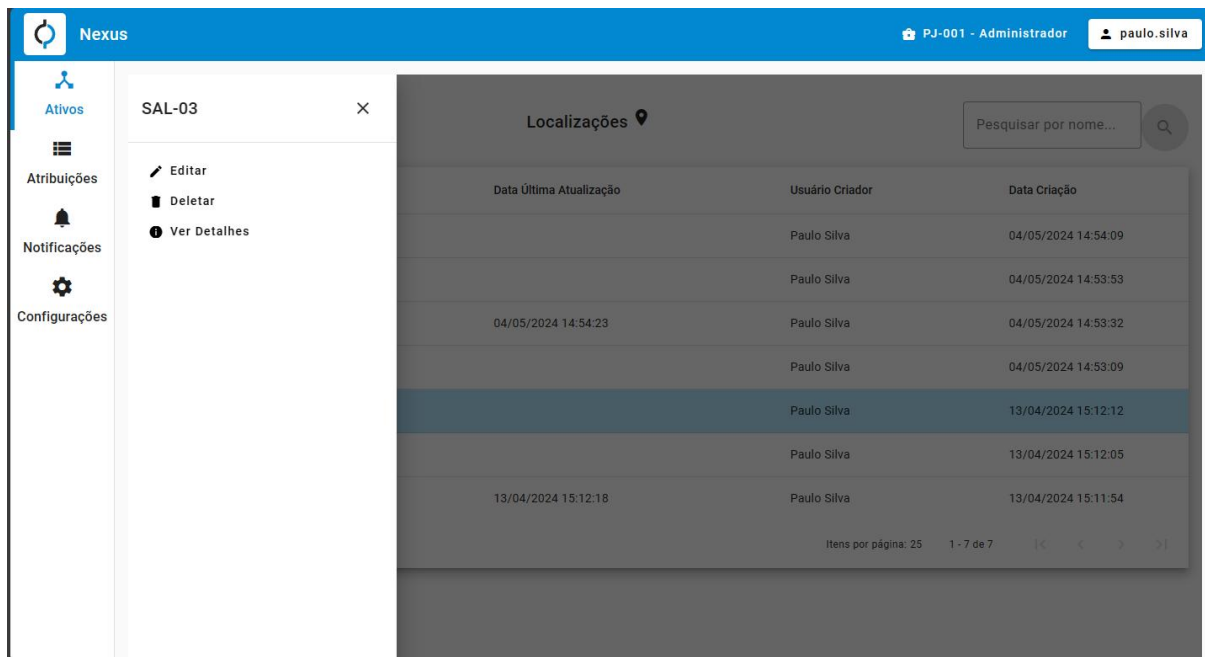
Nome	Atualizado por	Data Última Atualização	Usuário Criador	Data Criação
LAB-02			Paulo Silva	04/05/2024 14:54:09
LAB-01			Paulo Silva	04/05/2024 14:53:53

Fonte: Desenvolvido pelo autor, 2024

## 5.6 Editar objetos

Para editar um objeto, na tela de busca, clique em uma linha e a opção aparecerá na tela em um menu lateral, conforme a Figura 35:

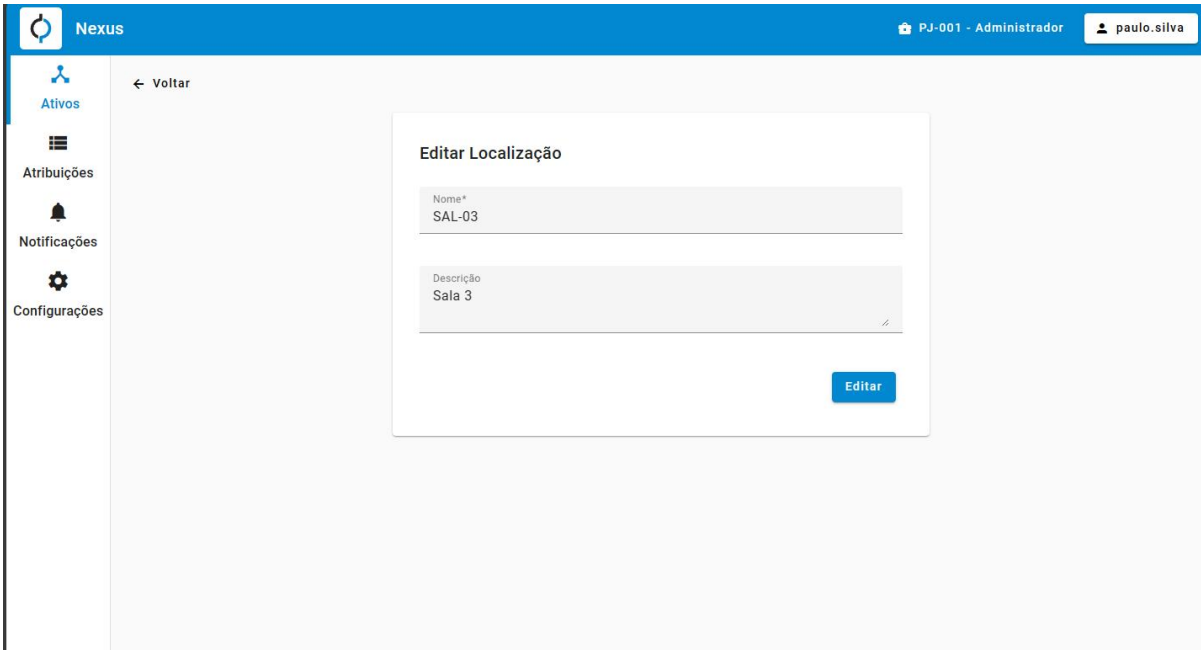
Figura 35: Opções de um objeto



Fonte: Desenvolvido pelo autor, 2024

Após isso, clique no botão com o rótulo de editar, e formulário com as informações já preenchidas do objeto aparecerá para permitir a edição, conforme a Figura a 36:

Figura 36: Tela de edição de objeto



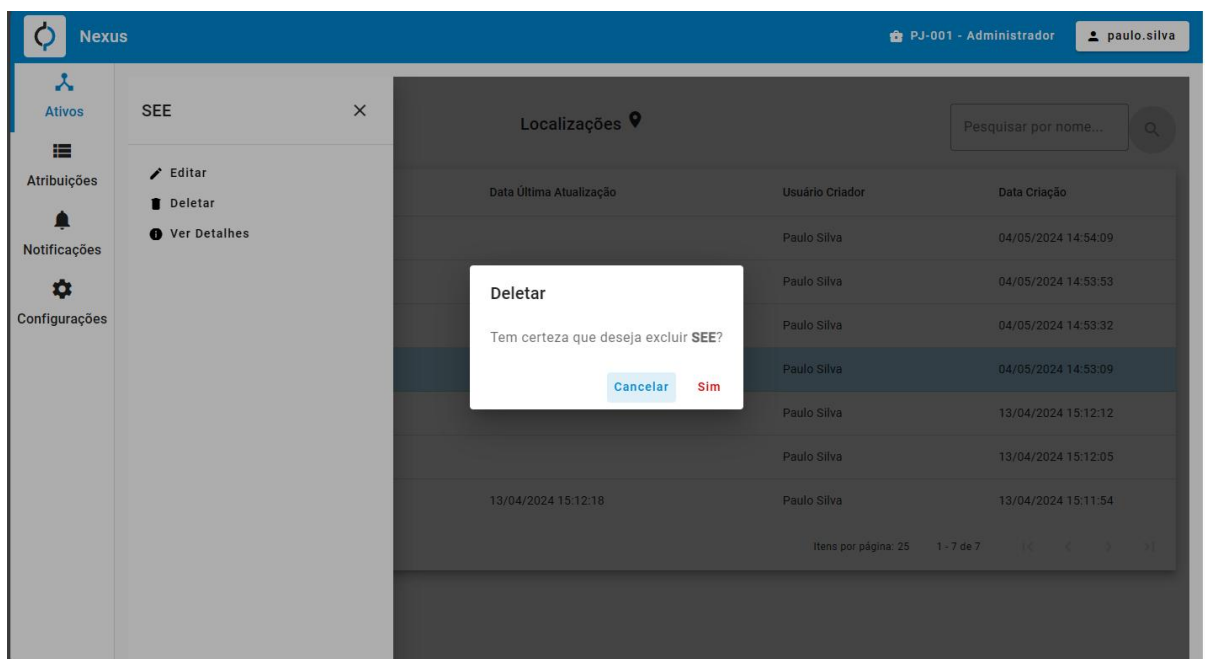
The screenshot shows the 'Editar Localização' (Edit Location) form within the Nexus system. The interface includes a blue header with the 'Nexus' logo, the user role 'PJ-001 - Administrador', and the user name 'paulo.silva'. A left sidebar contains navigation options: 'Ativos', 'Atribuições', 'Notificações', and 'Configurações'. The main content area features a 'Voltar' (Back) button and a central form titled 'Editar Localização'. The form contains two input fields: 'Nome\*' with the value 'SAL-03' and 'Descrição' with the value 'Sala 3'. A blue 'Editar' (Edit) button is positioned at the bottom right of the form.

Fonte: Desenvolvido pelo autor, 2024

## 5.7 Deletar objetos

Para deletar um objeto, clique no botão de deletar no menu de opções e uma caixa de diálogo aparecerá para confirmar a exclusão. Caso o resultado seja positivo, o objeto será excluído do sistema. Como pode ser visto na imagem a 37:

Figura 37: Caixa de diálogo para excluir objeto

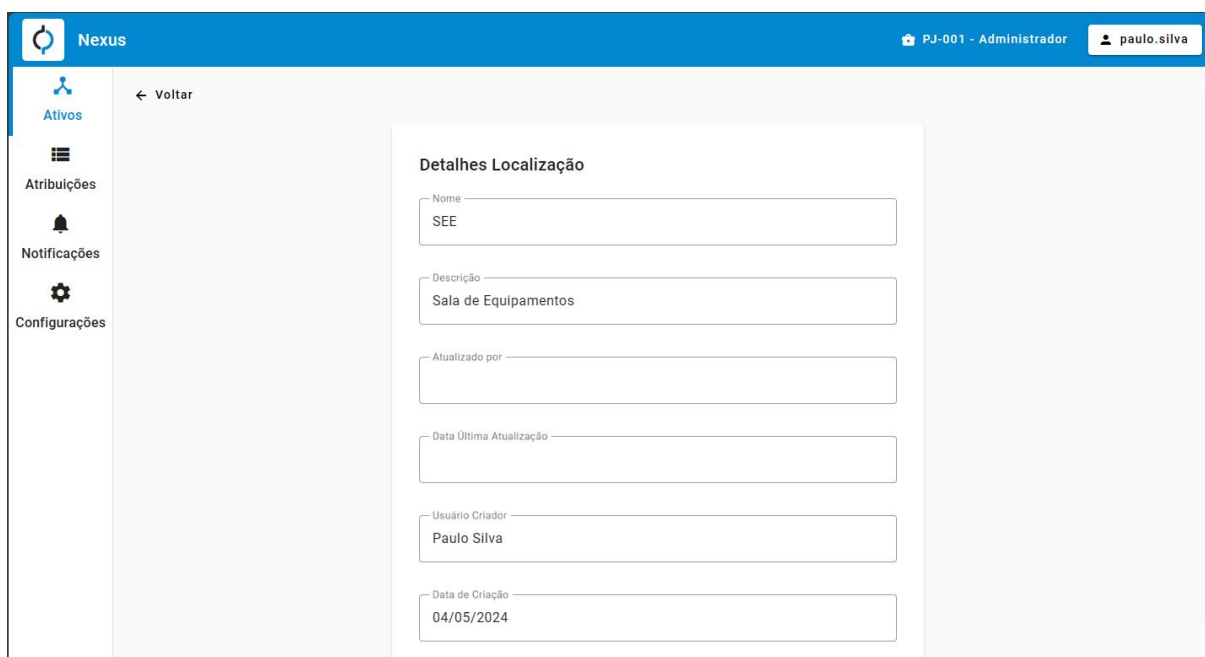


Fonte: Desenvolvido pelo autor, 2024

## 5.8 Detalhes de objeto

Para ver detalhes um objeto, clique no botão de detalhes no menu de opções. Assim, poderá ser visto todas as informações do objeto, como pode ser visto na imagem 38:

Figura 38: Tela de detalhes dos objetos

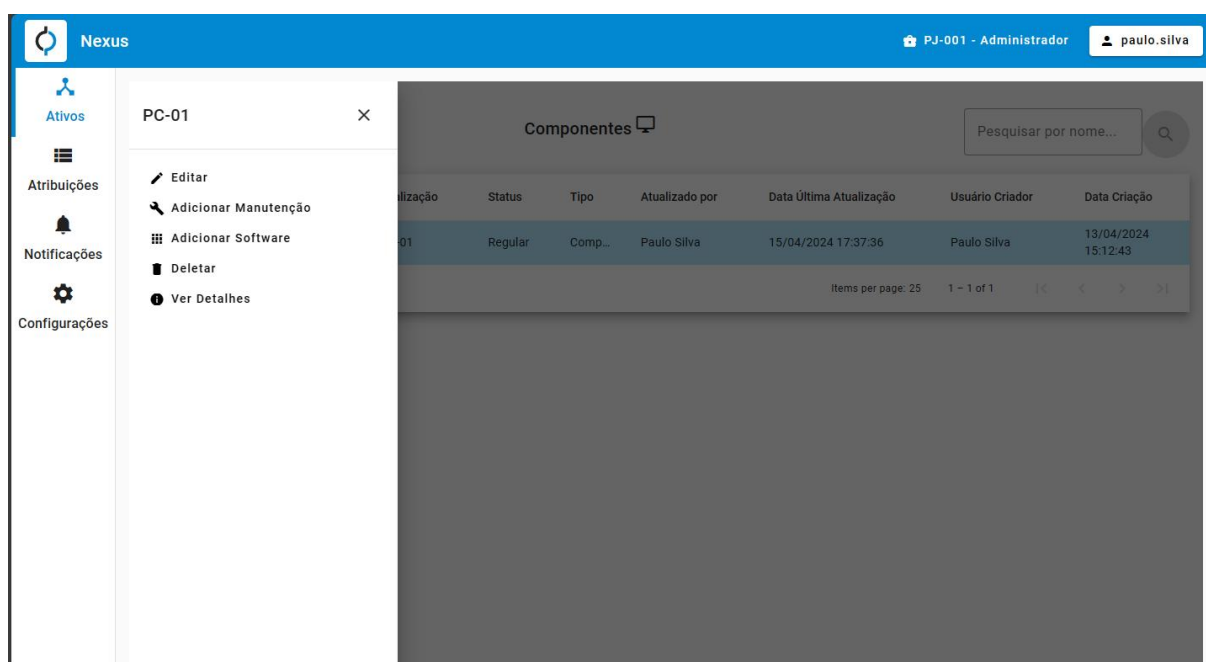


Fonte: Desenvolvido pelo autor, 2024

## 5.9 Adicionar manutenções e softwares

Diferente de outros objetos, para adicionar softwares e manutenções é preciso buscar um componente e adicionar a manutenção ou software por meio do menu de ações, como mostra a Figura 39. Assim, automaticamente o software ou manutenção ficará vinculada ao componente em questão.

Figura 39: Ações de adicionar manutenção e software

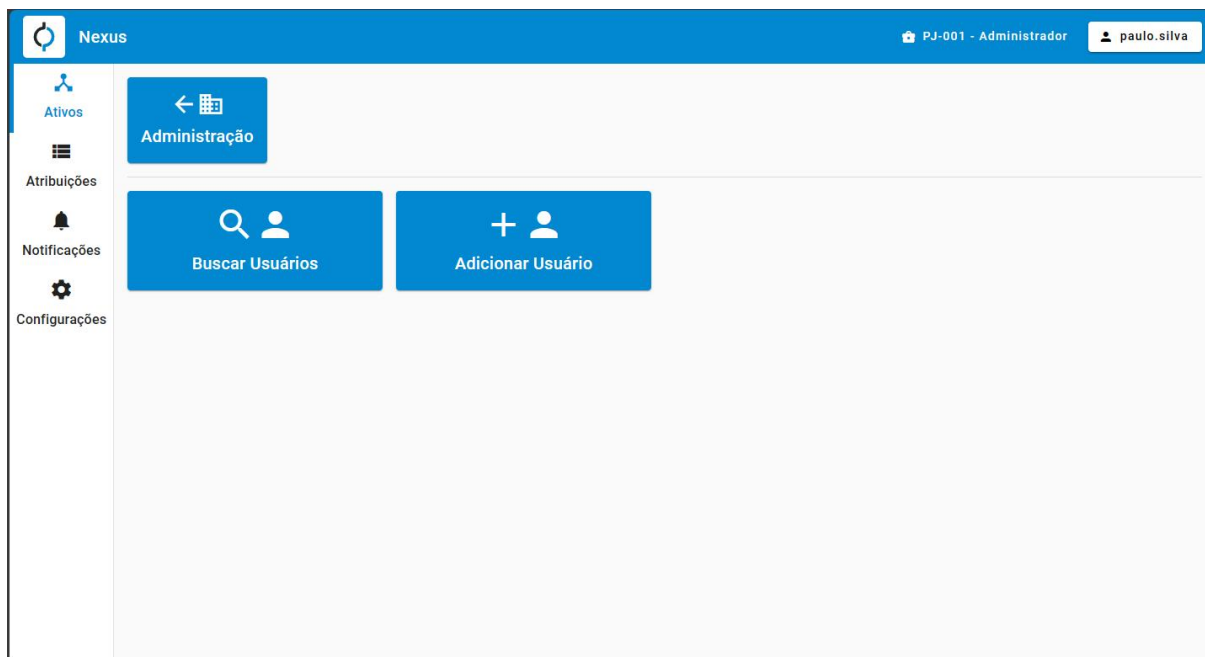


Fonte: Desenvolvido pelo autor, 2024

## 5.10 Administração

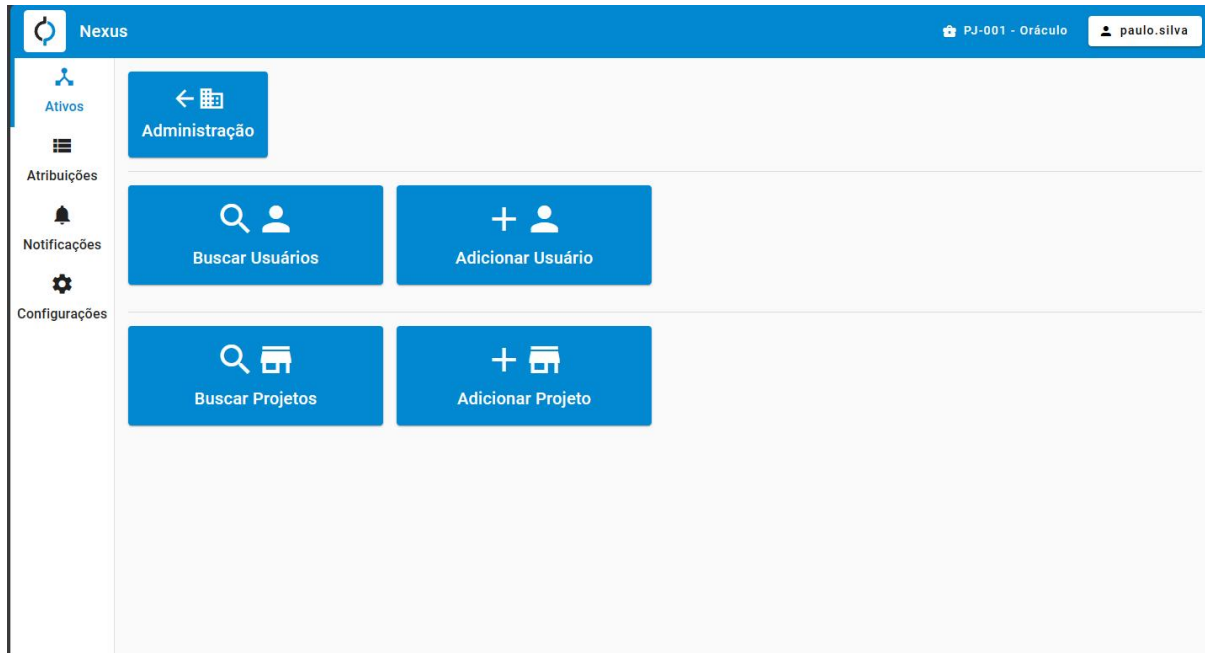
No menu de administração, é possível acessar as funções administrativas, que são a manipulação de usuários e projetos. Administradores podem acessar as opções para usuários, enquanto pessoas com o perfil oráculo podem acessar as funções de usuários e projetos, como mostram as imagens 40 e 41:

Figura 40: Tela de administração para perfil administrador



Fonte: Desenvolvido pelo autor, 2024

Figura 41: Tela de administração para perfil oráculo



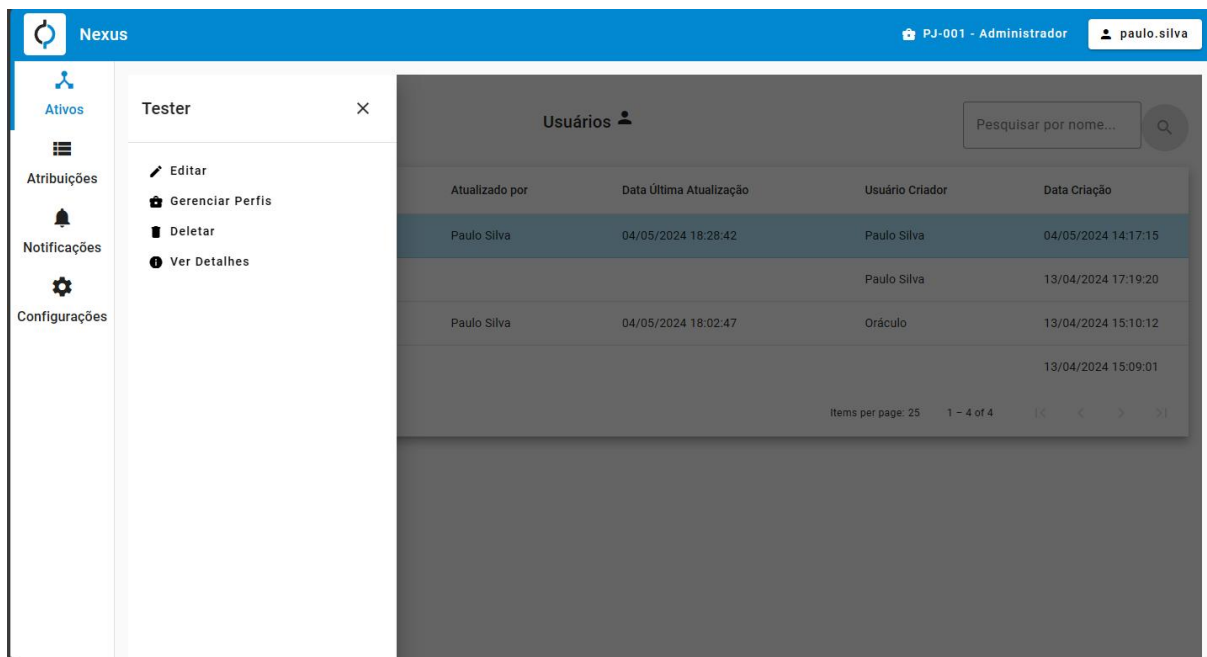
Fonte: Desenvolvido pelo autor, 2024

Em ambos os casos, será possível realizar a manipulação dos objetos, como fora descrito nas subseções anteriores.

## 5.11 Gerenciar Perfis

Dentre as ações disponíveis para a manipulação de usuários, está o gerenciamento de perfis. Para ver essa opção, selecione um usuário na busca, como mostra a imagem 42:

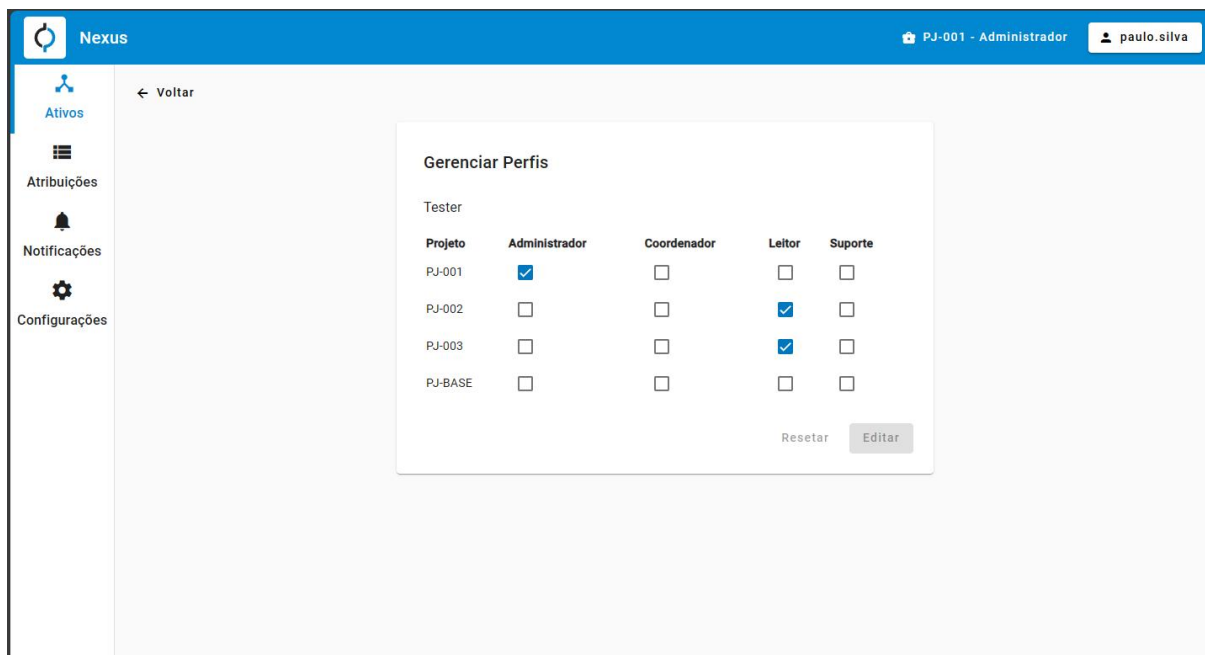
Figura 42: Ação de gerenciar perfis



Fonte: Desenvolvido pelo autor, 2024

Após clicar no botão, será possível remover ou dar acesso a perfis para o determinado usuário, conforme o projeto, tal como a imagem 43:

Figura 43: Tela de gerenciar perfis

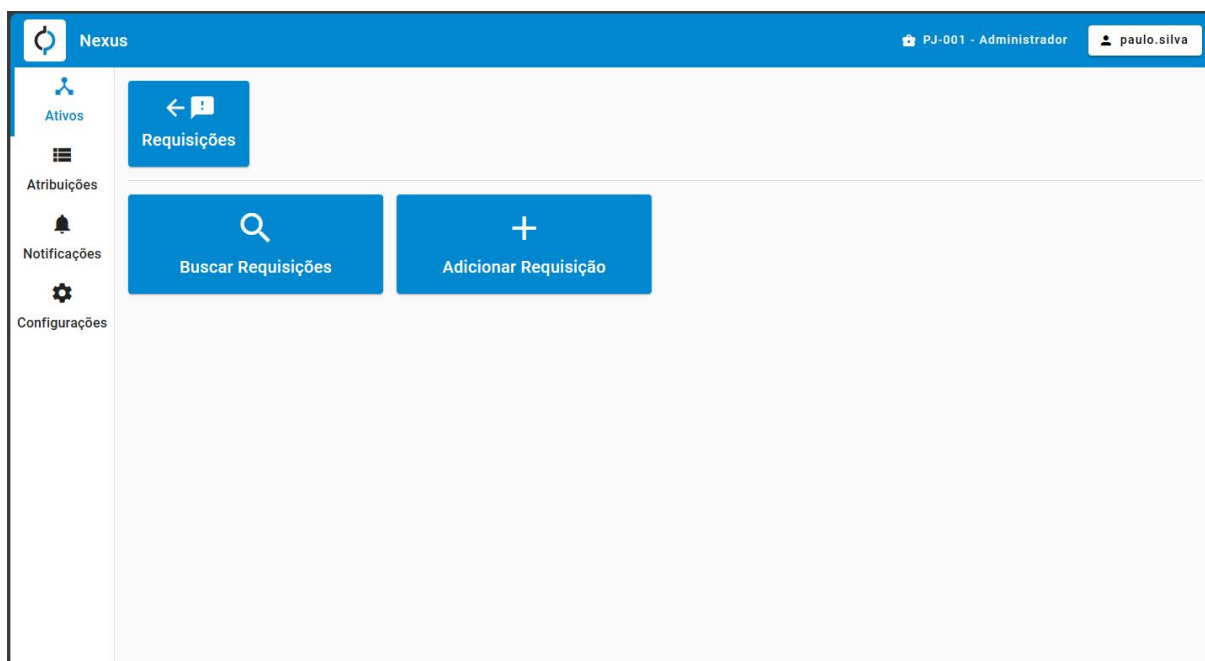


Fonte: Desenvolvido pelo autor, 2024

## 5.12 Análise de requisição

Para iniciar o ciclo de vida de análise de requisição, é preciso criar uma requisição por meio do menu de ativos, como mostra a imagem 44:

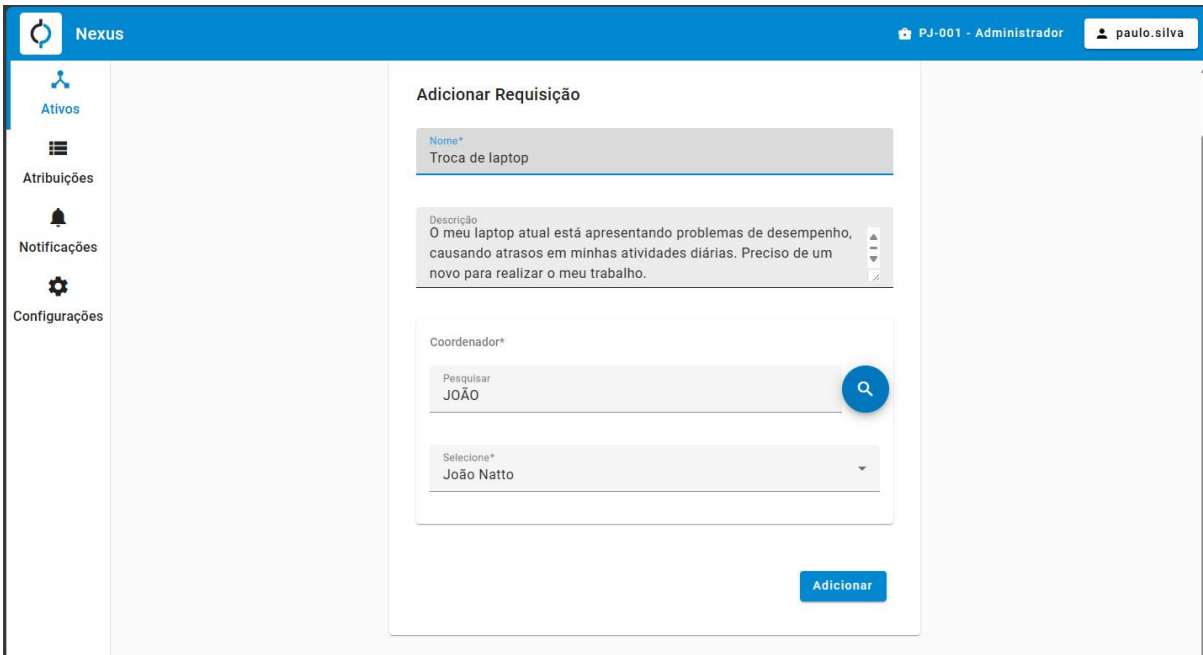
Figura 44: Menu de requisições



Fonte: Desenvolvido pelo autor, 2024

Após clicar na opção de adicionar requisição, será necessário escrever o nome e atribuí-la a um coordenador responsável pela análise, além de que opcionalmente é possível descrever em detalhes o pedido. Essas informações podem ser preenchidas a partir do formulário presente na Figura 45:

Figura 45: Tela de criação de requisição



The screenshot shows the 'Adicionar Requisição' (Add Request) form in the Nexus system. The interface includes a sidebar with navigation options: 'Ativos', 'Atribuições', 'Notificações', and 'Configurações'. The main content area contains the following fields:

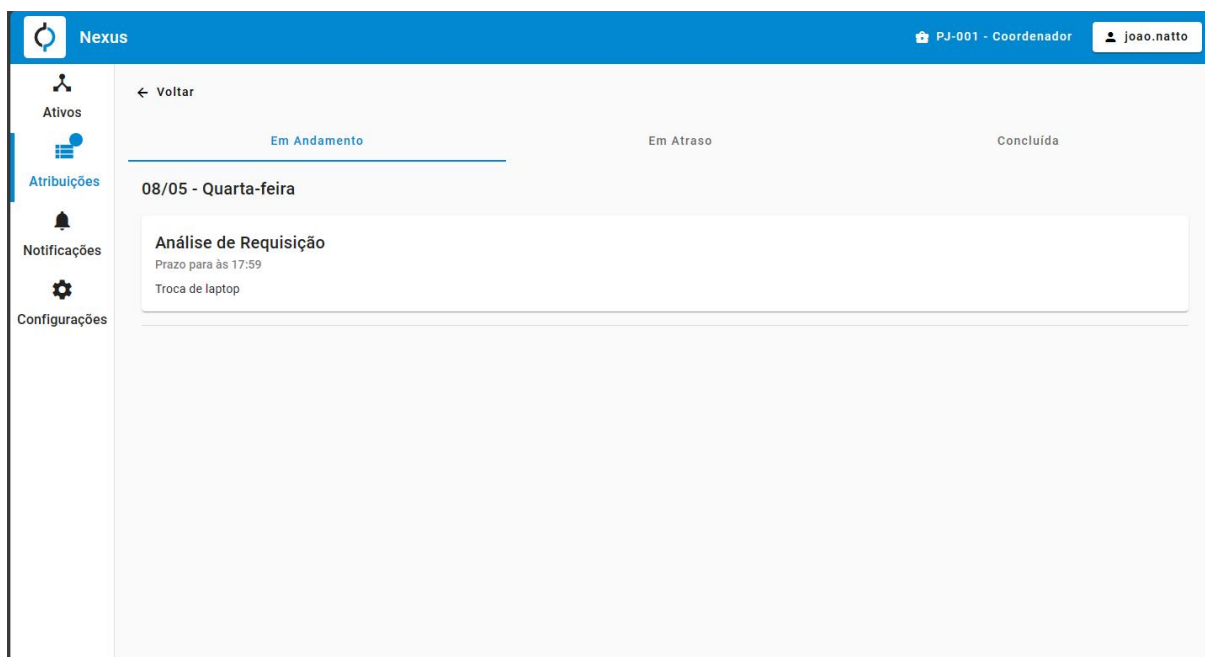
- Nome\***: A text input field containing 'Troca de laptop'.
- Descrição**: A text area containing the text: 'O meu laptop atual está apresentando problemas de desempenho, causando atrasos em minhas atividades diárias. Preciso de um novo para realizar o meu trabalho.'
- Coordenador\***: A section with a search input field containing 'JOÃO' and a search icon, and a dropdown menu labeled 'Selecione\*' with 'João Natto' selected.
- Adicionar**: A blue button at the bottom right of the form.

The top navigation bar shows the user is logged in as 'paulo.silva' with the role 'PJ-001 - Administrador'.

Fonte: Desenvolvido pelo autor, 2024

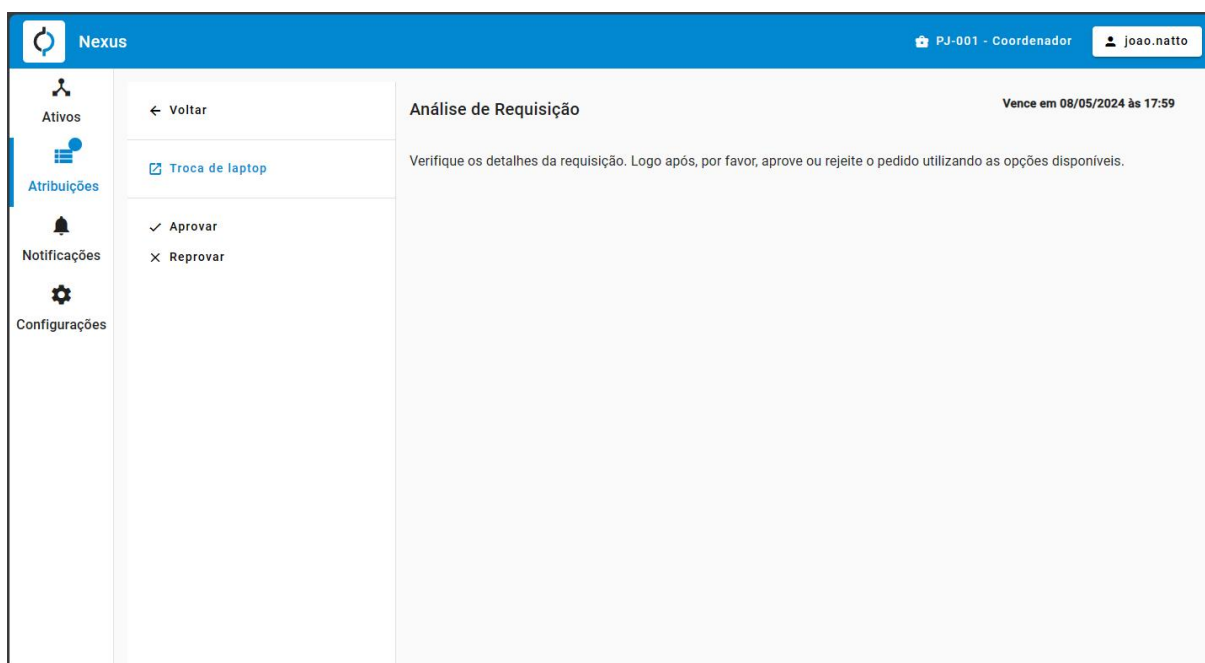
Após a criação, automaticamente o coordenador receberá uma atribuição para que seja feita a análise da requisição, sendo possível aprovar ou rejeitar o pedido, além de ver os detalhes da requisição, conforme mostram as imagens 46 e 47:

Figura 46: Tela de atribuições com requisição



Fonte: Desenvolvido pelo autor, 2024

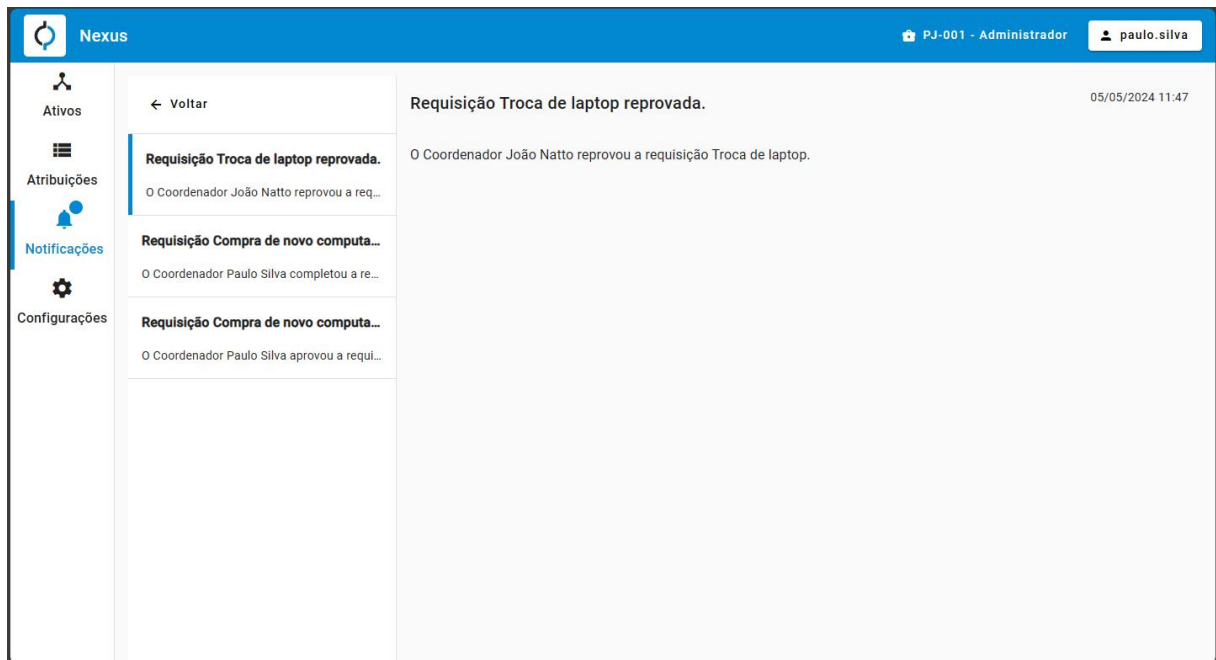
Figura 47: Tela de análise de requisição



Fonte: Desenvolvido pelo autor, 2024

Se a requisição for reprovada, automaticamente o solicitante receberá uma notificação informando que seu pedido foi recusado pelo coordenador responsável e o ciclo se encerra, tal como é possível ver na Figura 48:

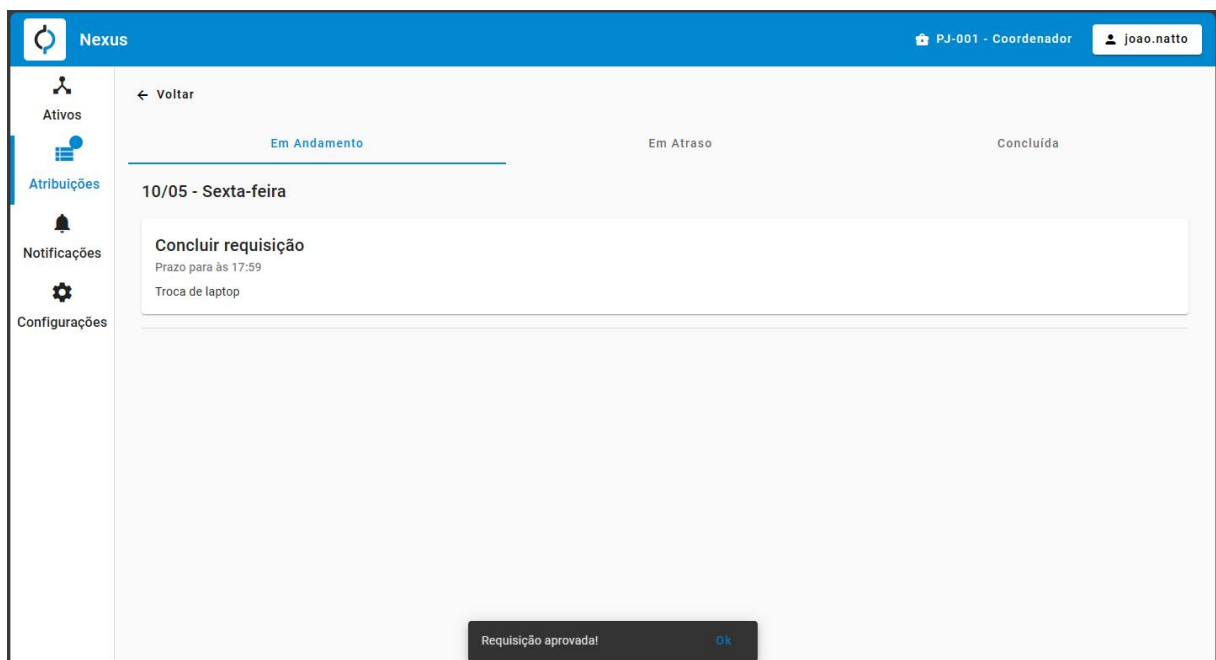
Figura 48: Notificação de requisição reprovada



Fonte: Desenvolvido pelo autor, 2024

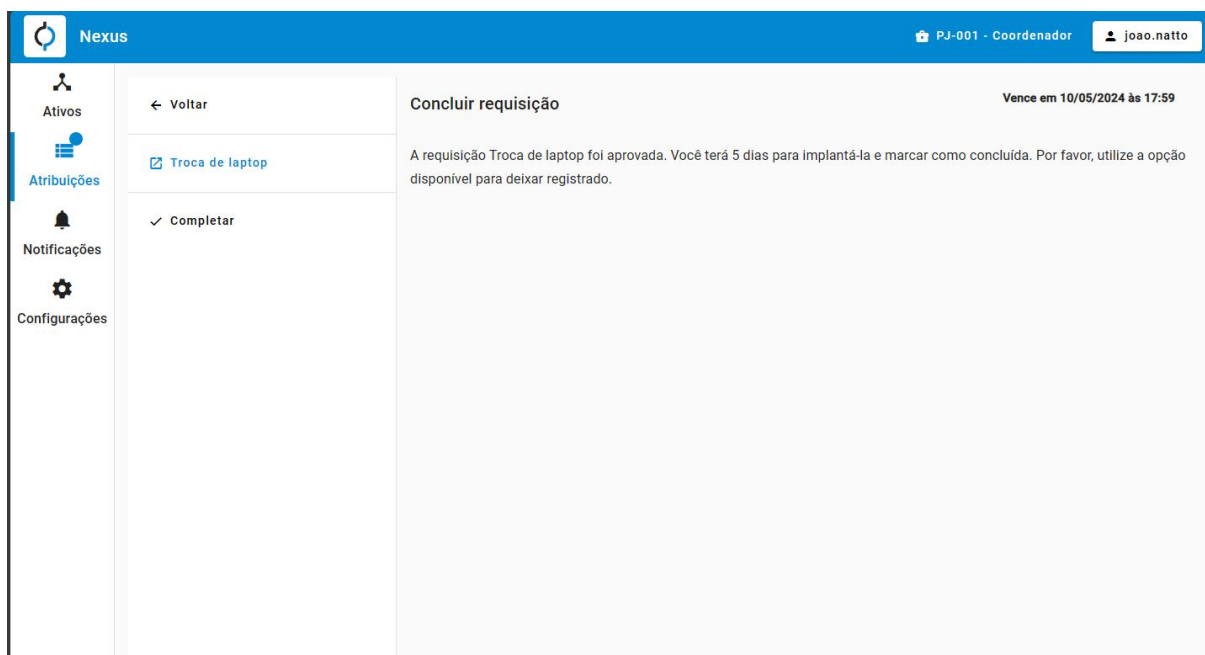
Caso contrário, o coordenador receberá uma nova atribuição para que possa marcar como concluída para cumprir a requisição do solicitante, como mostram as imagens 49 e 50:

Figura 49: Atribuição de concluir requisição



Fonte: Desenvolvido pelo autor, 2024

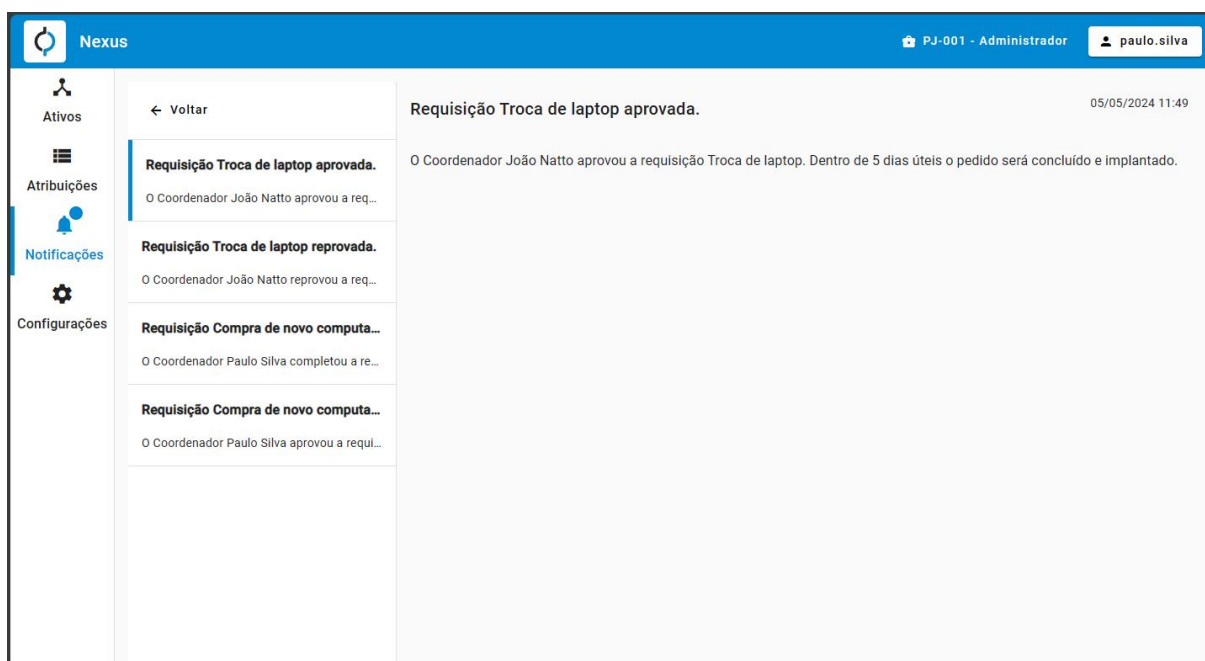
Figura 50: Tela de concluir requisição



Fonte: Desenvolvido pelo autor, 2024

O solicitante, inclusive, também receberá uma notificação informando que sua requisição foi aprovada, conforme a imagem 51:

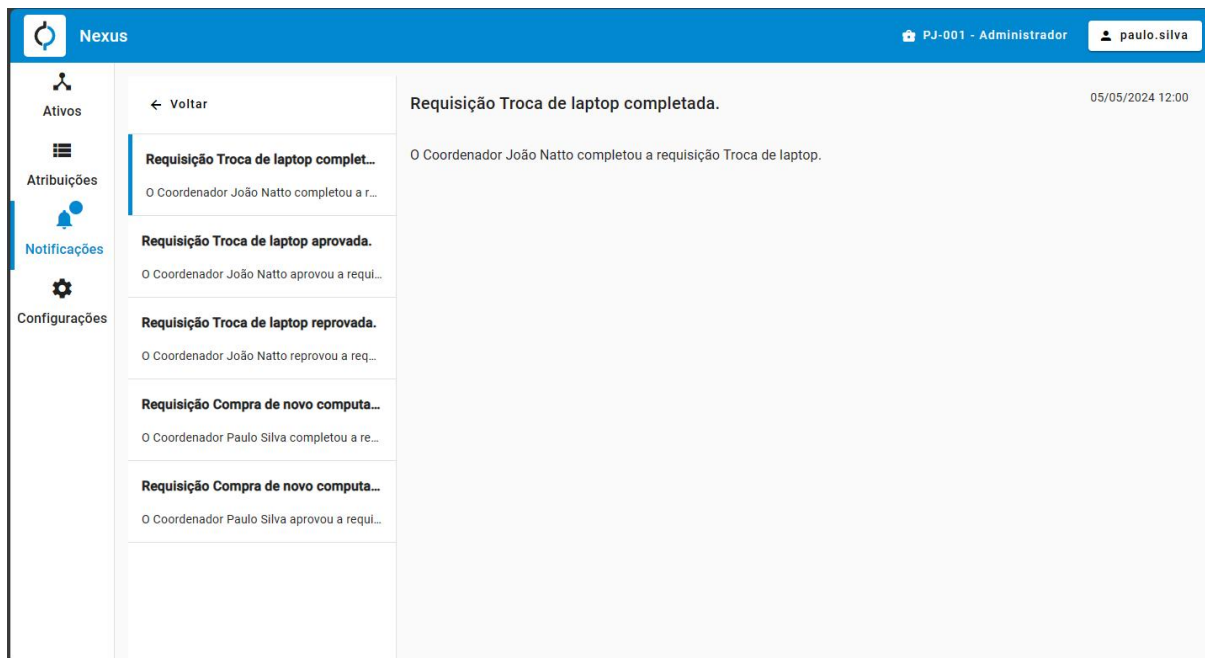
Figura 51: Notificação de requisição aprovada



Fonte: Desenvolvido pelo autor, 2024

Após clicar no botão completar, automaticamente o ciclo de vida acaba e o usuário recebe uma notificação informando a ação, como mostra a Figura 52.

Figura 52: Notificação de conclusão da requisição

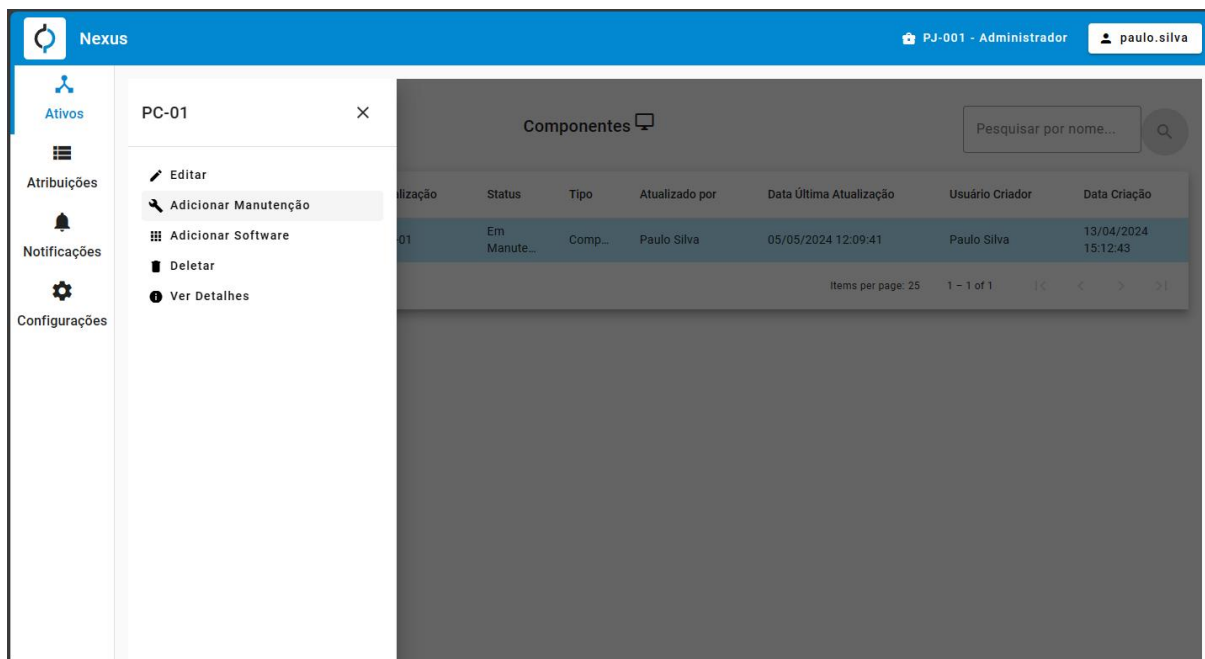


Fonte: Desenvolvido pelo autor, 2024

### 5.13 Verificação de manutenção

Para iniciar o ciclo de vida de verificação de manutenção, é preciso criar uma manutenção por meio da busca de componentes e preencher as informações do formulário, nome e o responsável, e opcionalmente uma descrição, como mostram as imagens 53 e 54:

Figura 53: Ação de adicionar manutenção



Fonte: Desenvolvido pelo autor, 2024

Figura 54: Tela de criação de manutenção

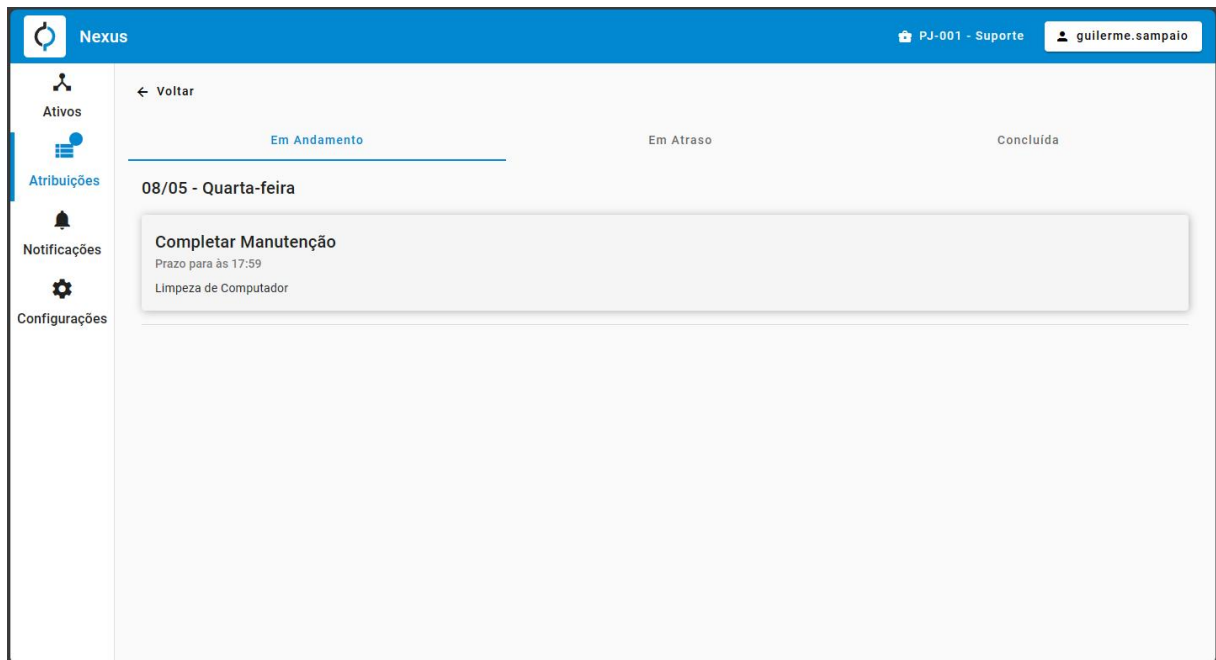
The screenshot shows the 'Adicionar Manutenção' form in the Nexus system. The form fields are as follows:

- Nome\***: Limpeza de Computador
- Descrição**: Prezado, o computador está travando por conta da sujeira. Por favor, limpe-o assim que possível. Abraços.
- Componente\***: PC-01
- Responsável\***:
  - Search input: GUI
  - Selected: Guilherme Sampaio

Fonte: Desenvolvido pelo autor, 2024

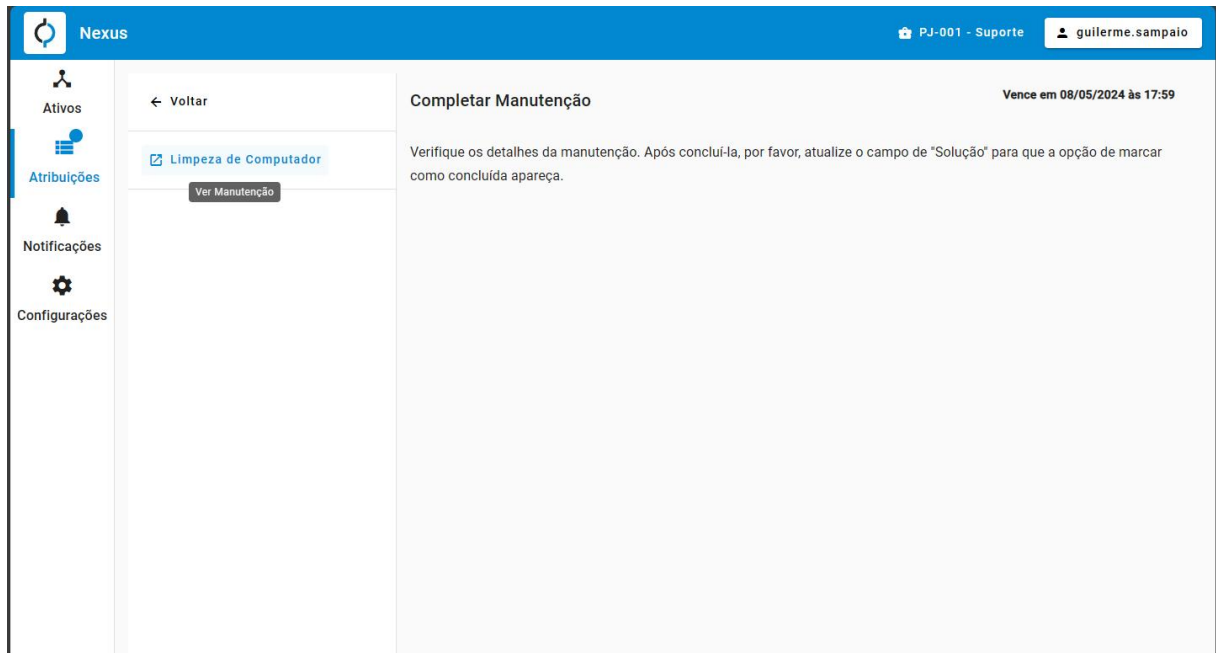
O usuário responsável receberá uma atribuição informando o procedimento e a manutenção para que, após concluí-la, completar a atribuição no sistema, conforme mostram as figuras 55 e 56:

Figura 55: Atribuição de completar manutenção



Fonte: Desenvolvido pelo autor, 2024

Figura 56: Atribuição de completar manutenção sem ações



Fonte: Desenvolvido pelo autor, 2024

É preciso preencher o campo de solução para que a opção de marcar como concluída apareça. Assim, basta clicar na opção de ver manutenção, e atualizar o

campo de solução no formulário e, após isso, a opção estará disponível, como mostram as figuras 57 e 58:

Figura 57: Tela de preenchimento de solução

The screenshot shows the 'Editar Manutenção' (Edit Maintenance) form in the Nexus system. The form is displayed in a light gray box with a white background. It contains the following fields:

- Nome\***: Limpeza de Computador
- Descrição**: Prezado,o computador está travando por conta da sujeira. Por favor, lir
- Componente\***: PC-01
- Responsável\***: Guilherme Sampaio
- Solução\***: Limpeza feita utilizando o kit básico do estoque.

A blue button labeled 'Editar' is located at the bottom right of the form. The interface includes a sidebar on the left with icons for 'Ativos', 'Atribuições', 'Notificações', and 'Configurações'. The top navigation bar shows 'Nexus', 'PJ-001 - Suporte', and the user 'guilherme.sampaio'.

Fonte: Desenvolvido pelo autor, 2024

Figura 58: Ação de completar manutenção

The screenshot shows the 'Completar Manutenção' (Complete Maintenance) screen in the Nexus system. The screen is divided into three main sections:

- Header**: 'Completar Manutenção' and 'Vence em 08/05/2024 às 17:59'.
- Message**: 'Verifique os detalhes da manutenção. Após concluí-la, por favor, atualize o campo de "Solução" para que a opção de marcar como concluída apareça.'
- Task List**: A list of maintenance tasks. The first task is 'Limpeza de Computador', which is highlighted in blue and has a 'Completar' button next to it.

The interface includes a sidebar on the left with icons for 'Ativos', 'Atribuições', 'Notificações', and 'Configurações'. The top navigation bar shows 'Nexus', 'PJ-001 - Suporte', and the user 'guilherme.sampaio'.

Fonte: Desenvolvido pelo autor, 2024

Após completar, o usuário solicitante receberá uma notificação informando a conclusão, e ciclo de vida acaba, de acordo com o que é mostrado na Figura 59:

Figura 59: Notificação de conclusão de manutenção

The screenshot displays the Nexus system interface. The top navigation bar is blue and contains the Nexus logo, the user's role 'PJ-001 - Administrador', and the user's name 'paulo.silva'. The left sidebar has four main sections: 'Ativos', 'Atribuições', 'Notificações', and 'Configurações'. The 'Notificações' section is active, showing a list of notifications. The main content area displays a notification titled 'Manutenção Limpeza de Computador completada.' with a timestamp of '05/05/2024 12:36'. The notification text states: 'O responsável Guilherme Sampaio completou a manutenção Limpeza de Computador do componente PC-01.' Below the notification, there is a list of other notifications, including 'Requisição Troca de laptop completa...', 'Requisição Troca de laptop aprovada.', 'Requisição Troca de laptop reprovada.', 'Requisição Compra de novo computa...', and 'Requisição Compra de novo computa...'.

Fonte: Desenvolvido pelo autor, 2024

## 6 CONSIDERAÇÕES FINAIS

É nítido que após a análise dos resultados obtidos, este trabalho foi capaz de desenvolver um protótipo funcional de um sistema de gerenciamento de ativos, que é capaz de simplificar e aumentar relativamente a produtividade na gestão. Isso pois o sistema tem funcionalidades capazes de aprimorar a comunicação entre indivíduos diferentes e o monitoramento dos ativos. No entanto, por conta das limitações do projeto, o software não seria funcional em um ambiente produtivo.

Os controles de ciclo de vida presentes na aplicação permitem uma agilidade maior para usuários se comunicarem e não perderem prazos estabelecidos pela organização. Além disso, a segregação dos dados por projetos e perfis garante maior segurança à informação da empresa. Com isso, o protótipo criado definitivamente é uma alternativa a processos manuais e pouco confiáveis ou eficientes. Portanto, o sistema possui capacidade de chegar ao patamar de ser funcional em um ambiente de produção, porém exigiria melhorias.

É possível, logo, dar continuidade no projeto com mais recursos e tempo e adaptar o sistema a um ambiente real de uma organização. O trabalho encerra-se criando uma ferramenta capaz de melhorar a gestão de ativos em empresas, se aprimorada, e dando oportunidade para que o desenvolvimento continue por indivíduos interessados no tema. Desse modo, o software NEXUS é capaz de tornar-se uma ótima solução para administração dos ativos nas empresas.

## REFERÊNCIAS

ADA, Ş.; GHAFFARZADEH, M. Decision Making Based On Management Information System and Decision Support System. **European Researcher**, v. 93, n. 4, p. 260–269, 15 mar. 2015. Disponível em: [http://www.erjournal.ru/journals\\_n/1430407576.pdf](http://www.erjournal.ru/journals_n/1430407576.pdf). Acesso em: 28 fev 2024.

**Angular**. Disponível em: <<https://angular.io>>. Acesso em: 27 abr. 2024.

BALARINE, O. F. O. Tecnologia da informação como vantagem competitiva. **Revista de Administração de Empresas**, São Paulo, v. 1, n. 1, p. 1-11, 2002.

**JWT.IO - JSON Web Tokens Introduction**. Disponível em: <<https://jwt.io/introduction>>. Acesso em: 27 abr. 2024.

LAURINDO, F. J.; SHIMIZU, T.; CARVALHO, M. M.; JR, R. R. O papel da tecnologia da informação (TI) na estratégia das organizações. **Gestão & Produção**, São Carlos, v. 8, n. 2, p.160-179, 2001.

MORAES, J. P.; SAGAZ, S. M.; DOS SANTOS, G. L.; LUCIETTO, D. A. TECNOLOGIA DA INFORMAÇÃO, SISTEMAS DE INFORMAÇÕES GERENCIAIS E GESTÃO DO CONHECIMENTO COM VISTAS À CRIAÇÃO DE VANTAGENS COMPETITIVAS: REVISÃO DE LITERATURA. **Revista Visão: Gestão Organizacional**, Caçador (SC), Brasil, v. 7, n. 1, p. 39–51, 2018. DOI: 10.33362/visao.v7i1.1227. Disponível em: <https://periodicos.uniarp.edu.br/index.php/visao/article/view/1227>. Acesso em: 7 jan. 2024.

PARENTE, Thiago Moreira de Carvalho. **Implantação Do Modulo Do Sistema De Inventario Para Hardware E Software De Ativos De TI No TRE-TO**. 56 fl. Monografia (Graduação). Curso de Ciências da Computação. Universidade Federal do Tocantins. Palmas, 2021. Disponível em: <http://hdl.handle.net/11612/3524>. Acesso em: 8 jan. 2024.

PEREZ, MARCELO MONTEIRO; FAMÁ, RUBENS. ATIVOS INTANGÍVEIS E O DESEMPENHO EMPRESARIAL. **Revista Contabilidade & Finanças**, [s. l.], 1 abr. 2006. DOI <https://doi.org/10.1590/S1519-70772006000100002>. Disponível em: <https://www.scielo.br/j/rcf/a/8WztjPbNVnyvNVhJCTJHFDx/#>. Acesso em: 7 jan. 2024.

PINHEIRO, Cassio de Castro. **Gestão da informação e conhecimento: proposta de uso de um sistema de informação para controle da configuração de ativos de ti**. 2012. Monografias de Especialização (Especialização em Gestão Estratégica da Informação) - Universidade Federal de Minas Gerais, [S. l.], 2012. Disponível em: <https://repositorio.ufmg.br/handle/1843/BUOS-98NH7E>. Acesso em: 7 jan. 2024.

RAY, M.; WEST, R. **O que é o SQL Server? - SQL Server**. Disponível em: <https://learn.microsoft.com/pt-br/sql/sql-server/what-is-sql-server?view=sql-server-ver16>>. Acesso em: 27 abr. 2024.

RIBEIRO, J. S. de A. N.; ZIVIANI, F.; TADEU, H. F. B.; NEVES, J. T. de R. Gestão do conhecimento e sistemas de informação na cadeia de suprimentos global. **Revista Brasileira de Biblioteconomia e Documentação**, [S. l.], v. 15, n. 2, p. 251–289, 2019. Disponível em: <https://rbbd.febab.org.br/rbbd/article/view/1203>. Acesso em: 7 jan. 2024.

SILVA, S.; PINTO, J. ANÁLISE DA IMPORTÂNCIA DA GESTÃO DE ATIVOS DE TI NO AMBIENTE DE MICRO E PEQUENAS EMPRESAS. **Revista Científica e-Locução**, v. 1, n. 15, p. 18, 17 jul. 2019.

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

STONE, M. et al. **IT asset management: financial services**. Gaithersburg, MD: National Institute of Standards and Technology, 2018. Disponível em: <http://doi.org/10.6028/NIST.SP.1800-5>. Acesso em: 24 fev. 2024.

SUCKOW, Giuliano. **Gerência de ativos de TI nas organizações públicas**. 2011. 62 f. Trabalho de Conclusão de Curso (Especialização) – Universidade Tecnológica Federal do Paraná, Curitiba, 2011. Disponível em: <http://repositorio.utfpr.edu.br/jspui/handle/1/20015>. Acesso em: 7 jan. 2024.

TANEMBAUM, A.; WETHERALL, D. **Redes de Computadores 5ª Edicao**. [S.l.]: Pearson, 2011.

UBAH, K. **Learn Web Development Basics – HTML, CSS, and JavaScript Explained for Beginners**. Disponível em: <https://www.freecodecamp.org/news/html-css-and-javascript-explained-for-beginners/>. Acesso em: 27 abr. 2024.

YUZRAAN. **BPM Tutorial | Modelação de Processos de Negócios Guia (Incluir metodologias)**. Disponível em: <https://creately.com/blog/pt/diagrama/tutorial-de-modelacao-de-processos-de-negocios/>. Acesso em: 27 abr. 2024.