

UNIVERSIDADE FEDERAL DO PARANÁ

LEONARDO ROBERTO ROCHA

LUIZ HENRIQUE DOMINICO

MARCO AURELIO VIEIRA OLIVETTE

MATEUS MENDES ZANELA

PATRICK MARCELO DZIURA

CALL2R - GERENCIADOR DE CHAMADOS UTILIZANDO ITIL

CURITIBA

2020

LEONARDO ROBERTO ROCHA
LUIZ HENRIQUE DOMINICO
MARCO AURELIO VIEIRA OLIVETTE
MATEUS MENDES ZANELA
PATRICK MARCELO DZIURA

CALL2R - GERENCIADOR DE CHAMADOS UTILIZANDO ITIL

Trabalho de conclusão de curso apresentado como requisito parcial para obtenção do grau de Tecnólogo em Análise e Desenvolvimento de Sistemas, do Setor de Educação Profissional e Tecnológica, da Universidade Federal do Paraná.

Orientador: Prof. Dr. Razer A. N. R. Montaña.

CURITIBA

2020



UNIVERSIDADE FEDERAL DO PARANÁ

ATA DE REUNIÃO
TERMO DE APROVAÇÃO

Luiz Henrique Dominico
Marco Aurelio Vieira Olivette
Mateus Mendes Zanela
Patrick Dziura
Leonardo Roberto Rocha

CALL2R - GERENCIADOR DE CHAMADOS UTILIZANDO ITIL

Monografia aprovada como requisito parcial à obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas, do Setor de Educação Profissional e Tecnológica da Universidade Federal do Paraná.

Prof. Razer Anthom Nizer Rojas Montaña

Orientador – SEPT/UFPR

Prof. Jaime Wojciechowski

SEPT/UFPR

Profa. Raíscia Mantovani Fontana

SEPT/UFPR

Curitiba, 22 de Outubro de 2020.

RESUMO

Atualmente é crescente o número de pessoas e empresas que contratam serviços dos mais diversos fornecedores, e com essas contratações acaba sendo necessário todo um processo de comunicação para se resolver eventuais problemas. Porém, cada um dos fornecedores possui um diferente canal de comunicação para se tratar desses incidentes e problemas, já que atualmente não há uma padronização a ser seguida. Então nesses cenários, com o objetivo de centralizar esse processo de comunicação e resolução de chamados, utilizando os conceitos do ITIL, é que nasce a plataforma Call2r. Este trabalho propõe uma plataforma *service desk* presente no ambiente *web* ou *mobile* que apresenta funcionalidades que visam facilitar e aproximar os dois interessados dessa interação, no caso o fornecedor e o usuário, evitando efeitos negativos e aumentando a velocidade e produtividade de todo o processo. A aplicação foi desenvolvida em três partes: *backend* como API utilizando PHP, *frontend* utilizando TypeScript e React e a aplicação *mobile* utilizando React. O *backend* da aplicação está na nuvem para que tanto a aplicação *mobile* quanto o *frontend* utilizem ela.

Palavras-chave: Gerenciador. Chamados. ITIL. Empresas.

ABSTRACT

Nowadays, the number of people interested in hiring a service from a provider is increasing and with this hiring process ends up being necessary a communication process to solve eventual problems. However usually each provider has a different communication channel to solve these incidents and problems, since that actually there is not a pattern to be followed. So in these scenarios with the goal to center this communication process and solve some calls, using the ITIL concepts, is where the platform Call2r was born. Service desk platform present at the web and mobile environment which features functions that aim to facility and to approach the two interested of this interaction, who are the provider and the client, avoiding negative effects and increasing the speed and productivity of all the process. The application was developed in three parts: backend like an API using PHP, frontend using TypeScript and React and a mobile application using React. The application's backend is on the cloud so that both the mobile application and the frontend use it.

Keywords: Manager. Tasks. ITIL. Companies.

LISTA DE FIGURAS

2.1	CICLO PDCA	20
2.2	GERENCIAMENTO DE INCIDENTES	21
2.3	NÍVEIS DE ATENDIMENTO.	22
2.4	NÍVEIS DE ATENDIMENTO.	23
2.5	USO DO PHP EM WEBSITES	26
2.6	EXEMPLO DE APLICAÇÃO NORMAL	29
2.7	EXEMPLO DE APLICAÇÃO COM DOCKER	29
2.8	EXEMPLO DE APLICAÇÃO DO MÉTODO SCRUM	31
2.9	EXEMPLO DE ARQUITETURA HEXAGONAL	33
3.1	FUNCIONAMENTO DE BRANCHES NO GITHUB	38
3.2	KANBAN UTILIZANDO GIT ISSUES	39
3.3	LISTA DE PULL REQUESTS	39
3.4	WBS PRIMEIRA ETAPA.	44
3.5	WBS SEGUNDA ETAPA	45
3.6	DIAGRAMA DE GANTT.	46
3.7	DIAGRAMA DE GANTT DA SEGUNDA ETAPA.	47
4.1	ARQUITETURA BACKEND	55
4.2	ARQUITETURA FRONTEND	56
4.3	TELA DE LOGIN	57
4.4	TELA DE RECUPERAÇÃO DE SENHA	58
4.5	TELA DE LISTA DE CHAMADOS.	59
4.6	TELA DE CRIAÇÃO DE CHAMADOS	59
4.7	TELA DE VISUALIZAÇÃO DE CHAMADO	60
4.8	TELA DE ENCAMINHAMENTO PARA CLIENTE.	61
4.9	TELA DE RESPOSTA DO CLIENTE	61
4.10	TELA DE TRANSFERÊNCIA DE CHAMADO	62
4.11	TELA DE RESOLUÇÃO DE CHAMADO	63

4.12	TELA DE APROVAÇÃO DE CHAMADO	63
4.13	TELA DE REPROVAÇÃO DE CHAMADO	64
4.14	TELA DE CANCELAMENTO	65
4.15	TELA DE EXPORTAR RELATÓRIOS	65
4.16	TELA DE PERFIL	66
4.17	TELA DE ALTERAÇÃO DE SENHA.	67
4.18	TELA DE LISTA DE GERENTES	67
4.19	TELA DE LISTA DE USUÁRIOS	68
4.20	TELA DE CRIAÇÃO DE USUÁRIO POR ADMINISTRADOR.	68
4.21	TELA DE CRIAÇÃO DE USUÁRIO POR GERENTE	69
4.22	TELA DE EDIÇÃO DE USUÁRIO	69
4.23	TELA DE WIKI	70
4.24	TELA DE LISTA DE ARTIGOS	71
4.25	TELA DE EDIÇÃO DE ARTIGO.	71
4.26	TELA DE EDIÇÃO DE EMPRESA	72
4.27	TELA DE LISTA DE EMPRESAS	73
4.28	TELA DE LOGIN MOBILE	74
4.29	TELA DE LISTA DE CHAMADOS MOBILE	75
4.30	TELA DE VISUALIZAR CHAMADO MOBILE	76
4.31	TELA DE FILTROS MOBILE	77
A.1	DIAGRAMA DE CASOS DE USO.	86
A.2	DIAGRAMA DE CASOS DE USO MOBILE.	87
B.1	TELA DE LOGIN	88
B.2	TELA DE CONFIGURAÇÃO DE CHAMADOS	90
B.3	TELA DE EDIÇÃO DE GERENTE	93
B.4	TELA DE LISTA DE GERENTES	93
B.5	TELA DE EDIÇÃO DE GERENTE	96
B.6	TELA DE LISTA DE GERENTES	96
B.7	TELA DE EDIÇÃO DE ÁREAS	99
B.8	TELA DE EDIÇÃO DA EMPRESA MÃE	101
B.9	TELA DE EDIÇÃO DA EMPRESA SUPORTE	103

B.10	TELA DE LISTA DE EMPRESAS SUPORTE	103
B.11	TELA DE LISTA DE CHAMADOS.	106
B.12	TELA DE LISTA DE CHAMADOS.	108
B.13	TELA DE LISTA DE CHAMADOS.	110
B.14	TELA DE LISTA DE CHAMADOS.	112
B.15	TELA DE ABERTURA DE NOVO CHAMADO	114
B.16	TELA DE APROVAÇÃO DE CHAMADO	116
B.17	TELA DE REPROVAÇÃO DE CHAMADO	117
B.18	TELA DE RESOLUÇÃO DE CHAMADO	118
B.19	TELA DE ATENDIMENTO DE CHAMADO	119
B.20	TELA DE DETALHES DO CHAMADO	120
B.21	TELA DE TRANSFERÊNCIA DE CHAMADO	121
B.22	TELA DE PESQUISA DE CHAMADOS.	123
B.23	TELA DE CANCELAMENTO DE CHAMADOS	124
B.24	TELA COM OPÇÕES DE FILTROS.	126
B.25	TELA PARA EXPORTAR RELATÓRIO	128
B.26	TELA PARA PRIORIZAR CHAMADO.	130
B.27	TELA DA WIKI.	132
B.28	TELA DE LISTA DE EDIÇÃO DA WIKI	132
B.29	TELA DE DETALHES DE EDIÇÃO DA WIKI	133
B.30	TELA DE ALTERAÇÃO DE SENHA.	136
B.31	TELA DE EDIÇÃO DE USUÁRIO	138
B.32	TELA DE LISTA DE USUÁRIOS	138
B.33	TELA DE EDIÇÃO DE USUÁRIO	141
B.34	TELA DE LISTA DE USUÁRIOS	141
B.35	TELA ALTERAR PERFIL	144
B.36	TELA DE ALTERAÇÃO DE PERFIL	145
B.37	TELA DE ALTERAÇÃO DE PERFIL	147
B.38	TELA DE ESQUECIMENTO DA SENHA	149
B.39	TELA DE ADIÇÃO DE LOG.	151
B.40	TELA DE EDIÇÃO DE CATEGORIAS	153

C.1	DIAGRAMA DE CLASSES	155
D.1	CLASSES DE IMPLEMENTAÇÃO	156
D.2	CLASSES DE IMPLEMENTAÇÃO DE AÇÕES	157
D.3	CLASSES DE IMPLEMENTAÇÃO DE COMANDOS	158
D.4	CLASSES DE IMPLEMENTAÇÃO DOCTRINE	159
D.5	CLASSES DE IMPLEMENTAÇÃO DE ENTIDADES	159
D.6	CLASSES DE IMPLEMENTAÇÃO DE PESQUISAS	160
D.7	CLASSES DE IMPLEMENTAÇÃO DE REPOSITÓRIOS	161
D.8	CLASSES DE IMPLEMENTAÇÃO DE SERVIÇOS	162
E.1	CRIAR EMPRESA	163
E.2	BUSCAR TODAS AS EMPRESAS	163
E.3	BUSCAR EMPRESA MÃE	164
E.4	BUSCAR EMPRESA POR ID	164
E.5	BUSCAR EMPRESA POR ÁREA	165
E.6	ATUALIZAR EMPRESA	165
E.7	BUSCAR TODAS AS ÁREAS	166
E.8	BUSCAR CATEGORIAS POR EMPRESA	166
E.9	ADICIONAR LOG AO CHAMADO	167
E.10	APROVAR CHAMADO	167
E.11	BUSCAR CHAMADOS	168
E.12	BUSCAR CHAMADOS POR ID	168
E.13	CRIAR CHAMADO	169
E.14	EXPORTAR CHAMADOS PARA PDF.	169
E.15	MOVER CHAMADO PARA AGUARDANDO RESPOSTA.	170
E.16	MOVER CHAMADO PARA CANCELADO	170
E.17	MOVER CHAMADO PARA EM ATENDIMENTO	171
E.18	REPROVAR CHAMADO	171
E.19	SUBMETER CHAMADO PARA APROVAÇÃO	172
E.20	TRANSFERIR CHAMADO PARA OUTRA EMPRESA	173
E.21	ATUALIZAR CHAMADO.	174
E.22	ATUALIZAR IMAGEM	174

E.23	ATUALIZAR USUÁRIO	175
E.24	BUSCAR USUÁRIOS POR CARGO	175
E.25	BUSCAR USUÁRIOS POR ID	176
E.26	CRIAR USUÁRIO	176
E.27	LOGIN.	177
E.28	RESETAR SENHA	177
E.29	ATUALIZAR ARTIGO	178
E.30	BUSCAR ARTIGOS POR EMPRESA.	178
E.31	BUSCAR ARTIGOS POR ID	178
E.32	CRIAR ARTIGO	179
E.33	DELETAR ARTIGO	179
F.1	DIAGRAMA DE ESTADOS	180
G.1	DIAGRAMA DE ENTIDADE RELACIONAMENTO	181
H.1	FLUXOGRAMA DO CHAMADO	182

LISTA DE QUADROS

3.1	DIVISÃO DE RESPONSABILIDADES	47
-----	--	----

LISTA DE ACRÔNIMOS

API	Application Programming Interface
AWS	Amazon Web Services
CCTA	Central Communications and Telecom Agency
COBIT	Control Objectives for Information and Related Technologies
CPF	Cadastro de Pessoas Físicas
CSS	Cascading Style Sheets
DevOps	Development and Operations
EC2	Elastic Compute Cloud
HDD	Hard Disk Drive
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
IDE	Ambiente de Desenvolvimento Integrado
IEEE	Instituto de Engenheiros Eletricistas e Eletrônicos
ITIL	Information Technology Infrastructure Library
JDK	Java Development Kit
JEE	Java Enterprise Edition
JPA	Java Persistence API
JRE	Java Runtime Environment
JS	JavaScript
JSON	JavaScript Object Notation
JVM	Java Virtual Machine
JWT	JSON Web Token
OGC	Office Government Commerce
PDCA	Plan, do, control, act
S3	Simple Storage Service
SEPT	Setor de Educação Profissional e Tecnológica
SLA	Service Level Agreement
SSD	Solid State Drive
TADS	Tecnologia em Análise e Desenvolvimento de Sistemas
TCP	Transmission Control Protocol
TI	Tecnologia da Informação
UDP	User Datagram Protocol
UFPR	Universidade Federal do Paraná
UML	Unified Modeling Language
WBS	Work Breakdown Structure

SUMÁRIO

1	INTRODUÇÃO	15
1.1	JUSTIFICATIVA	16
1.2	OBJETIVOS	17
1.2.1	Objetivo Geral	17
1.2.2	Objetivos Específicos	17
1.3	ESTRUTURA DO DOCUMENTO	17
2	FUNDAMENTAÇÃO TEÓRICA	19
2.1	FUNDAMENTAÇÃO TEÓRICA DO NEGÓCIO	19
2.1.1	Governança de TI	19
2.1.2	ITIL	20
2.1.3	Softwares semelhantes	23
2.2	FUNDAMENTAÇÃO TEÓRICA DAS TECNOLOGIAS	24
2.2.1	Unified Modeling Language UML	24
2.2.2	PHP	25
2.2.3	Symfony	26
2.2.4	React	27
2.2.5	TypeScript	27
2.2.6	MySQL	28
2.2.7	Docker	28
2.2.8	Métodos Ágeis	30
2.2.9	Git	32
2.2.10	Arquitetura Hexagonal	32
2.2.11	S.O.L.I.D.	33
2.2.12	Linguagem Ubíqua	34
3	METODOLOGIA	36
3.1	MATERIAIS E MÉTODOS	36
3.1.1	Astah	36
3.1.2	GanttProject	36

3.1.3	Trello	37
3.1.4	Adobe XD	37
3.1.5	Docker	37
3.1.6	GitHub	37
3.1.7	Amazon AWS	40
3.1.8	Nginx	40
3.1.9	Infraestrutura de Desenvolvimento	41
3.2	MODELO DE PROCESSO DE ENGENHARIA DE SOFTWARE	42
3.2.1	Adaptação Scrum	43
3.3	PLANO DE ATIVIDADES	43
3.3.1	WBS	43
3.3.2	Gantt	46
3.4	RESPONSABILIDADES.	47
3.5	DESENVOLVIMENTO DO PROJETO.	48
3.5.1	Sprint 1	49
3.5.2	Sprint 2	49
3.5.3	Sprint 3	49
3.5.4	Sprint 4	49
3.5.5	Sprint 5	49
3.5.6	Sprint 6	50
3.5.7	Sprint 7	50
3.5.8	Sprint 8	50
3.5.9	Sprint 9	50
3.5.10	Sprint 10.	51
3.5.11	Período sem sprints	51
3.5.12	Sprint 11.	51
3.5.13	Sprint 12.	51
3.5.14	Sprint 13.	52
3.5.15	Sprint 14.	52
3.5.16	Sprint 15.	52
3.5.17	Sprint 16.	52

3.5.18	Sprint 17.	52
3.5.19	Sprint 18.	52
3.5.20	Sprint 19.	53
3.5.21	Sprint 20.	53
3.5.22	Sprint 21.	53
3.5.23	Sprint 22.	53
4	APRESENTAÇÃO DO SOFTWARE	54
4.1	ARQUITETURA DO SOFTWARE.	54
4.1.1	Arquitetura Backend.	54
4.1.2	Arquitetura Frontend	55
4.1.3	Arquitetura Mobile.	56
4.2	SOFTWARE.	56
4.2.1	Sistema Web	57
4.2.2	Sistema Mobile	73
4.3	CONSIDERAÇÕES À APRESENTAÇÃO DO SOFTWARE.	78
5	CONSIDERAÇÕES FINAIS.	79
5.1	RECOMENDAÇÕES PARA TRABALHOS FUTUROS	79
	APÊNDICE A – DIAGRAMA DE CASOS DE USO	86
	APÊNDICE B – ESPECIFICAÇÕES DE CASOS DE USO	88
	APÊNDICE C – DIAGRAMA DE CLASSES	155
	APÊNDICE D – DIAGRAMAS DE CLASSES DE IMPLEMENTAÇÃO	156
	APÊNDICE E – DIAGRAMAS DE SEQUÊNCIA	163
	APÊNDICE F – DIAGRAMA DE ESTADOS	180
	APÊNDICE G – DIAGRAMA DE ENTIDADE RELACIONAMENTO	181
	APÊNDICE H – FLUXOGRAMA	182

1 INTRODUÇÃO

O cenário atual do mundo de tecnologia permitiu, através de uma constante evolução, que empresas dos mais diversos ramos de atuação construíssem relações econômicas externas. Tais relações, após fixadas, permitiram que não somente produtos e insumos fossem comercializados, mas também que serviços dos mais diferentes tipos fossem disponibilizados a quem tivesse interesse.

O estado da demanda atual é crescente no segmento de prestação de serviços especializados. A Pesquisa Anual de Serviços apurou a existência de 1.286.621 empresas que tem como principal atividade a prestação de serviços não financeiros (IBGE, 2015). Observou-se que técnicas e boas práticas foram criadas para melhor atendimento dos clientes, buscando melhorias de condições, principalmente no cenário competitivo.

Com essas inúmeras mudanças de paradigma e com o crescimento exacerbado, uma série de falhas começaram a surgir no âmbito da prestação de serviço. As causas raízes são diversas, as quais muitas vezes ocorrem de ações dos funcionários em relação aos problemas no sistema (BITNER; BOOMS; TETREULT, 1990). O acúmulo de falhas e falta de comunicação para a resolução desses problemas, resultam em um número consideravelmente grande de reclamações e insatisfações. Segundo Lovelock e Wright (2006) o sentimento resultante da comparação entre o esperado e o obtido, ocasiona efeitos negativos em algumas empresas prestadoras de serviços, como por exemplo afetando sua credibilidade, que mediante ao mercado, pode ser manchada.

Com tantos efeitos atuando sobre a área de TI, mas sabendo a importância desta para os negócios, a proposta apresentada pela biblioteca ITIL torna-se extremamente relevante: um conjunto de boas práticas e processos que visam criar um vínculo forte entre a área de tecnologia e negócio (FILHO, 2012).

Quando um consumidor (nesse caso podendo ser um cliente, usuário final, ou até mesmo quando uma outra companhia) deseja realizar alguma reclamação ou crítica, observa-se que os usuários realizaram 43% de suas reclamações entrando em contato diretamente com a empresa responsável pela prestação do serviço, segundo um estudo do Simpósio de Excelência em Gestão e Tecnologia (SANTOS; DA COSTA; SANDER, 2008). Este indica claramente a forma como esse contato é feito e abordado. Logo, visto essa demanda existente para atender a grande fatia de mercado que busca essa comunicação, foram criados softwares com os mais variados objetivos e com as mais variadas funcionalidades que visam aproximar o cliente do seu prestador quando o assunto tratado é uma reclamação ou injúria.

Os softwares em questão são chamados de softwares de *help desk*, e atuam no primeiro nível de contato com o cliente. Quando tais softwares são trabalhados de uma maneira que o técnico receba e resolva os indecentes que o cliente constatou, podem receber a nomenclatura

de softwares *Service Desk*. Esse termo se refere a uma ponte de contato entre o prestador e o usuário, que tem como foco a comunicação de incidentes (OGC, 2001). O usuário pode até mesmo ser de outras empresas, o qual necessita de um serviço específico.

O objeto desse trabalho é o sistema Call2r. Ele foi construído com base na biblioteca ITIL para conter um processo seguro, além de ser construído para conter interfaces simples e um fluxo direto, facilitando assim a interação do usuário.

Os detalhes de construção são apresentados de uma maneira moderna, abstraindo dificuldades e fornecendo ferramentas como: controle do fluxo de chamados, gestão de empresas fornecedoras e gestão de usuários responsáveis.

1.1 JUSTIFICATIVA

Visto o cenário em que as empresas de prestação de serviços e seus clientes se encontram, sistemas que atuem diretamente no problema citado são necessários. Eles precisam exercer o papel principal no gerenciamento do controle focado nesses atendimentos e na forma como eles ocorrem. Desta forma, é possível aproximar as partes interessadas e diminuir desgastes, tanto por parte do usuário quanto por do prestador de serviço.

Quando se fala dos desgastes, na parte do cliente é possível citar problemas relacionados à lentidão na resolução de chamados, feitas através de métodos tradicionais como ligações, falta ou distância de contato entre os interessados e falta de notificação do andamento das ações e medidas que estão sendo tomadas (MOVIDESK, 2017). Já quando se refere ao prestador de serviço, há problemas relacionados com o atraso na resolução decorrente de problemas de comunicação, falta de controle interno do que tem que ser resolvido e falta de redirecionamento quando o problema não pode ser resolvido naquele ambiente (MOVIDESK, 2017).

Existem dois tipos de softwares atuais no mercado. O primeiro deles são os softwares pagos, que possuem boas funcionalidades porém tem um preço de licença muito alto, e os softwares gratuitos, que não atendem o usuário da maneira satisfatória. Geralmente, ambos costumam pecar em excessos de funcionalidades, gerando confusão, assim como não acrescentar pontos positivos para sanar os desgastes citados anteriormente. Isso acarreta ainda mais críticas negativas, uma vez que os consumidores se tornaram muito mais opinativos em relação aos produtos e serviços que consomem. Isso se deve a ascensão do acesso à informação e do crescimento da comunicação mundial.

Outro ponto que pode ser abordado é relacionado ao processo de tratamento de requerimentos realizado por uma das diversas empresas que prestam serviços. Esse processo pode ser realizado de várias maneiras, porém nem todas as empresas tem a maturidade necessária para atingir todas as metas e satisfazer seus clientes, por conta da dificuldade de implantação do ITIL, que segundo Pereira e Silva (2010) e Marrone e Kolbe (2011) é motivada pela falta de definição clara das etapas de execução.

Para isso softwares de auxílio são necessários, principalmente para se iniciar esse processo de maturidade através da implementação de diversos conceitos desenvolvidos, e então implementados por empresas referência no seu setor de atuação.

Nesse contexto é onde se encontra a necessidade do Call2r, pois ele foi construído com premissa na resolução desses problemas.

Sendo intuitivo e uma ótima introdução ao ITIL, o que pode beneficiar os interessados de várias maneiras, oferecendo suas ferramentas para economizar recursos, como o tempo.

1.2 OBJETIVOS

Dados os problemas e justificativa apresentados, nas próximas seções são apresentados os objetivos do projeto.

1.2.1 Objetivo Geral

Tem-se como objetivo geral, o planejamento e a criação de uma plataforma *web* e *mobile* que visa auxiliar na resolução e registro de incidentes e resoluções entre cliente e seus prestadores de serviço.

1.2.2 Objetivos Específicos

Os objetivos de negócio do projeto em questão, são:

- Desenvolver uma aplicação *web* para criação, gerenciamento e resolução de chamados;
- Desenvolver uma aplicação *mobile* pequena para facilitar a visualização de chamados;

1.3 ESTRUTURA DO DOCUMENTO

O documento em questão está condicionado e formatado dentro dos moldes necessários, possuindo suas informações e desenvolvimento distribuídos em capítulos com temas específicos. A descrição de cada capítulo será detalhada a seguir.

O primeiro capítulo, ou seja, o capítulo em questão, tem como objetivo principal situar, contextualizar e introduzir o leitor aos temas e conceitos que serão abordados no decorrer do documento.

O segundo capítulo tem como principal foco, descrever e fundamentar todos os assuntos que dão base ao sistema e tem caráter fundamental para o entendimento restante. Relacionado aos assuntos relativos ao negócio, podemos citar exemplos como a Governança de Tecnologia da Informação e a biblioteca ITIL. Para tornar o negócio funcional é necessário a abordagem de algumas tecnologias, que foram descritas e utilizadas durante todo o período do desenvolvimento do projeto.

O terceiro capítulo possui o foco em abordar a metodologia utilizada para se realizar o trabalho, assim como suas variações e adaptações, apresentando as tecnologias abordadas e a forma de uso delas, justificando também o porquê da sua escolha.

O quarto capítulo busca apresentar em formato de caixa branca o planejamento e funcionamento da arquitetura do software, além de apresentar o software em si com suas principais funcionalidades e características, abordando e apresentando ao leitor figuras relativas ao estado do programa desenvolvido.

Por fim o quinto capítulo tem seu foco voltado para a conclusão e considerações após o término do que foi desenvolvido durante o processo produtivo. Apresentando tudo aquilo realizado e aquilo que porventura foi inviabilizado e suas respectivas explicações, além de apresentar possibilidades para o futuro do projeto.

No fim do documento estão presentes todos os artefatos que foram pensados e construídos durante o desenvolvimento, com o objetivo de facilitar o entendimento e guiar todo o processo.

2 FUNDAMENTAÇÃO TEÓRICA

A fundamentação teórica tem como foco e objetivo específico contemplar e apresentar com caráter introdutório, todos os temas relevantes referentes ao projeto desenvolvido. O atual capítulo será dividido em Fundamentação Teórica de Negócio e Fundamentação Teórica das Tecnologias.

A fundamentação de negócio irá abordar temas que contextualizam e direcionam a área de atuação do sistema, implicando até mesmo em sua concepção algumas regras de negócio. A fundamentação tecnológica tratará de maneira sucinta sobre temas relativos às tecnologias utilizadas para se realizar o desenvolvimento do projeto.

2.1 FUNDAMENTAÇÃO TEÓRICA DO NEGÓCIO

Quando se trata de negócio, tudo que está descrito neste documento é referente aos temas que constroem o negócio propriamente dito, como no caso, as diretrizes da Governança de TI.

2.1.1 Governança de TI

A Governança de TI surgiu a partir da necessidade em um período trágico, sendo ela a promessa de solução de diversos problemas, falhas e mal gerenciamento. É definida e estruturada de maneira a harmonizar as atividades de TI e todos seus processos envolvidos com os objetivos estratégicos da corporação. Segundo Gasetta (2012, p.5), há uma busca por alinhar o setor de TI ao negócio, combinando equilíbrio entre tecnologia, negócio e gestão de TI, perseguindo uma contribuição efetiva dessa área na companhia.

A implantação dessa estrutura ganhou muito espaço depois que foi avaliado que o setor de TI possui muito a entregar para com os objetivos da empresa, uma vez que deixou de lado a visão que se tinha de ser apenas um setor que acarretava em despesas. Esta mudança veio quando a TI, por si só ganhou muito poder no cenário atual de competitividade do mundo dinâmico dos negócios, uma vez que é possível se avaliar o desenvolvimento dessa área através de mensurações (ATLASSIAN, 2020).

Essa base traz consigo todo um pacote de concepções que devem ser colocadas em pauta e discutidas no ambiente organizacional. Por exemplo: a garantia do alinhamento estratégico, a entrega real de valor do setor de Tecnologia da Informação, e melhores gerenciamentos, como o de risco, recurso e desempenho (GASETA, 2012). Tudo isso para que outros benefícios já citados sejam vistos, como: o fornecimento de uma visão integrada e completa do ambiente de TI, permissão de tomadas de decisões mais efetivas e eficientes, avaliação objetiva de maturidade, etc.

A composição da Governança pode ser implementada de maneira sistêmica, após alguns passos: montar e confeccionar o perfil da sua empresa e clientes; ter uma noção completa e total dos procedimentos existentes e recorrentes dentro da organização; ser identificado as pessoas que farão parte desse projeto de implementação; criar maneiras e métricas para se acompanhar todos os processos que foram identificados previamente (GASETA, 2012).

Com mudanças nas empresas e início da informatização na década de 1980 e 1990, foi necessária a criação de metodologias para a organização desta mudança. Por causa disso foram criados o ITIL, que será apresentado na próxima seção, e o COBIT. O COBIT é um *framework* para auxiliar as empresas a atingirem objetivos na governança de TI e é utilizado mundialmente por empresas com responsabilidades em processos e tecnologias. Com seu uso, é possível obter qualidade, confiabilidade e controle das informações e tecnologia relacionada (ISACA, 2019).

2.1.2 ITIL

Foi concebido no final da década de 1980 pela OGC (*Office Government Commerce*), antiga CCTA (*Central Communications and Telecom Agency*) no Reino Unido (FILHO, 2012). O ITIL é uma biblioteca que contém processos e práticas de prestação e gerenciamento de serviços de TI, com foco no cliente. Com a sua implementação, permite-se criar e manter um vínculo forte e duradouro entre a tecnologia e negócio, por meio da qualidade total dos serviços. Este vínculo possui um direcionamento em descrever certos processos e alinhar os serviços, planejamento e suporte de TI com o melhoramento contínuo.

Figura 2.1: CICLO PDCA



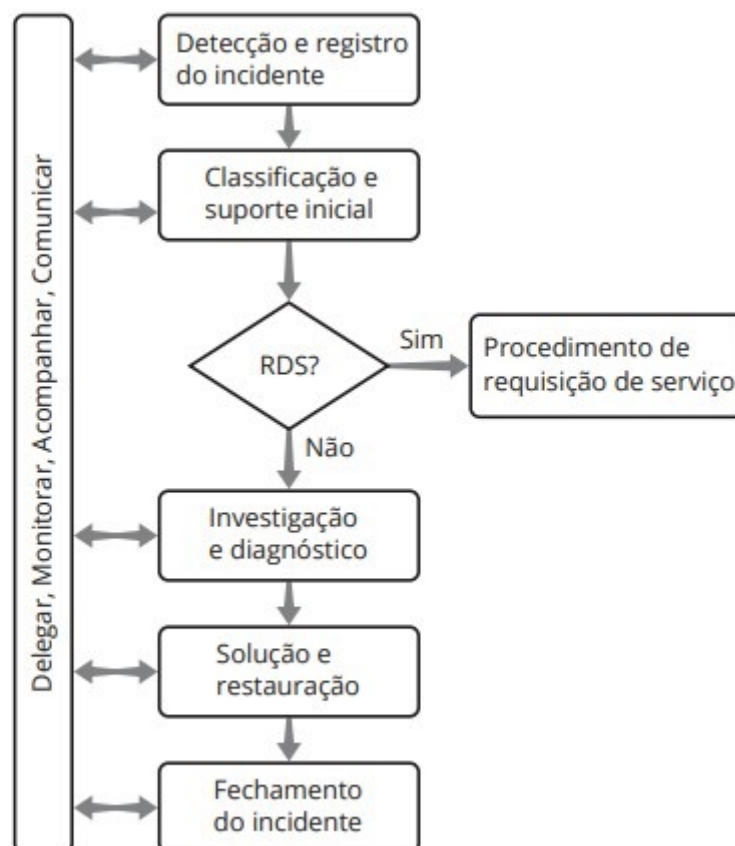
FONTE: Periard (2011)

O ITIL tem seu funcionamento e arquitetura baseado no Ciclo de Vida de Serviço, que como citado previamente deve ter uma ligação com os negócios da empresa em questão. O ciclo de vida, segundo Filho (2012, p.5), se inicia com a definição das políticas que serão seguidas e

com a determinação dos objetivos que são almejados, seguindo para as fases de planejamento e desenvolvimento. Logo vem a contemplação de ajustes e mudanças, entrega e operações cotidianas de suporte ao que foi entregue. Por fim, englobando todos os outros passos do ciclo tem a parte de aprendizado e melhorias, que busca verificar os acontecimentos de cada fase anterior e identificar os problemas para futuras melhorias e evoluções na próxima execução do ciclo, se utilizando o ciclo PDCA como ferramenta de apoio.

Dentro do escopo do que é ITIL, há a definição de gerenciamentos, que são basicamente sugestões de como deve ser implementado e controlado algum determinado segmento dentro do escopo da biblioteca. Um gerenciamento muito relevante para o entendimento do projeto em si, é o gerenciamento de incidentes que aborda e trabalha com todos os incidentes que ocorreram e foram observados por ferramentas de monitoramento e/ou usuários (FILHO, 2012). Busca-se então sanar e restaurar o serviço de maneira rápida e prática sempre mantendo uma linha de comunicação com o cliente interessado, atuando muitas vezes como uma solução de contorno e não com uma solução permanente, solução essa buscada no gerenciamento de problemas.

Figura 2.2: GERENCIAMENTO DE INCIDENTES



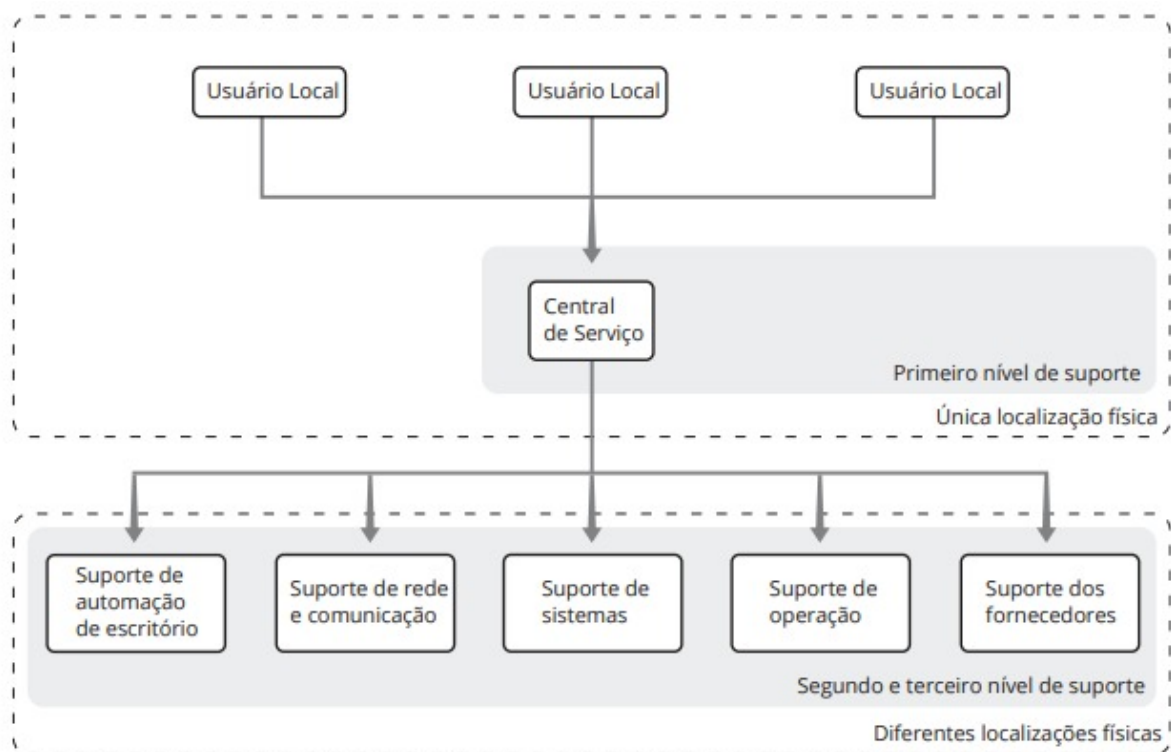
FONTE: FILHO, 2012

A Figura 2.2 demonstra que o funcionamento do gerenciamento de incidentes se dá inicialmente pelo descobrimento e registro do incidente, o qual passa por um processo inicial de

ajuda e classificação baseado na urgência, se referindo diretamente ao quesito de velocidade, e no impacto, que é o quão danoso será aquele problema se não for resolvido. Após as análises iniciais há a fase de verificação na base de registros de incidentes, caso já esteja homologado segue-se o caminho já criado, se não, cria-se o caminho para a solução. Por fim, este caminho de solução contempla toda a parte investigativa e diagnósticos do incidente, resultando em uma proposta e implementação de uma solução, encerrando o processo com o fechamento do incidente após ser completamente resolvido (FILHO, 2012).

O *help desk* consiste em um serviço que busca ser o único ponto de contato entre os clientes e a empresa, trabalhando e resolvendo problemas simples e rotineiros, como recolhimento de dúvidas e sugestões, considerado a primeira etapa de contato com o usuário. Já segundo Filho (2012, p. 122), o *Service Desk* (Central de Serviços) é uma das funcionalidades dentro da tecnologia da informação que tem como objetivo separar os profissionais de suporte daqueles que resolvem os problemas. Geralmente é um atendimento de segundo nível por exigir um nível de capacidade técnica maior.

Figura 2.3: NÍVEIS DE ATENDIMENTO



FONTE: FILHO (2012)

O processo de atendimento ao cliente passa primeiro, como representado na Figura 2.3, pelo serviço centralizado de *Help Desk*. Caso não seja resolvido ou seja complexo para um atendimento de primeiro nível, é transferido para o setor de *Service Desk*, que possui competências melhores para resolver o problema ou incidente em questão.

2.1.3 Softwares semelhantes

No segmento de mercado dos softwares *service desk*, há diversas soluções implementadas que buscam resolver os problemas introduzidos. Como exemplos, podem ser citados os softwares: Ellevio, ServiceNow, Zendesk, Jira Service Desk, Sinergia e Desk Manager.

Os softwares citados, implementam de alguma maneira o gerenciamento de incidentes e serviram de inspiração para o desenvolvimento do Call2r. Eles se diferenciam na abordagem, no custo e nas ferramentas entregues ao usuário de alguma maneira.

Tomando o Desk Manager como exemplo, ele traz uma forma de gerenciar os atendimentos através de um quadro de chamados (DESK MANAGER, 2020), que para o Desk Manager é o seu diferencial. Seguindo este mesmo exemplo, o Call2r implementa uma possível forma de se abordar o gerenciamento de incidentes, utilizando um padrão de listagem dos chamados presentes na aplicação.

Figura 2.4: NÍVEIS DE ATENDIMENTO

The screenshot displays a web-based service desk interface for '[your company]'. At the top right, there are navigation links: 'home', 'support', 'logout', and 'about us'. The main content area is titled 'Update Case' and contains the following fields and sections:

- Update Case Form:**
 - Ticket code: 3
 - Opened Date: 25/07/2006
 - Requester: admin
 - Priority: low
 - Subject: My dog won't bark no more
 - Description: my dog is not barking as usual
 - Assign to: tech
 - Postponed to: (empty)
 - Solution: the dog just wanted a bone, I gave one to him, now it's ok
- Current Status:** Assigned
- Set status as:** Wait For User Confirmation
- Additional Information:** (empty)
- Contact Information:** (empty)
- Case Notes (1):**
 - Date: 23/07/2006 10:13:44 p.m.
 - Note text: hey tech, be careful with the bite
 - From: dispatcher
- Case History (2):** (empty)

At the bottom of the form, there is an 'Update' button and a copyright notice: '© 2006 [your company]. All rights reserved.'

FONTE: RODRIGUEZ (2013)

Fazendo uma análise entre preço e ferramentas dos softwares, podemos identificar alguns padrões. Observando o espectro que diz respeito ao preço, os softwares ServiceNow e Ellevo apresentam funcionalidades bem desenvolvidas porém nos levam diretamente para uma guia de orçamento, não permitindo nem mesmo realizar testes das suas implementações. Já o Jira Service Desk, permite em seu pacote gratuito até 10 usuários ou 3 agentes (JIRA SERVICE DESK, 2020).

Este comportamento no qual softwares bem desenvolvidos são de difícil acesso ou nem mesmo apresentam seus preços, é recorrente. Em contrapartida, softwares *service desk* de baixo custo acabam pecando em entregar uma boa experiência de usuário, como observado na Figura 2.4 com o software Sinergia.

Apesar desses pontos opostos aparentes entre preço e qualidade. O cliente acaba preferindo na maior parte dos casos qualidade, buscando benefícios como boa aparência, confiança, segurança e durabilidade (BARRETO, 2015). E é no meio-termo desses pontos opostos onde o Call2r se situa, buscando entregar uma boa experiência para o usuário, porém de maneira acessível.

2.2 FUNDAMENTAÇÃO TEÓRICA DAS TECNOLOGIAS

Diferente da abordagem relativa aos negócios, na abordagem das tecnologias, o foco passa a ser tudo o que será utilizado como ferramenta para tornar a proposta viável.

Nas seções seguintes são detalhadas as tecnologias e *frameworks* utilizados durante o projeto. Dentre as tecnologias detalhadas, estão as linguagens utilizadas, banco de dados, arquitetura e metodologias.

2.2.1 Unified Modeling Language UML

A UML é uma sigla para “*Unified Modeling Language*” que surgiu da metade para o final dos anos noventa. Como a definição da sigla já diz, a UML consiste em uma linguagem unificada e visual. É a compactação de três outras linguagens de modelagem (OMT, OOSE e Booch), que é utilizada para especificar, construir, e documentar artefatos de sistemas (LARMAN, 2007).

A UML tem características que permitem que ela seja implementada e utilizada em projetos dos mais variados tipos: que se distancie do método de desenvolvimento; que não necessite de uma ferramenta específica de modelagem; que possa ser estendida conforme a necessidade e que por fim agrupa um conjunto com valor significativo das mais diversas técnicas e diagramas.

O funcionamento e estruturas da UML são bem abrangentes como já citado previamente, e contemplam um conjunto bem distinto de representações e notações que segundo Fowler (2003, p.13) são bem precisas na mensagem que querem transmitir, além de poderem ser utilizadas nos mais diferentes níveis de abstração e de domínio de problemas. Toda essa comunicação é realizada através de representações gráficas denominadas de diagramas. Estes

são produzidos para permitir a visualização de um projeto sob diferentes pontos de vista, sendo que o mesmo elemento pode em mais de um diagrama dependendo de sua utilidade (BOOCH; RUMBAUGH; JACOBSON, 2006).

Dentre os diagramas existentes é possível citar exemplos como:

- Diagrama de caso de uso: Apresenta e aponta o relacionamento entre uma determinada ação dentro de um sistema e seu respectivo ator, responsável por realizar e iniciar a ação designada. Trabalha com uma ideia de escopo, já que representa de maneira geral todas as funcionalidades que compõem o escopo do projeto;
- Diagrama de classe: Tem como principal foco apresentar visualmente um conjunto interativo, através de relações dos mais diversos tipos, entre classes, interfaces e colaboradores, como sistemas externos por exemplo. As relações dentro de um diagrama de classes podem ser no nível mais básico de generalização, associação ou até mesmo de dependência (DA SILVA; VIDEIRA, 2001);
- Diagrama de estado: Também conhecido pelo meio como diagrama de transição de estados, ele nos permite realizar o planejamento e a representação do comportamento de um objeto específico, de um subsistema e até mesmo de um sistema denominado de global (DA SILVA; VIDEIRA, 2001);
- Diagrama de sequência: O principal ponto do diagrama de sequência é o eixo representante do tempo. Este auxilia e passa ao observador do diagrama o momento exato no tempo que algo deve ser realizado, ou seja, demonstra uma ideia organizada que deve ser executada sequencialmente, estrutura que foi definido pelos determinados estímulos planejados e executados;

2.2.2 PHP

O PHP, *PHP: Hypertext Preprocessor*, foi criado por Rasmus Lerdof em 1994, inicialmente como um conjunto de binários em linguagem C. Não muito tempo depois, ela foi atualizada para permitir integração com bancos de dados e HTML para que fosse possível construir aplicações *web*. Em 1995, Lerdof liberou o código-fonte do PHP para que outros pudessem atualizar e corrigir ela (PHP, 2020a). O PHP é uma linguagem de script de propósito geral, mas possui foco da utilização no desenvolvimento *web*.

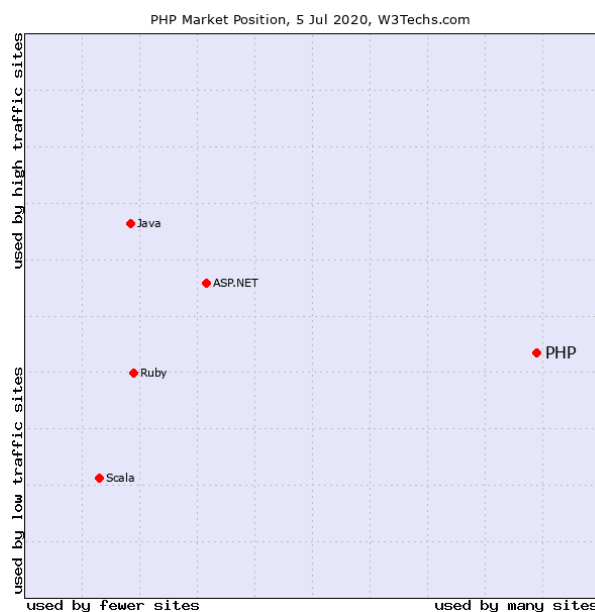
A linguagem pode ser utilizada dentro de arquivos HTML e o motivo que diferencia ela do popular JavaScript, que é outra linguagem que pode ser utilizada em conjunto com HTML, é a forma e a localidade de onde o código é executado. O PHP tem seu código executado no servidor, gerando o HTML que é enviado para o navegador. Navegador esse que não conhece o código fonte, sendo passível de configuração para até mesmo processar todos os arquivos HTML com o PHP (PHP, 2020b).

A linguagem se tornou popular por possuir uma curva de aprendizagem simples porém com a apresentação de recursos avançados para quem despende mais tempo a fim de masterizar-lá.

Essas características tornaram o PHP a linguagem mais utilizada quando o assunto é a renderização do lado do servidor. Sendo utilizada por grandes *websites* como o Facebook, Wikipedia, Wordpress e Globo (W3TECHS, 2020).

O PHP é utilizado por 79% dos *websites* que possuem essa característica e se situa em uma posição relativamente boa quando comparada em termos de popularidade e tráfego com outras linguagens de programação do lado do servidor (W3TECHS, 2020), assim como pode ser avistado na Figura 2.5.

Figura 2.5: USO DO PHP EM WEBSITES



FONTES: W3TECHS (2020)

Para a organização de arquivos, caminhos e para facilitar realizar o *autoload* de arquivos, foi utilizado o padrão PSR-4 do PHP. Utilizando *autoload* na aplicação PHP, não é necessário ficar realizando *includes* nas classes, já que o *composer* gerencia os arquivos conforme eles são necessários.

O PSR-4 é um padrão que descreve como devem ser organizados os caminhos para classes que serão realizados o *autoload* e nomenclatura para essas classes e caminhos (PHP-FIG, 2013).

2.2.3 Symfony

O Symfony é um *framework* de desenvolvimento para a linguagem PHP que traz consigo um conjunto de componentes, uma filosofia e uma forte comunidade que funcionam em harmonia (SYMFONY, 2020a).

Tem seu foco na criação de aplicações *web* e *websites*, que podem ser construídos se beneficiando das vantagens do *framework*. O conjunto de componentes disponibilizado pelo Symfony são dissociados e reutilizáveis, o que permite uma alta flexibilidade para o desenvolvimento.

A comunidade como citada previamente é composta por mais de 600.000 desenvolvedores pertencentes a mais de 120 países (SYMFONY, 2020a), fazendo com que qualquer dúvida possa ser rapidamente sanada utilizando o *Stack Overflow* ou o Slack (SYMFONY, 2020b).

Por fim, o Symfony apresenta algumas vantagens em sua utilização, como uma série de recursos facilitadores, está em constante evolução, possui uma boa reputação e permanência no mercado, além de apresentar um bom grau de interoperabilidade (SYMFONY, 2020c).

2.2.4 React

O React é em sua essência uma pequena biblioteca declarativa da linguagem de programação ECMAScript, popularmente conhecido como JavaScript, voltada especialmente para se criar interfaces para o usuário. Foi idealizada e concebida pelo Facebook, especialmente pelo engenheiro de software Jordan Walke.

Suas intenções eram encarar problemas relacionados com escalabilidade e *websites* orientados a dados, dando um foco na utilização de JavaScript funcional para compor grandes interfaces com a utilização de blocos de códigos pequenos e únicos denominados de componentes (BANKS; PORCELLO, 2017).

Ganhou popularidade por não possuir uma elevada curva de aprendizado, sendo segundo Banks e Porcello (2017, p.3) uma das prioridades na concepção do React, uma vez que busca evitar confusão no processo de aprendizado, construindo uma base sólida de conhecimento.

Atualmente um diferencial competitivo do React é que ele pode ser renderizado em um servidor e ser utilizado para a criação e desenvolvimento de aplicações móveis, fazendo uso de um recurso chamado de React Native (FACEBOOK, 2019).

2.2.5 TypeScript

O TypeScript é uma linguagem de código aberto que se baseia no JavaScript e tem como intuito atuar como um super conjunto de ferramentas para o próprio JavaScript, oferecendo principalmente recursos de tipagem.

Segundo a Microsoft (2020), tipos são uma forma de descrever a forma de um objeto, provendo uma melhor documentação e permitindo que o TypeScript atue validando se o código desenvolvido está funcionando corretamente.

Vale ressaltar que todo o código TypeScript é transformado em JavaScript através de um compilador, que pode ser o próprio compilador do TypeScript ou outros, como o Babel. Essa transformação permite que o código TypeScript desenvolvido seja executado nos mesmos

ambientes que um código JavaScript, como por exemplo em navegadores, no Node.js ou em aplicativos (MICROSOFT, 2020).

A adoção do TypeScript em um ambiente de desenvolvimento não é binária, logo é possível ir adaptando e convertendo conforme a necessidade e confiança na tecnologia. (MICROSOFT, 2020).

2.2.6 MySQL

O MySQL é um sistema de gerenciamento de banco de dados de código aberto, que faz a utilização do paradigma relacional, suportando a linguagem SQL para programação de suas funcionalidades (MYSQL, 2018).

Foi criado e desenvolvido pela MySQL AB Limited Company em meados da década de oitenta, porém hoje o MySQL faz parte da empresa Oracle após a aquisição da Sun Microsystems (NEVES e RUAS, 2001). Seu sucesso veio atrelado a suas principais características, como: portabilidade, suportando diversas plataformas, compatibilidade com dezenas de linguagens de programação, boa constância e desempenho, pouco consumo da máquina utilizada para sua execução, fácil controle de suas instruções e acima de tudo é um software livre, o que fez com que as mais variadas pessoas olhassem para o MySQL.

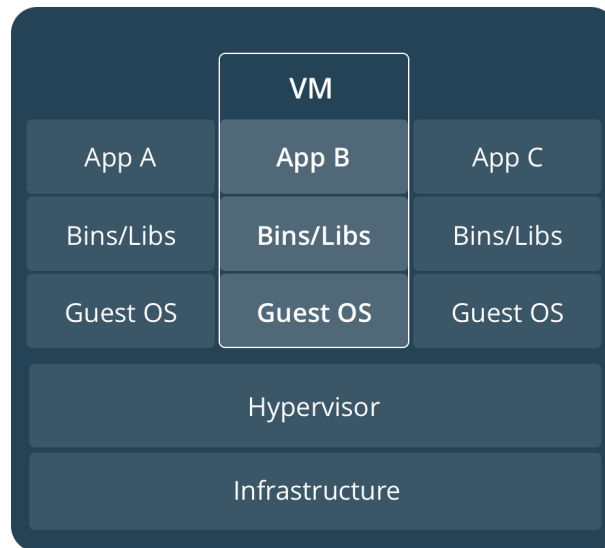
O MySQL busca cumprir o papel de ser o sistema que controla e gerencia a criação, leitura, atualização, deleção, correta manutenção e acesso dos dados existentes e armazenados em um único ou em um conjunto de bancos de dados existentes. Entregando ao seu usuário um servidor de banco de dados robusto e muito rápido (MYSQL, 2018).

A utilização da linguagem de consulta estruturada, conhecida como SQL, para seu funcionamento e execução elevou o nível da implementação permitindo o acesso ao conhecimento, mesmo que de forma embrionária, acerca do conjunto de dados existentes. A implementação do paradigma garantiu bons frutos. Ela trabalha com conceito de entidades e relacionamentos. Isso permite algumas características importantes para os dados persistidos na base, como a tendência a um banco de dados bem construído não ter dados inconsistentes, extraviados ou até mesmo duplicados (MYSQL, 2018).

2.2.7 Docker

O Docker é uma plataforma existente desde o começo da década atual criada por Solomon Hykes, para se construir, compartilhar e executar aplicações fazendo a utilização de contêineres para a instalação do que se tiver interesse através de uma camada de automação e abstração (DOCKER, 2019b). A vantagem desse processo é que com a utilização de contêineres seu projeto ganha características como flexibilidade, leveza, portabilidade, encapsulamento, escalabilidade e segurança.

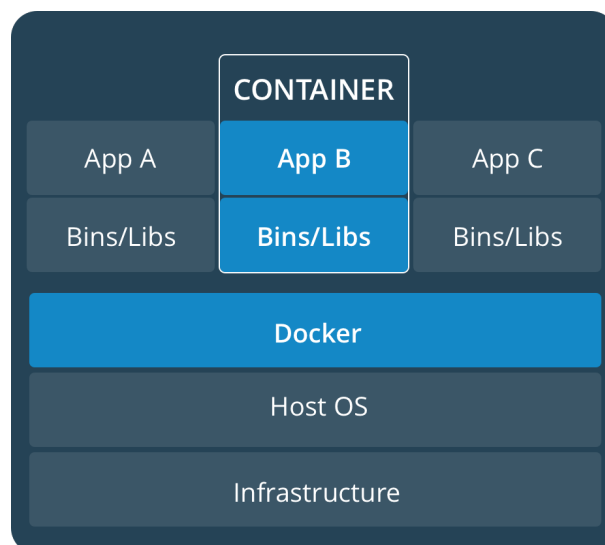
Figura 2.6: EXEMPLO DE APLICAÇÃO NORMAL



FONTE: Docker (2019b)

Todas essas vantagens são possíveis pela diferença de contêineres e máquinas virtuais, já que assim como visto na Figura 2.6 as máquinas virtuais simulam um sistema operacional sendo necessário a configuração de um ambiente completo para a execução da aplicação desejada, diferentemente do contêiner observado na Figura 2.7, que é muito menor evitando uma sobrecarga de operação. Isso se justifica no compartilhamento de recursos da máquina hospedeira com os outros contêineres alocados naquele ambiente. Dessa forma, um ambiente completo não precisa ser reproduzido, mas o contêiner acaba contendo dentro de si todos os recursos, como bibliotecas necessárias, para a execução sem falhas da respectiva aplicação.

Figura 2.7: EXEMPLO DE APLICAÇÃO COM DOCKER



FONTE: Docker (2019b)

O Docker tem seu funcionamento pautado na criação de imagens, que ditam as regras e configurações de um contêiner, como se fosse a configuração do sistema operacional que será executado em um ambiente genérico. A imagem inclui tudo que é necessário para se executar a aplicação, garantindo que o contêiner por si só seja capaz de executar o que é desejado, tendo em vista que um dos preceitos de tal tecnologia é o isolamento da máquina hospedeira e de outros contêineres (DOCKER, 2019b). Essa criação de imagens traz outros benefícios, como por exemplo o compartilhamento de imagens, que facilitam e evitam retrabalhos na criação de outros ambientes.

2.2.8 Métodos Ágeis

Os métodos ágeis segundo Cohen (2004, p.3), surgiram na metade da década de noventa como uma resposta aos métodos tradicionais para o desenvolvimento de softwares e tiveram evoluções com o decorrer do tempo, visto que o método tradicional demandava uma série de documentações e requerimentos que eram muitas vezes frustrantes e até desmotivadores para o processo e para a equipe envolvida com o projeto.

Além de serem uma resposta evolutiva ao método tradicional, segundo Ribeiro (2015, p.22) os métodos ágeis trabalham de uma forma muito mais dinâmica e suscetível à mudança do que o padrão burocrático tradicional. Com isso acabou tornando-se um ponto crucial no mundo dos desenvolvedores de softwares, já que os clientes se converteram e passaram a ser muito mais exigentes e participativos em todo esse ecossistema.

A metodologia ágil é pautada em alguns princípios descritos pelas mãos de um grupo de desenvolvedores que publicaram seu respectivo manifesto, que conta com alguns conceitos como: indivíduos e interações acima de processos e ferramentas, software funcionando acima de documentação compreensiva, colaboração com o cliente acima de negociação de contratos e respostas a mudanças acima do cumprimento de um plano (AGILE MANIFESTO, 2001).

A metodologia em questão, segundo Jim Highsmith, significa ser capaz de “Entregar rápido, modificar rápido e modificar frequentemente” (COHEN et al, 2004). Para isso ser efetivamente implementado há diversas ferramentas que praticam e enfatizam pontos como o desenvolvimento iterativo, foco na interação do grupo e comunicação e redução de artefatos intermediários. Dentre tais ferramentas que compartilham dos mesmos valores, existem: o “*Extreme Programming*”, o “*Crystal Method*”, o “*Lean Development*” e o popular e conhecido “Scrum”.

2.2.8.1 Scrum

O Scrum é um método ágil, o qual busca ser uma estrutura para o desenvolvimento de produtos através de diversas interações com entregas incrementais. Além disso, possui algumas características empíricas durante o processo, uma vez que se adquire experiência prática com o desenvolvimento do ciclo de vida do produto.

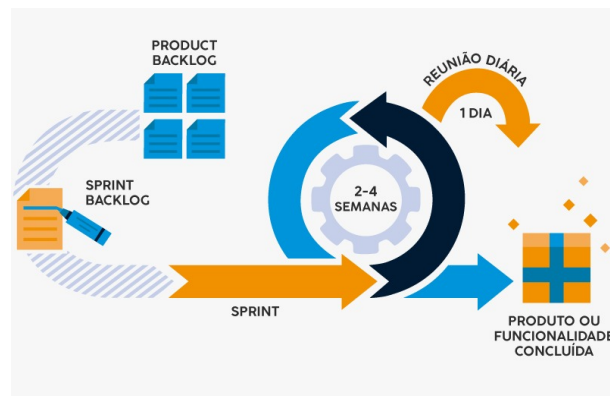
Tal estrutura foi construída e aprimorada partindo do artigo “The New New Product Development Game” (TAKEUCHI; NONAKA, 1986), onde foi apresentada uma jogada característica do rugby chamada de Scrum. Tal jogada inspirou e originou o nome para o *framework*, onde os atletas se organizam com as cabeças baixas com o objetivo de empurrar o adversário para adquirir a posse de bola através de um trabalho de força em conjunto (RIBEIRO; RIBEIRO, 2015). Essa jogada é utilizada para se descrever as vantagens de se trabalhar em grupo para alcançar o objetivo desejados e foi adaptada para o ambiente de desenvolvimento de softwares.

Dentro da metodologia há a criação de dois papéis principais, o Scrum *Master* (mestre de Scrum) e *Product Owner* (dono do produto). O mestre de Scrum segundo Cohn (2011, p.139) faz analogia a um treinador que existe para auxiliar o seu time com o uso do Scrum, sempre incentivando sua equipe para condicioná-los a aplicar o Scrum da maneira correta. Já o dono do produto segundo Audy (2015, p.45) representa a visão do cliente, ou seja, é aquela figura que determina o que será realizado ou não, o que deve ser priorizado e o que agregará mais ao produto que está sendo desenvolvido.

O Scrum tem seu funcionamento pautado em três pilares: transparência, adaptação e inspeção, os quais transmitem uma estrutura prática para se desenvolver o processo de desenvolvimento e entrega do produto.

O processo representado pela Figura 2.8 é construído por meio de etapas e artefatos. O primeiro artefato é o “*Product Backlog*” que é onde se concentra todas os requerimentos necessários para a conclusão do projeto. Após o primeiro artefato, se tem o “*Sprint Backlog*”, que são todos os requerimentos que devem ser desenvolvidos e concluídos dentro da *sprint*.

Figura 2.8: EXEMPLO DE APLICAÇÃO DO MÉTODO SCRUM



FONTE: Docker (2019b)

Logo após o segundo *backlog*, temos o processo da “*sprint*” propriamente dita, que consiste na etapa onde será efetivamente desenvolvido as atividades que estão sendo requeridas. E por fim, temos a fase de entrega do que foi desenvolvido durante o ciclo em questão.

A metodologia conta também com eventos de tempo definido, como por exemplo a reunião, evidenciada também na Figura 2.8, que segundo Schwaber e Sutherland (2017, p.12) é utilizada pelo time de desenvolvimento para visualizar o progresso em direção aos objetivos da

sprint, realizada através de simples perguntas como: o que eu fiz?, o que eu vou fazer? e o que me impede de fazer?.

2.2.9 Git

Dentro do desenvolvimento de software, um dos artefatos que mais requerem cuidado e atenção é o código-fonte da aplicação e para isso, existem diversas ferramentas para se manter o controle de versionamento do código-fonte. Este processo tem como objetivo principal gerenciar as mudanças em um arquivo durante o tempo, facilitando o processo e permitindo que a equipe toda trabalhe ao mesmo tempo.

O Git foi desenvolvido por Linus Torvalds, com o intuito de se fazer o gerenciamento do código-fonte do projeto de código aberto do *kernel* do Linux (CHACON; GIT, 2005a). A principal mudança aparente para os outros sistemas de versionamento se encontra no modelo adotado, que no caso se trata do modelo de *branches*.

Cada *branch* é basicamente uma área de trabalho independente de qualquer outra, o que permite que o desenvolvedor possa realizar suas atividades e posteriormente anexá-las a área de trabalho principal. Essa característica permite um desenvolvimento não linear, criando os chamados fluxos de trabalho do Git.

Além do modelo instaurado pelo Git, outras vantagens são apresentadas, como: velocidade, uma vez que o git é executado de maneira local sem diversas requisições a um servidor, distribuição, segurança das informações e sem custo de utilização por ser um projeto de código aberto (CHACON; GIT. 2005b). Todas essas vantagens e pontos, fizeram com que o Git fosse amplamente difundido pela comunidade de desenvolvedores, tornando ele essencial na maioria dos projetos desenvolvidos atualmente.

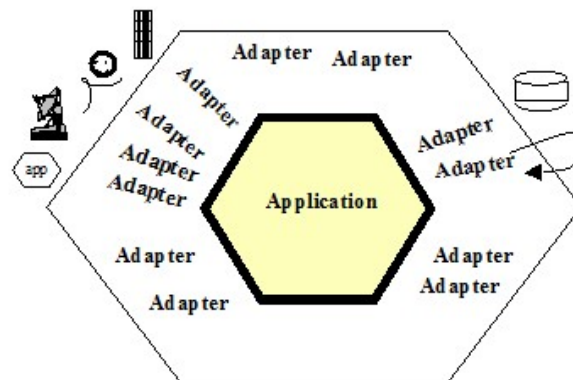
2.2.10 Arquitetura Hexagonal

A arquitetura escolhida para ser usada na aplicação é a arquitetura hexagonal. Nesta arquitetura a aplicação é dividida em camadas, visando separar as regras de negócio, o núcleo do sistema, de outras funções e classes do sistema. O motivo para tal separação é evitar misturar regras de negócio com códigos que não tem relação, o que acaba deixando o código mais complexo e complicado para manutenção.

Realizar a separação em camadas permite um código mais limpo e legível. Também facilita para manutenção do código já que os métodos podem ser separados de acordo com seus objetivos.

A ideia da separação na arquitetura hexagonal é apresentada na Figura 2.9, onde a aplicação é encapsulada e as entradas e saídas são feitas por *adapters*. Os *adapters* são classes ou interfaces que tem a função de realizar a troca de dados entre a camada interna e a camada externa à ela. Além disso, é feita quaisquer conversões de dados necessárias para que ambas as camadas possam se comunicar através dos *adapters*. A aplicação envia informações através

Figura 2.9: EXEMPLO DE ARQUITETURA HEXAGONAL



FONTE: Cockburn (2005)

de uma porta para o *adapter*, que converte o que recebeu para um formato compatível com a respectiva saída. Para a entrada de informações, o fluxo é apenas invertido (COCKBURN, 2005). Com essa arquitetura, o núcleo da aplicação não necessita ter nenhum conhecimento do que está externo a ela do outro lado dos *adapters*.

2.2.11 S.O.L.I.D.

S.O.L.I.D. são cinco princípios escritos por Robert C. Martin, também conhecido como *Uncle Bob*, para a programação orientada a objetos (POO). Esses princípios tem o objetivo de fazer os desenvolvedores escreverem códigos mais limpos, com maior reaproveitamento de código e um melhor gerenciamento de dependências (MARTIN, 2005).

O primeiro princípio é o *Single Responsibility Principle* (Princípio da responsabilidade única) na qual uma classe deve ter um, e apenas um, motivo para mudar (MARTIN, 2005), ou seja, a classe deve possuir apenas uma responsabilidade e ser especializada em apenas um assunto.

O segundo princípio é o *Open-Closed Principle* (Princípio aberto-fechado) que diz que o desenvolvedor pode estender comportamentos de classes sem modificá-las (MARTIN, 2005), ou seja, as classes devem estar abertas para extensão e fechadas para modificação. O objetivo desse princípio é evitar que códigos fontes antigos que já estão em funcionamento sejam alterados, podendo produzir erros em um objeto ou classe que já estava funcionando corretamente.

O *Liskov Substitution Principle* (Princípio da substituição de Liskov) é o terceiro princípio de Robert C. Martin que diz que classes derivadas devem ser substituíveis por suas classes base (MARTIN, 2005). O princípio foi introduzido por Barbara Liskov em 1988 e tem como objetivo permitir o uso do polimorfismo e usar classes derivadas se referindo à classe base sem preocupações.

O princípio seguinte é o *Interface Segregation Principle* (Princípio da segregação da interface) que diz que devem ser feitas classes refinadas que são específicas do cliente (MARTIN, 2005). O princípio diz que devem ser feitas interfaces específicas ao invés de interfaces mais genéricas.

O último princípio é o *Dependency Inversion Principle* (Princípio da inversão da dependência), que diz para depender de abstrações e não de implementações (MARTIN, 2005). Depender de abstrações facilitam as mudanças e evoluções do código, já que mudam menos que implementações.

Também foi utilizado o *Object calisthenics* no projeto. O *Object calisthenics* foi introduzido por Jeff Bay e são princípios do SOLID aplicados. As regras introduzidas por Jeff Bay tem como objetivo a manutenção e legibilidade do código, permitindo que o código tenha uma compreensão melhor para que haja menos trabalho ao alterá-lo. As regras do *Object calisthenics* são (BAY, 2008):

1. Usar apenas um nível de indentação por método;
2. Não usar *ELSE*;
3. Encapsular todos os primitivos e *Strings*;
4. Envolver as coleções em classes;
5. Usar apenas um ponto por linha;
6. Não abreviar;
7. Manter todas as entidades pequenas;
8. Não ter classes com mais de duas variáveis de instância;
9. Não usar *Getters* ou *Setters*.

2.2.12 Linguagem Ubíqua

No livro "*Domain-Driven Design: Tackling Complexity in the Heart of Software*", Evans (2003) argumenta que os especialistas do domínio possuem um conhecimento limitado sobre os jargões que são utilizados no desenvolvimento de software e acabam utilizando termos da sua própria área de atuação. Em contrapartida os desenvolvedores acabam criando diversas abstrações que suportam a forma na qual arquitetaram o seu projeto, mas que não necessariamente são os mesmos jargões utilizados pelos especialistas.

Isso acaba, segundo o autor, gerando uma divisão linguística dentro do próprio projeto, onde poucos especialistas do domínio e desenvolvedores se tornam bilíngues, ocasionando gargalos no fluxo de informação e até mesmo traduções errôneas.

Neste tipo de projeto, sem uma linguagem comum, há de maneira recorrente a necessidade de diversas traduções, essas que podem causar confusões que afetam diretamente os conceitos de algum modelo presente no sistema.

Como é muito custoso a sobrecarga das traduções mais o risco de desentendimento, Evans (2003) defende que há a necessidade de se ter uma linguagem comum a todos, mais robusta do que simplesmente o menor denominador comum. Também é necessário contar com um esforço coletivo do time para que a linguagem seja ubíqua e esteja presente e interconectada por todos, facilitando principalmente a comunicação entre os membros.

O vocabulário da linguagem ubíqua inclui em seu escopo nomes de classes, operações proeminentes e até mesmo termos presentes para se discutir regras do modelo. (EVANS, 2003).

Por fim, a linguagem do grupo é enriquecida com nomes de modelos que os times comumente aplicam ao modelo de domínio, fazendo com que os significados das frases e palavras soem semânticas ao modelo abordado.

3 METODOLOGIA

Ao decorrer do desenvolvimento de um projeto há diversos estudos que devem ser realizados, ferramentas que devem ser aplicadas e conhecimentos que devem ser adquiridos para se garantir que o objetivo final seja alcançado. Todas essas especificidades que envolvem o processo produtivo se encontram no capítulo em questão.

3.1 MATERIAIS E MÉTODOS

O projeto está pautado e foi desenvolvido sob a ótica da metodologia ágil, utilizando conceitos que englobam e trabalham com quase todo o escopo do Scrum permitindo que o sistema fosse desenvolvido de maneira incremental através de diversas iterações evolutivas.

Fazendo-se o uso de tais procedimentos há a tentativa de garantia de preceitos de adaptabilidade contínua a qualquer tipo de alterações que venham a existir. Para buscar isso por um meio mais assertivo é necessário se trabalhar em conjunto com diversas ferramentas de controle e organização, que auxiliam e atualizam a equipe com o passar do tempo em relação ao andamento das tarefas, e em relação às medidas que devem ser tomadas considerando o atual cenário.

As ferramentas de auxílio em questão serão descritas nas subseções a seguir.

3.1.1 Astah

No início do projeto foi necessário criar diagramas padronizados usando UML e para isso, foi utilizada a ferramenta Astah. Com ela, foi possível criar os diagramas de caso de uso, de classe, de implementação e de sequência, que visam mostrar de forma simplificada as funcionalidades do sistema e facilitar o desenvolvimento do projeto (ASTAH, 2019).

A ferramenta Astah foi escolhida por dar suporte a licença de estudante gratuita para uso da versão Astah UML, facilitando o uso completo da ferramenta para desenvolvimento dos diagramas.

3.1.2 GanttProject

Para que fosse possível gerenciar as tarefas, responsabilidades e esforço de cada membro da equipe, foi necessário criar um diagrama de Gantt antes de iniciar o desenvolvimento do projeto. Para isso, foi utilizada a ferramenta GanttProject, na qual foi possível gerenciar as tarefas, o tempo de desenvolvimento e o responsável por elas, ficando mais fácil e visual o gerenciamento das tarefas (GANTTPROJECT, 2019).

As responsabilidades foram atribuídas para cada membro da equipe de acordo com o tempo disponível e os conhecimentos necessários para se realizar a tarefa. Já a duração de cada tarefa foi baseada em atividades passadas.

3.1.3 Trello

Para a organização das tarefas e prazos, foi necessário utilizar uma ferramenta na qual fosse possível criar um quadro virtual de Kanban compartilhado, para que cada membro da equipe pudesse visualizar as tarefas a se fazer, sendo feitas, impedidas, em processo de revisão e entregues. A ferramenta escolhida pela equipe foi o Trello (TRELLO, 2018), devido a familiaridade que todos os membros da equipe possuem e por ser uma ferramenta conhecida e prática.

3.1.4 Adobe XD

Para o desenvolvimento de protótipos das telas, foi necessário uma ferramenta robusta e que facilitasse o desenvolvimento no *frontend* futuramente. O Adobe XD permitiu organizar os componentes que seriam feitos e então desenvolver os protótipos das telas com o desenvolvimento do *frontend* pensado, incluindo interações entre botões e novas telas que o sistema apresenta (ADOBE, 2019).

Além das funcionalidades, o Adobe XD foi escolhido como ferramenta por ser gratuito e pelo membro da equipe responsável ter bom conhecimento da ferramenta, poupando tempo e permitindo o melhor desenvolvimento das telas.

3.1.5 Docker

No início do desenvolvimento do projeto, foi discutido a possibilidade de se usar alguma ferramenta para um ambiente DevOps para facilitar atualizações e implementações de novas funções durante o desenvolvimento. DevOps serve para otimizar e aperfeiçoar produtos em um ritmo mais rápido do que o das empresas que usam processos tradicionais utilizando um *pipeline* de entrega com construção, teste, *release* e então monitoração da aplicação (AMAZON, 2019). Além disso, usar a containerização para evitar problemas de incompatibilidade entre os diferentes hardwares utilizados (DOCKER, 2019a).

A ferramenta escolhida para isso foi o Docker, ferramenta gratuita e que os membros da equipe já possuíam algum conhecimento prévio.

3.1.6 GitHub

O Github se apresenta como uma plataforma que tem como propósito oferecer a hospedagem para códigos-fonte, permitindo que diversos usuários de qualquer lugar do mundo cooperem entre si em um projeto, seja ele privado ou público. Dentro deste sistema há integração com o controle de versionamento git, que traz consigo todas os benefícios desse sistema.

Dentre as ações existentes no GitHub, há a possibilidade de se copiar um projeto público, de contribuir com um projeto público através da ferramenta de *pull requests* ou de participar de equipes de desenvolvimento. Atualmente o GitHub abriga mais de 100 milhões de projetos, dentre projetos pessoais e projetos empresariais (GITHUB, 2019).

A escolha pelo GitHub se deu por causa da estabilidade da ferramenta e por ser possível utilizar a ferramenta de forma gratuita, mesmo que com poucas limitações. As *branches* principais utilizadas pela equipe são: *master*, que é a base para o Amazon S3, e *develop*, *branch* na qual as atualizações são aplicadas para então serem testadas. Caso as atualizações estejam funcionando e não tragam malefícios e erros graves, como fazer a aplicação não funcionar mais, é realizado o *merge* da *branch develop* com a *branch master*. Outras *branches* são utilizadas para desenvolvimento de funções e logo após finalizar o desenvolvimento, é realizado o *merge* com a *branch develop* para testes e a *branch* utilizada para desenvolvimento é excluída.

Figura 3.1: FUNCIONAMENTO DE BRANCHES NO GITHUB

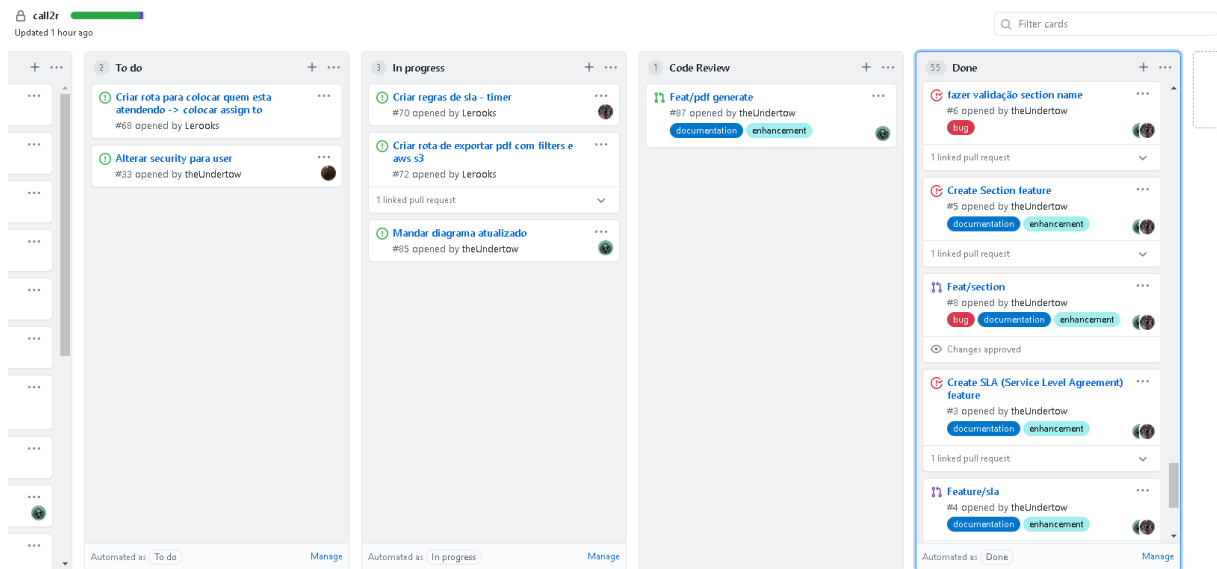


FONTE: CHACON; GIT (2005b)

Uma forma de organizar o desenvolvimento da aplicação, foi por meio das funcionalidades de *issues* e *pull requests* disponíveis no GitHub. Através das *issues* é possível organizar todo o *backlog* da aplicação e categorizar com diferentes bandeiras as novas funcionalidades, correções e melhorias que devem ser desenvolvidas, permitindo que todos os participantes do projeto se situem do atual estado da aplicação.

As *issues* permitem também comentários, menções e podem ser assignadas dentro de uma meta, o que ajuda e auxilia o processo de desenvolvimento ágil (GITHUB, 2014). Todas essas ferramentas e utilizações podem ser observadas na Figura 3.2.

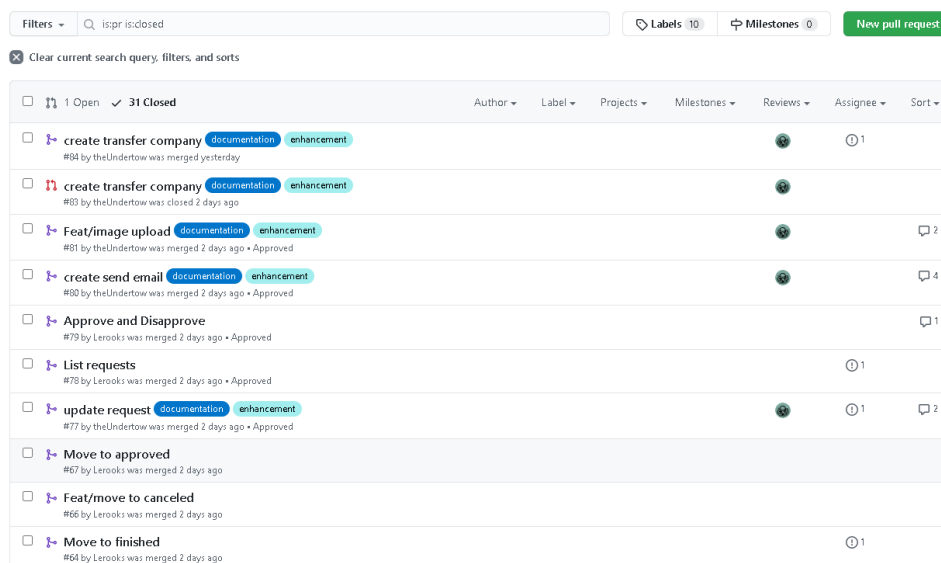
Figura 3.2: KANBAN UTILIZANDO GIT ISSUES



FONTE: Os autores (2020)

Outra funcionalidade utilizada foi a de *pull requests*, que assim como seu nome explicita é uma requisição de envio de um código para o repositório central. Essa requisição pode ser analisada pelos membros da equipe através de um processo de revisão de código que permite comentários, menções e requisições de alterações (GITHUB, 2020), um exemplo pode ser visualizado na Figura 3.3.

Figura 3.3: LISTA DE PULL REQUESTS



FONTE: Os autores (2020)

Assim que esse processo é validado e o aprovado por todos os avaliadores, é possível realizar o processo de *merge*. Isso garante um filtro que evita problemas em *branches* essenciais para o projeto, como a *master* que tem conexão direta com o servidor de produção da aplicação.

3.1.7 Amazon AWS

O serviço utilizado para hospedar o *backend* da aplicação foi o Amazon AWS. O serviço surgiu em 2006 com a criação da versão *beta* do Amazon EC2, ou Amazon *Elastic Compute Cloud* (AMAZON, 2006). Atualmente são oferecidos mais de 175 serviços pela Amazon através do AWS para milhões de clientes (AMAZON, 2020b). Para esta aplicação foram utilizados os serviços Amazon EC2 e S3 que serão detalhados a seguir.

O Amazon *Elastic Compute Cloud* (EC2) foi criado em 2006 como um serviço de computação escalável na nuvem. Com uma interface *web* simples, é possível criar máquinas virtuais para realizar o processamento e facilitar para os desenvolvedores (AMAZON, 2006). É possível configurar máquinas virtuais de acordo com a necessidade e plano da conta e, caso necessário, é possível escalar de maneira fácil as máquinas virtuais e aplicações.

O Amazon EC2 foi configurado com uma máquina virtual com o sistema operacional Linux e foram instaladas apenas as aplicações necessárias, incluindo o Docker e a aplicação do projeto.

O Amazon *Simple Storage Service* (S3) é um serviço de armazenamento e recuperação de dados, que tem seu funcionamento pautado através de *buckets*, que são basicamente o local onde serão armazenados os dados, esses dados por sua vez são tratados como objetos, que são formado de um arquivo e qualquer informação ao seu redor, como por exemplo metadados (AMAZON, 2020c). Há diversas configurações de *buckets* e objetos, elas podem dizer a respeito desde a política de acesso até a escolha da posição geográfica de armazenamento de um *bucket* (AMAZON, 2020a).

Foi utilizado o Amazon S3 para trabalhar com as imagens de perfil do usuário, através de *uploads* e requisições desse recurso para o servidor. Isso garantiu que o sistema não tenha que se preocupar com as formas de armazenar, editar e recuperar essas imagens dentro do banco de dados relacional.

3.1.8 Nginx

O Nginx é basicamente um servidor HTTP e de *proxy* reverso, um servidor de *proxy* de e-mails e um servidor de *proxy* TCP/UDP genérico e um balanceador de carga HTTP (NGINX, 2020). O Nginx é arquitetado de maneira orientada a eventos que são processados de forma assíncrona, o que o torna prático e performático. Permitindo múltiplas conexões simultâneas, essa arquitetura permite também uma escalabilidade horizontal que traz uma série de vantagens para o projeto.

3.1.9 Infraestrutura de Desenvolvimento

Para o desenvolvimento do projeto foram utilizados os seguintes computadores:

- Computador 1:
 - Nome da máquina: Patrick-PC;
 - Proprietário: Patrick Marcelo Dziura;
 - Sistema Operacional: Ubuntu 20.04 | Windows 10 Pro;
 - Processador: AMD Ryzen 5 1600;
 - Memória RAM: 16 GB;
 - Espaço de Armazenamento: 512 GB SSD;

- Computador 2:
 - Nome da máquina: Matt;
 - Proprietário: Mateus Mendes Zanela;
 - Fabricante: Lenovo;
 - Sistema Operacional: Ubuntu 18.x | Windows 10;
 - Processador: Intel Core i5 7200u;
 - Memória RAM: 8 GB;
 - Espaço de Armazenamento: 1 TB HDD;

- Computador 3:
 - Nome da máquina: Leonardo;
 - Proprietário: Leonardo Roberto Rocha;
 - Fabricante: Dell;
 - Sistema Operacional: Windows 10;
 - Processador: Intel Core i7 7500u;
 - Memória RAM: 8 GB;
 - Espaço de Armazenamento: 256 GB SSD;

- Computador 4:
 - Nome da máquina: Olivette;
 - Proprietário: Marco Aurelio Vieira Olivette;
 - Fabricante: Asus;

- Sistema Operacional: Linux Mint 19.1 XFCE;
- Processador: Intel Core i5 7200u;
- Memória RAM: 8 GB;
- Espaço de Armazenamento: 240 GB SSD;

3.2 MODELO DE PROCESSO DE ENGENHARIA DE SOFTWARE

Atualmente um software que foi desenvolvido deve ser capaz de enfrentar diversos problemas modernos dentro do mundo da tecnologia. Dentre os pontos que devem ser observados se encontram exemplos como: os softwares estão difundidos em todo e qualquer local e dentre os mais diferentes aspectos da vida, empresas e pessoas criam laços de forma crescente por softwares para tomadas de decisões e auxílio e quanto mais uma aplicação recebe valor e visibilidade maior será o número de usuário interessados (PRESSMAN, 2011).

Para se alcançar os objetivos determinados e enfrentar os desafios abordados é necessário se pensar no planejamento e na construção do software de uma maneira objetiva e consciente, assim surge o modelo de processo de engenharia de software.

A definição de engenharia de software de acordo com o IEEE (1990, p.67) é de que engenharia de software é "a aplicação de uma abordagem sistemática, disciplinada e quantificável no desenvolvimento, na operação e na manutenção de software; isto é, a aplicação de engenharia ao software".

Segundo a definição acima, a engenharia de software consiste em tratar os problemas de uma forma que segue um sistema pré-definido, nesse caso podemos chamar esses sistemas de metodologias. Segundo Pressman (2011, p.40), as metodologias são *frameworks* que determinam a base estrutural para um processo de engenharia de software completo, não importando sua complexidade e/ou tamanho. E essas metodologias de maneira genérica compreendem atividades como: Comunicação, planejamento, modelagem, construção e emprego.

Além do mais, há diversas metodologias existentes atualmente que implementam essas atividades, como por exemplo as metodologias ágeis com seu principal representante, o Scrum. Por fim, para auxiliar todas essas atividades há diversas atividades de apoio, que são definidas por Pressman (2011, p.41) como atividades aplicadas ao longo de um desenvolvimento de projetos para auxiliar questões como controle de progresso, controle de qualidade, gestão de mudanças e de risco.

Por fim, as metodologias existentes dão uma boa noção de como o projeto no todo deve ser conduzido e desenvolvido, porém em alguns casos especiais é necessário realizar algumas adaptações para atender às idiossincrasias do projeto, como nesse caso em questão. As adaptações realizadas na utilização da metodologia aplicada serão descritas nas subseções seguintes.

3.2.1 Adaptação Scrum

O Scrum apresenta diversas diretrizes e artefatos dentro da forma de seu funcionamento, assim como apresentado anteriormente, entretanto nem sempre é viável se aplicar uma técnica ou ponto exatamente como é descrito pela metodologia.

Dessa forma algumas adaptações foram implementadas pela equipe. A mais visível e impactante foi relacionado às reuniões, em que o Scrum prega a realização de uma reunião diária, entretanto a equipe resolveu se reunir presencialmente junto ao orientador Prof. Dr. Razer Anthom Nizer Rojas Montaña semanalmente para debater assuntos como andamento do projeto, alinhamento e eventuais dúvidas. Na segunda parte da execução do projeto as reuniões entre os membros da equipe passaram a serem executadas virtualmente e quando necessário, era realizada uma reunião com o orientador do projeto.

Outra mudança que teve um impacto significativo foi a escolha do próprio orientador como *Product Owner* (Dono do produto) e *Scrum Master* (Mestre Scrum), geralmente o que ocorre dentro do Scrum é a escolha de dois indivíduos capacitados da própria equipe para realizarem essas atividades. Por fim, foi utilizada a ferramenta Trello para se planejar e realizar um controle acerca daquilo que estava sendo desenvolvido.

3.3 PLANO DE ATIVIDADES

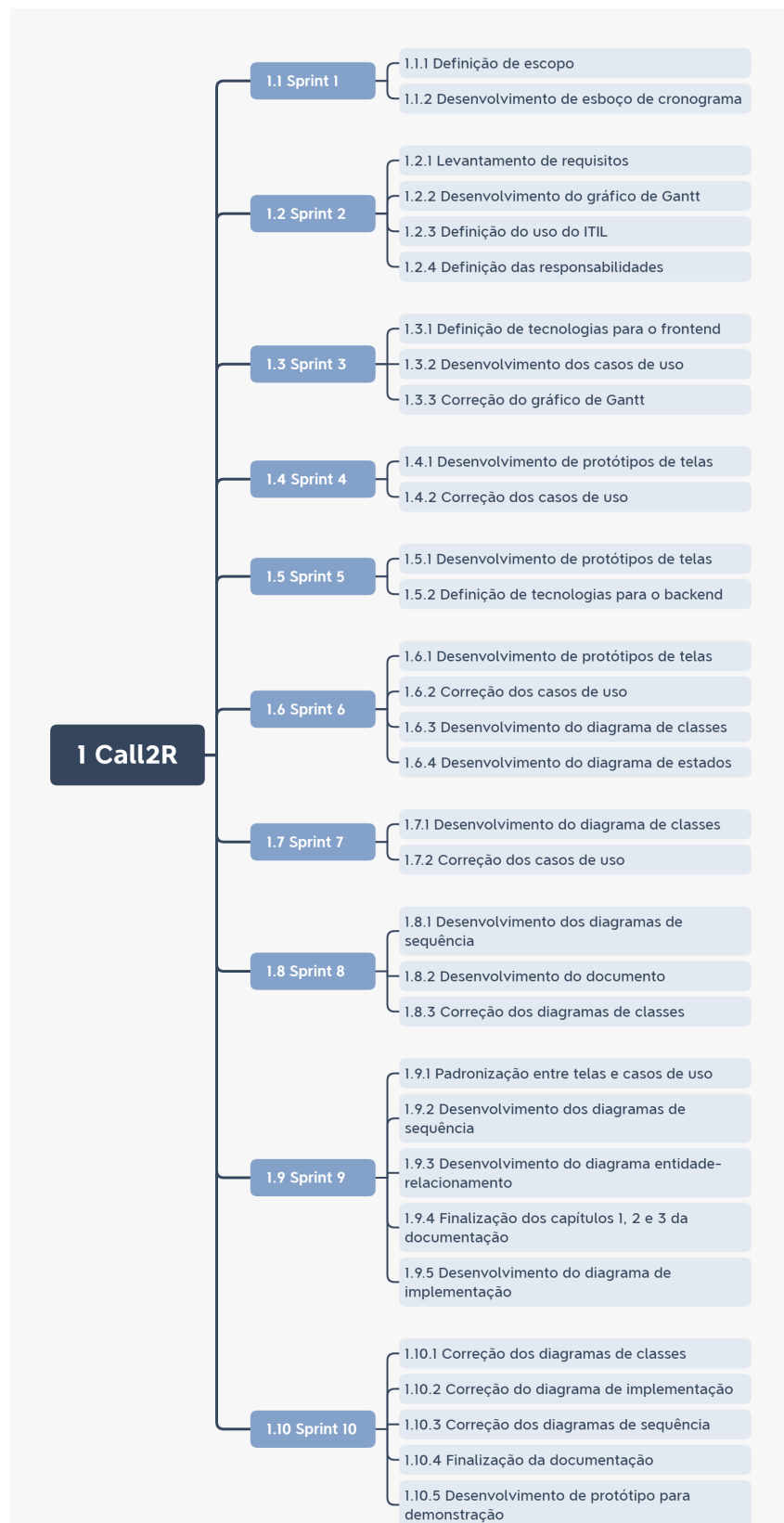
O Plano de atividades ou plano de ação, em simples palavras, é um conjunto de artefatos utilizado para se fazer um planejamento e atingir resultados desejados (DE PAULA, 2016). As informações e artefatos presentes buscam apresentar informações como objetivos, ações e responsáveis, além de respectivas datas de entrega.

3.3.1 WBS

O WBS ou EAP (Estrutura Analítica de Projeto) em português, é uma ferramenta que segundo Petenate (2019) “tem por objetivo decompor o trabalho de um empreendimento em partes mais facilmente administráveis”, auxiliando na definição e entendimento do escopo de um projeto através da exposição de uma árvore hierárquica. Para evitar problemas e garantir que a aplicação seja efetiva e eficiente o WBS deve receber o maior investimento possível, para que seja, ainda segundo Petenate (2019), “o mais completo, organizado e enxuto possível, de modo que sua medição seja facilitada”.

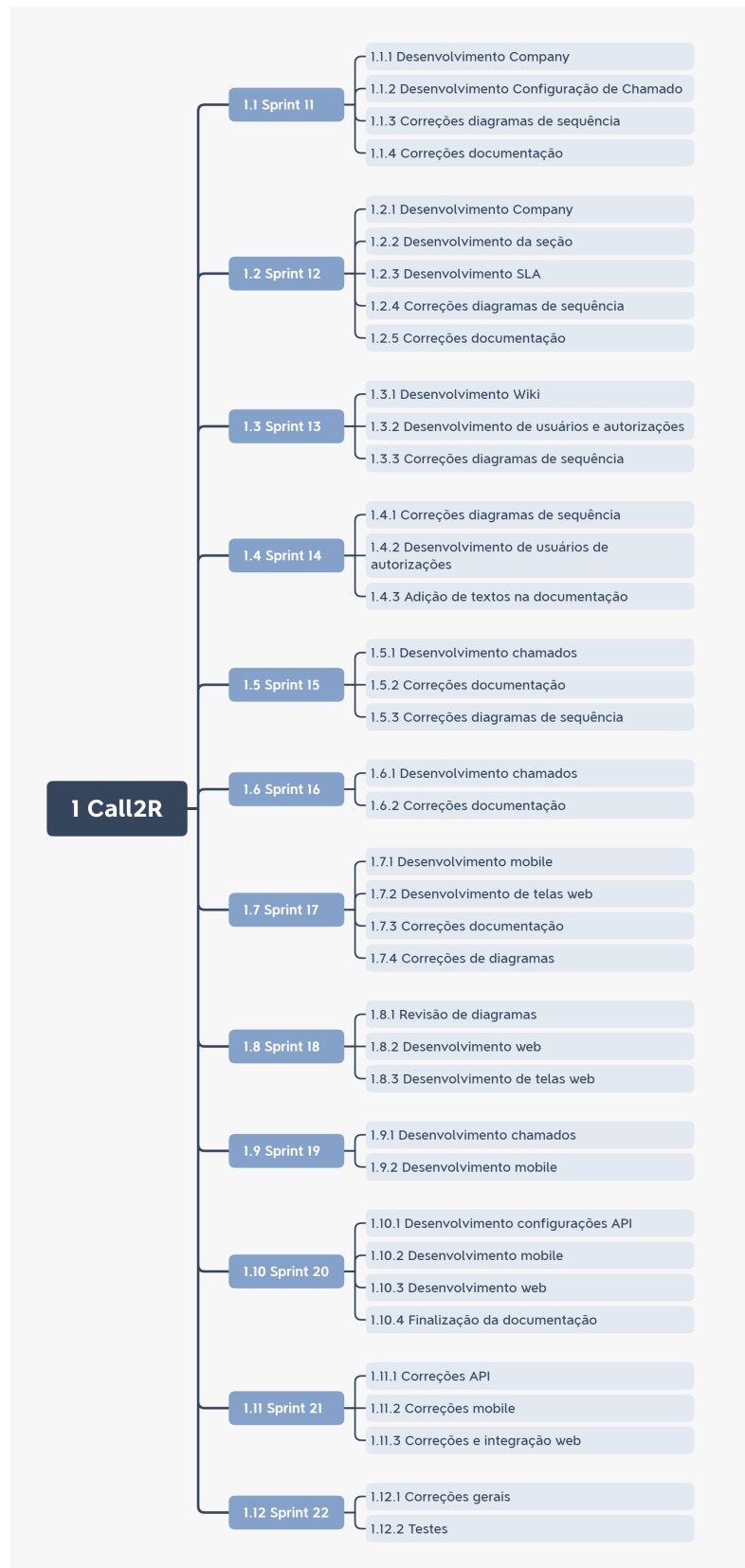
As Figuras 3.4 e 3.5 indicam uma divisão por *sprints* do projeto, apresentando dentro de seus respectivos escopos as atividades e artefatos entregues e/ou realizadas durante o período. Com essa divisão é possível observar de uma maneira geral como se deu o desenvolvimento das atividades e identificar eventuais gargalos e falhas.

Figura 3.4: WBS PRIMEIRA ETAPA



FONTE: Os autores (2020)

Figura 3.5: WBS SEGUNDA ETAPA

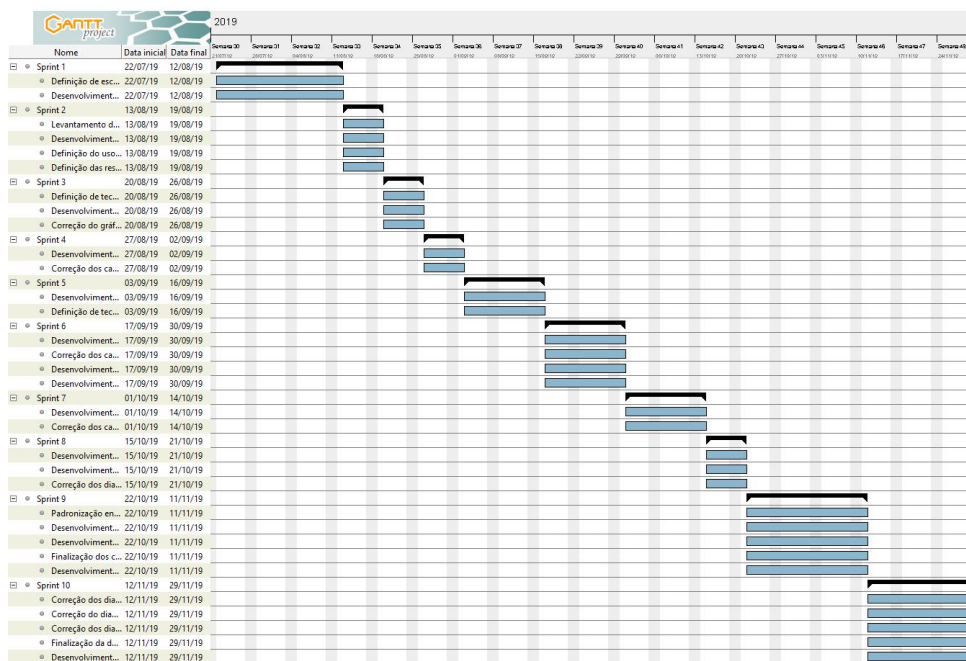


FONTE: Os autores (2020)

3.3.2 Gantt

O diagrama de Gantt, é o resultado da combinação de dados de esforço, data de início e duração de cada atividade presente no projeto (PRESSMAN, 2011). E tem como foco apresentar um cronograma do projeto e suas respectivas atividades, estas que se encontram na coluna presente a esquerda. No eixo das abscissas é representada a duração de cada tarefa considerando os recursos que serão dispostos para a sua realização. Um ponto fundamental para o entendimento do diagrama de Gantt é que segundo Pressman (2011 , p.638) “quando ocorrem múltiplas barras ao mesmo tempo no calendário, é sinal de que há concorrência de tarefas”, o que permite visualizar pontos onde é necessário mais de um recurso e pontos onde a execução de uma atividade é essencial para o andamento de todo o projeto.

Figura 3.6: DIAGRAMA DE GANTT

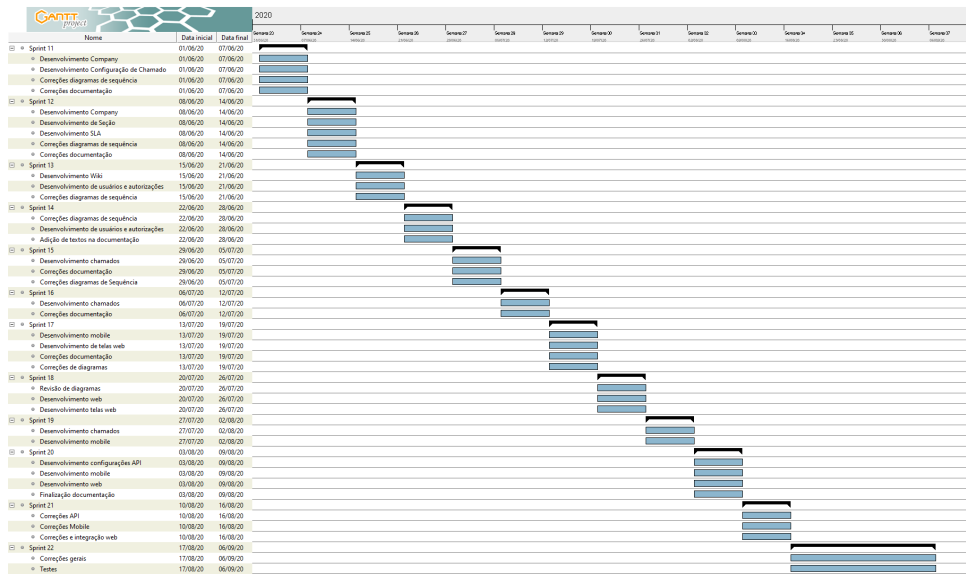


FONTE: Os autores (2019)

A Figura 3.6 apresenta como se deu o planejamento das atividades, disposição durante o tempo e alocação de recursos no projeto através do já explicado diagrama de Gantt. Dessa forma foi possível se ter uma ideia do tempo total de desenvolvimento das atividades, além de informações sobre quais poderiam ser realizadas em paralelo e quais atividades possuíam dependências alheias.

A Figura 3.7 apresenta o planejamento das atividades durante a segunda etapa de desenvolvimento do projeto.

Figura 3.7: DIAGRAMA DE GANTT DA SEGUNDA ETAPA



FONTE: Os autores (2020)

3.4 RESPONSABILIDADES

Durante o desenvolvimento do projeto foi atribuído inicialmente determinadas tarefas para os membros da equipe, porém algumas mudanças foram necessárias e algumas tarefas foram realocadas. A seguir se encontra o quadro relativo às responsabilidades que foram efetivamente realizadas por cada membro da equipe.

QUADRO 3.1: DIVISÃO DE RESPONSABILIDADES

Responsabilidades	
Integrantes	Atividades
01 Leonardo Roberto Rocha	Levantamento de Requisitos; <i>Sprints</i> ; Interfaces; Ambiente de Desenvolvimento; Desenvolvimento <i>backend</i> ; Desenvolvimento <i>frontend</i> ; Integração entre <i>frontend</i> e <i>backend</i> ;
02 Luiz Henrique Dominico	Levantamento de Requisitos; Diagrama de Gantt; Diagrama de Classes; Especificação de Casos de Uso; Documentação; Desenvolvimento <i>mobile</i> ;

03	Marco Aurelio Vieira Olivette	Levantamento de Requisitos; <i>Sprints</i> ; WBS; Diagrama de Classes; Diagrama de Entidade de Relacionamento; Diagrama de Estados; Diagrama de Sequência; Diagrama de Implementação; Diagrama de Casos de Uso; Descrição de Casos de Uso; Especificação de Casos de Uso; Interfaces; Desenvolvimento <i>backend</i> ; Ambiente de Desenvolvimento;
04	Mateus Mendes Zanela	Levantamento de Requisitos; Diagrama de Sequência; Diagrama de Implementação; Diagrama de Casos de Uso; Descrição de Casos de Uso; Especificação de Casos de Uso;
05	Patrick Marcelo Dziura	Levantamento de Requisitos; WBS; Diagrama de Estados; Diagrama de Casos de Uso; Descrição de Casos de Uso; Especificação de Casos de Uso; Documentação; Interfaces;

FONTE: Os autores(2020)

3.5 DESENVOLVIMENTO DO PROJETO

Após todo o planejamento, foi observado a estimativa de tempo para a execução de todas as atividades propostas. As atividades da primeira etapa foram realizadas através da execução de dez *sprints* com duração de uma semana cada, com algumas exceções de *sprints* de maior duração. Após um período sem *sprints*, a segunda etapa foi realizada através da execução de doze *sprints*, sendo todas de uma semana cada e apenas a última tendo duração

maior. As decisões que foram tomadas ao decorrer de cada *sprint* e os artefatos concluídos serão referenciados nesta seção.

3.5.1 Sprint 1

Nesta *sprint* a equipe realizou um *brainstorming* para começar o levantamento de requisitos. Por ter sido realizado o *brainstorming*, muitas ideias surgiram e o sistema ficou grande demais. Foi necessário realizar algumas reuniões para definir os requisitos mais importantes, definir um produto mínimo e o escopo do projeto. Além disso, foi criado um esboço de cronograma para o projeto.

Esta *sprint* foi mais longa devido a estes problemas e a equipe não ter reuniões diárias e teve início no dia 23/07/2019 e terminou dia 12/08/2019. Ao final da *sprint* foi entregue o esboço do cronograma do projeto.

3.5.2 Sprint 2

Nesta *sprint* foram definidos os requisitos do sistema e as responsabilidades dos membros da equipe (QUADRO 3.1). A equipe também alinhou o uso das bibliotecas de boas práticas do ITIL para o escopo do projeto. Além disso, o cronograma de atividades foi transcrito para a ferramenta GanttProject (FIGURA 3.6). A *sprint* teve início no dia 13/08/2019 e terminou dia 19/08/2019 e os requisitos foram entregues nesta *sprint*.

3.5.3 Sprint 3

Esta *sprint* se iniciou no dia 20/08/2019 e se estendeu até o dia 26/08/2019 e a equipe também definiu o uso da biblioteca React para a componentização e desenvolvimento do *frontend* do sistema. Além disso foram desenvolvidos os casos de uso e foram realizadas correções no gráfico de Gantt. Ao final da *sprint* foram entregues os casos de uso (Apêndices A e B) e o gráfico de Gantt corrigido (FIGURA 3.6) para o professor orientador.

3.5.4 Sprint 4

Na *sprint* entre os dias 27/08/2019 e 02/09/2019 se iniciou o desenvolvimento de protótipos de telas para o sistema e foram realizadas correções nos casos de uso entregues anteriormente. Durante a *sprint* foi alinhado com o professor orientador recomendações sobre os casos de uso e correções a serem feitas. Também foi mostrado a prototipação de algumas telas do sistema.

3.5.5 Sprint 5

Na *sprint* do dia 03/09/2019 a 16/09/2019 foi definido a linguagem de programação do *backend* como Java, já que é uma linguagem que todos os membros da equipe possuem

familiaridade, e React para o *frontend* e *mobile*. A equipe também definiu que será necessário realizar o desenvolvimento de um repositório de artigos para o sistema de chamados, uma Wiki, a fim de contemplar o gerenciamento de problemas do ITIL. A prototipação das telas continuou durante esta *sprint*.

3.5.6 Sprint 6

A *sprint* dos dias 17/09/2019 a 30/09/2019 teve objetivo da equipe realizar a revisão dos casos de uso, realizando correções e adicionando casos de uso faltantes em relação a alterações de estados dos chamados. Também foram desenvolvidos os diagramas de classes (APÊNDICE C) e de estados (APÊNDICE F). A prototipação das telas continuou durante esta *sprint* ainda. A equipe entregou os novos casos de uso (APÊNDICE B), os diagramas de classes, de casos de uso (APÊNDICE A) e de estados.

3.5.7 Sprint 7

Nesta *sprint* foram realizadas correções no diagrama de classes e nos casos de uso para corrigir inconsistências e erros. Também teve continuidade o desenvolvimento das telas. As correções foram entregues ao final da *sprint*. Esta etapa durou do dia 01/10/2019 ao dia 14/10/2019 e foram entregues o diagrama de classe (APÊNDICE C) e os casos de uso (APÊNDICE B) corrigidos.

3.5.8 Sprint 8

Na *sprint* dos dias 15/10/2019 a 21/10/2019 foram realizadas as últimas alterações nos casos de uso e no diagrama de classes. Nesta *sprint* teve início o desenvolvimento dos diagramas de sequência e do documento do projeto. No final foram entregues os casos de uso (APÊNDICE B) e diagrama de classes (APÊNDICE C) finalizados, além da prototipação das telas.

3.5.9 Sprint 9

Nesta *sprint* foi realizada uma revisão entre a prototipação das telas e os casos de uso, para padronizar e evitar inconsistências. Também foram desenvolvidos os diagramas de sequência, de entidade-relacionamento e de implementação. Além disso, foram finalizados os capítulos 1, 2 e 3 da documentação do projeto. Esta *sprint* durou do dia 22/10/2019 ao dia 11/11/2019 e ao final foram entregues os diagramas de sequência (APÊNDICE E), de implementação (APÊNDICE D), de entidade-relacionamento (APÊNDICE G) e de classes (APÊNDICE C), além dos capítulos 1, 2 e 3 da documentação.

3.5.10 Sprint 10

Nesta *sprint* foram realizadas as últimas correções nos diagramas de classes, de implementação e de sequência. A documentação do projeto foi finalizada e a equipe desenvolveu um protótipo da aplicação para fins de demonstração. A *sprint* teve início no dia 12/11/2019 e terminou no dia 29/11/2019 e foram entregues os diagramas de classes (APÊNDICE C), de implementação (APÊNDICE D) e de sequência (APÊNDICE E) finalizados com as correções necessárias.

3.5.11 Período sem sprints

Entre o desenvolvimento da primeira parte do projeto, que consiste na documentação, diagramas, casos de uso e protótipos de telas, e a segunda parte do projeto foi utilizado o tempo para discussão do projeto, correções, melhorias e testes. Os testes realizados pela equipe foram para conhecer novas metodologias, tecnologias, linguagens e *frameworks* e então decidir quais delas seriam utilizadas no projeto.

Também foi neste período que foram discutidas sobre as *sprints* e decidida a utilização de *sprints* padrões de uma semana, com as entregas e início da nova *sprint* no mesmo dia. Além disso, foi desenvolvido parte do *frontend*, configurado corretamente o ambiente de desenvolvimento e de produção na Amazon e decidida a utilização de *gitflow* e a função *issues* do GitHub.

Também neste período foi produzido um esboço de fluxograma que foi baseado no diagrama de estados (APÊNDICE F) para facilitar o entendimento e desenvolvimento futuro da aplicação.

3.5.12 Sprint 11

Na *sprint* entre os dias 01/06/2020 e 08/06/2020 foi iniciado o desenvolvimento do projeto utilizando PHP e foram desenvolvidas as funções relacionadas a companhia e configurações de chamados. Além disso, foram ajustados diagramas de sequência (APÊNDICE E) e documentação. Ao final da *sprint* foram entregues as funções relacionadas a companhia.

3.5.13 Sprint 12

Na *sprint* entre os dias 08/06/2020 e 15/06/2020 foram realizados ajustes nas funções relacionadas a companhia e desenvolvidas as funções relacionadas a configuração de SLA (*Service Level Agreement*) e seção. Também foram ajustados diagramas de sequência (APÊNDICE E) e documentação. Foram entregues as funções relacionadas a configurações de chamado e SLA e as funções relacionadas a companhia com os ajustes necessários.

3.5.14 Sprint 13

Nesta *sprint* foi iniciado o desenvolvimento de funções relacionadas a wiki e aos usuários, além de ajustes nos diagramas de sequência (APÊNDICE E). Devido a complexidade com os papéis dos usuários e autorizações, o desenvolvimento não foi finalizado nesta *sprint*. Esta *sprint* durou do dia 15/06/2020 a 22/06/2020.

3.5.15 Sprint 14

Esta *sprint* durou do dia 22/06/2020 ao dia 29/06/2020 e foram finalizadas as funções relacionadas a wiki e usuários. Também foram realizadas alterações na documentação, inclusão de textos nos capítulos 3 e 4 e foi iniciado o desenvolvimento das funções relacionadas a chamados. Ao final da *sprint* foram entregues os novos textos dos capítulos 3 e 4 e as funções relacionadas a wiki e usuários na aplicação.

3.5.16 Sprint 15

Nesta *sprint* o desenvolvimento das funções relacionadas a chamados teve continuação, mas não finalização, e os diagramas de sequência tiveram ajustes, assim como a documentação. Esta *sprint* durou do dia 29/06/2020 ao dia 06/07/2020. Foram entregues os ajustes necessários na documentação nesta *sprint*.

3.5.17 Sprint 16

Na *sprint* entre os dias 06/07/2020 e 13/07/2020 foi finalizado o desenvolvimento das funções relacionadas a chamados e foram realizadas correções na documentação. Foram entregues as funções relacionadas a chamados e os ajustes na documentação ao final desta *sprint*.

3.5.18 Sprint 17

Na *sprint* entre os dias 13/07/2020 e 20/07/2020 foi iniciado o desenvolvimento *mobile* e o desenvolvimento das telas para a aplicação *web*. Também foram ajustados textos da documentação e diagramas.

3.5.19 Sprint 18

Nesta *sprint* foi dada sequência ao desenvolvimento *mobile*, ao desenvolvimento das telas para a aplicação *web* e à revisão de diagramas da aplicação. Esta *sprint* durou do dia 20/07/2020 ao dia 27/07/2020.

3.5.20 Sprint 19

Esta *sprint* durou do dia 27/07/2020 até o dia 03/08/2020 e foi realizado o desenvolvimento das funções relacionadas a chamados da aplicação e também continuou-se o desenvolvimento da aplicação *mobile*. Além disso, foram revisados e alterados os diagramas de sequência. Os itens entregues ao final desta *sprint* foram as funções relacionadas a chamados na aplicação.

3.5.21 Sprint 20

Nesta *sprint* foi concluído o desenvolvimento do *backend* da aplicação *web*, da aplicação *mobile* e da documentação. Esta *sprint* durou do dia 03/08/2020 ao dia 10/08/2020. Ao final desta *sprint* foram entregues o *backend* da aplicação, a aplicação *mobile* e a documentação completa.

3.5.22 Sprint 21

Nesta *sprint* que durou do dia 10/08/2020 ao dia 17/08/2020 foram realizadas correções no *backend* da aplicação. Também foi realizada a integração entre o *frontend* com o *backend* da aplicação *web*. Ao final desta *sprint* foram entregues as correções necessárias e a aplicação integrada.

3.5.23 Sprint 22

Após a *sprint* 21, a equipe usou o tempo restante do projeto para testes e correções de problemas no *backend*, *frontend*, diagramas e documentação. Nesta *sprint* também foi finalizado o esboço do fluxograma (APÊNDICE H) e adicionado na documentação. Ao final deste período, o projeto foi entregue.

4 APRESENTAÇÃO DO SOFTWARE

Neste capítulo é apresentada e detalhada a implementação do projeto. Na próxima seção é primeiramente explicada brevemente a arquitetura do software por completo e então é detalhada nas seções subsequentes a arquitetura *backend* e a arquitetura *frontend*.

4.1 ARQUITETURA DO SOFTWARE

A arquitetura da aplicação se divide em três grandes núcleos, que são eles: *Backend* constituído de uma API RESTful, *frontend* implementado em ReactJS e por fim uma frente *mobile* que compõe o sistema através do React Native.

4.1.1 Arquitetura Backend

A arquitetura do *backend*, que é o setor responsável por lidar com as requisições provenientes tanto da interface *web* quanto da *mobile* e devolver um recurso que satisfaça tal requisição, foi estruturada de acordo com a Figura 4.1.

Na Figura 4.1, o fluxo se inicia com o estímulo da requisição, que passa por uma camada de autenticação. Essa camada filtra todas as requisições verificando se há um *token* de autenticação proveniente do sistema, caso o *token* seja incorreto ou não seja fornecido o recurso simplesmente não é exibido e o usuário recebe uma mensagem de erro. Para as requisições dos usuários que não dispõem desse *token*, há uma recurso "desprotegido" intencionalmente para se realizar a validação.

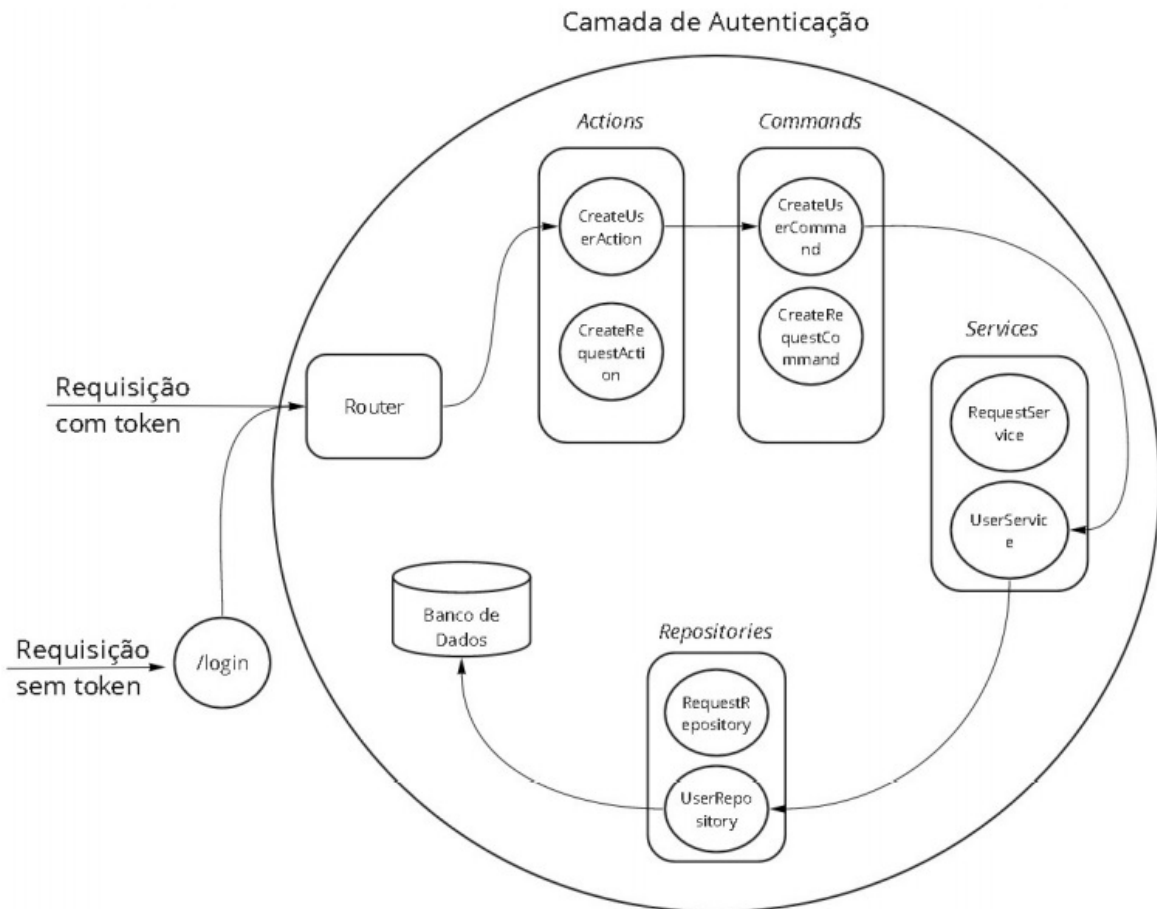
Uma vez validada a requisição submetida pelo usuário, as informações são enviadas para o *router*, que é responsável por realizar o redirecionamento da requisição e invocar o respectivo método responsável.

Após o roteamento da requisição, a mensagem é encaminhada ao *command*, nesse ponto é executado toda a tratativa da requisição, verificando se todos os dados estão presentes e se são consistentes, sem nenhum tipo de informação fora do esperado.

A requisição assim que tratada é encaminha a camada de serviço, que na sua essência tem como responsabilidade lidar com todas as regras de negócio da aplicação, seja elas das mais diversas fontes e complexidade, garantindo que a separação dessas regras do restante do código sejam evidentes e bem claras.

Por fim a mensagem é destinada as classes *Repositories*, responsáveis por serem um ponto de acesso único que encapsulam a lógica de acesso aos dados, garantindo a persistência dos dados e o distanciamento da lógica de negócio da lógica de acesso e manipulação do banco de dados.

Figura 4.1: ARQUITETURA BACKEND



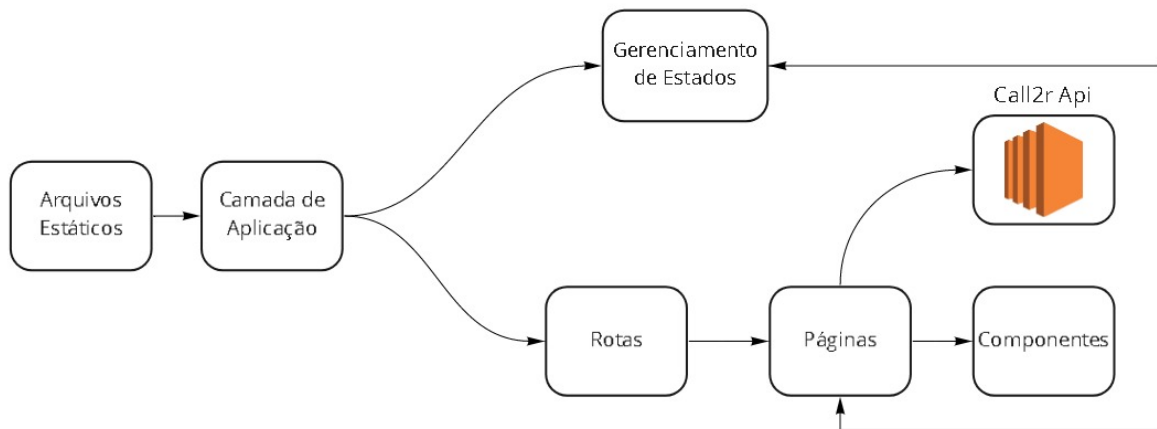
FONTE: Os autores (2020)

Por fim, a mensagem faz seu caminho de volta até a origem do estímulo, onde uma entidade de resposta HTTP é encaminhada. Lembrando que toda a comunicação é realizada através do formato JSON (*JavaScript Object Notation*).

4.1.2 Arquitetura Frontend

A aplicação *web*, assim como observado na Figura 4.2, possui um fluxo que se inicia pela disposição de arquivos estáticos dos tipos HTML, CSS e JS. Essa disposição representa a camada de aplicação da estrutura do ReactJS, que se subdivide em suas respectivas rotas e gerenciamento dos estados. Rotas essas que acessam as páginas presentes na aplicação que são basicamente estruturadas por componentes e responsáveis por realizar as chamadas de recursos de uma API. As páginas e o gerenciamento de estados tem uma relação dupla, ou seja, as páginas influenciam os estados da aplicação assim como os estados da aplicação influenciam as páginas.

Figura 4.2: ARQUITETURA FRONTEND



FONTE: Os autores (2020)

4.1.3 Arquitetura Mobile

A arquitetura da aplicação móvel é pautada na tecnologia do React Native, logo seu funcionamento segue o fluxo de componentes que o React apresenta. A aplicação desenvolvida tem como objetivo ser uma forma de visualização dos estados do chamado, com esse intuito a aplicação é composta de três telas, que são: *Login*, Página Principal e Página de Chamado.

A tela de *login* é responsável por enviar as informações de CPF e senha para a rota *'/login'* da API que devolve as informações de login com o *token* JWT caso seja positivo o retorno. O retorno é tratado pela própria tela de login, que executa uma função de *AsyncStorage* para salvar o *token* recebido da API no armazenamento interno do dispositivo. Caso o login retorne uma falha, a mensagem de erro é apresentada na tela, caso seja um sucesso, o usuário é redirecionado para a página principal, onde a página é construída.

A construção da página principal é realizada através de uma requisição na API, que recebe o *token* que estava salvo no armazenamento do dispositivo e apresenta por fim uma lista com os chamados daquele respectivo usuário. Cada item da lista pode ser selecionado se o desejo do usuário for visualizar mais informações sobre aquele respectivo item.

A página do chamado carrega o item selecionado da lista e apresenta além das informações uma pequena lista interna com os *logs* através do tempo. Para permitir os usuários terem total conhecimento sobre um item.

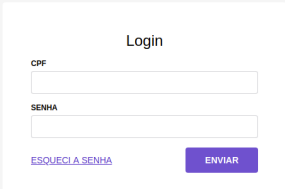
4.2 SOFTWARE

Nesta seção é apresentado o funcionamento do software, primeiramente o sistema *web* e então o sistema *mobile*.

4.2.1 Sistema Web

O sistema apresenta inicialmente a tela de *login* para que o usuário realize a autenticação. Para que o usuário possa acessar a página inicial do sistema, ele deve inserir os dados necessários na tela de *login*, no caso, CPF e senha e estes dados devem estar previamente cadastrados no sistema (FIGURA 4.3).

Figura 4.3: TELA DE LOGIN



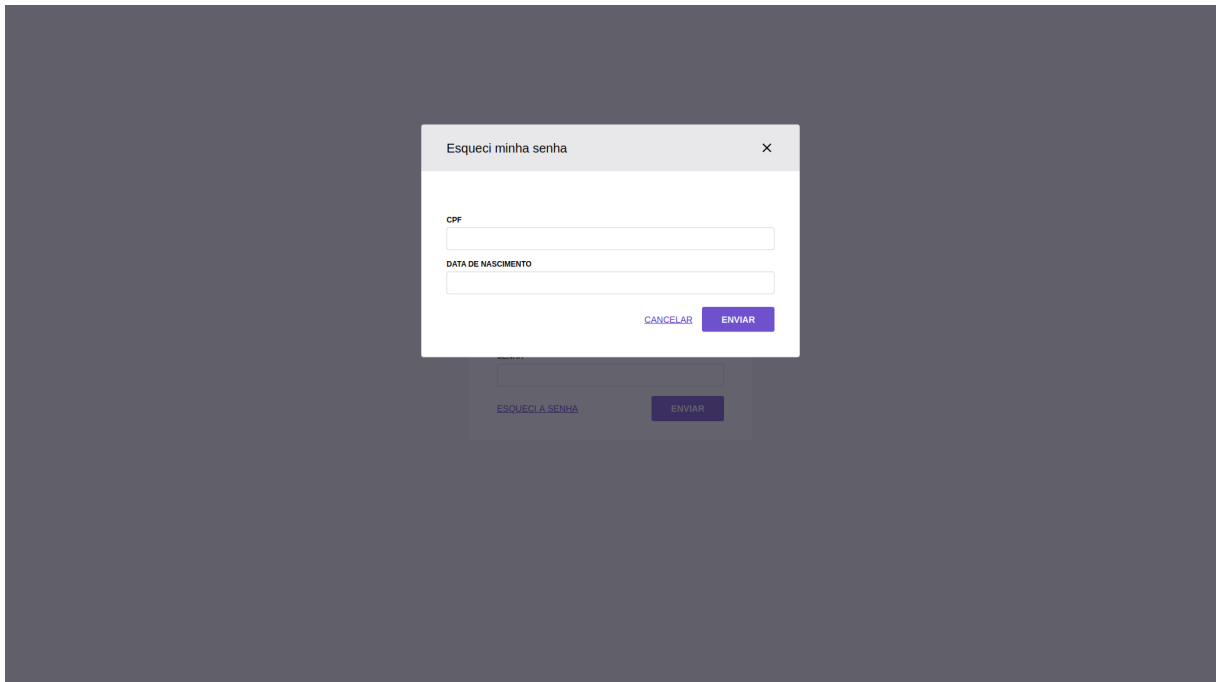
A imagem mostra uma interface de login simples. No topo, o título "Login" está centralizado. Abaixo dele, há dois campos de entrada: o primeiro rotulado "CPF" e o segundo rotulado "SENHA". Abaixo dos campos, há um link azul "ESQUECI A SENHA" e um botão azul "ENVIAR".

FONTE: Os autores (2020)

Caso o usuário não se lembre da senha, é possível clicar na opção Esqueci a senha. O sistema então apresenta a tela de recuperação de senha, apresentada na figura 4.4. O usuário entra com o CPF e a data de nascimento e então o sistema envia um e-mail para o e-mail cadastrado daquele usuário com uma nova senha provisória.

No sistema o usuário pode ser classificado como usuário administrador, usuário comum ou usuário gerente. Além disso, um usuário comum ou usuário gerente pode ser classificado como um suporte ou como um cliente. Cada tipo de usuário possui funcionalidades em comum, mas também funcionalidades exclusivas daquele tipo de perfil de usuário. O usuário administrador apenas administra o sistema. Os usuários gerentes são os gerentes dos usuários comuns, que são os colaboradores daquela empresa. Os usuários suporte são usuários das empresas suporte e os usuários cliente são usuários da empresa mãe.

Figura 4.4: TELA DE RECUPERAÇÃO DE SENHA

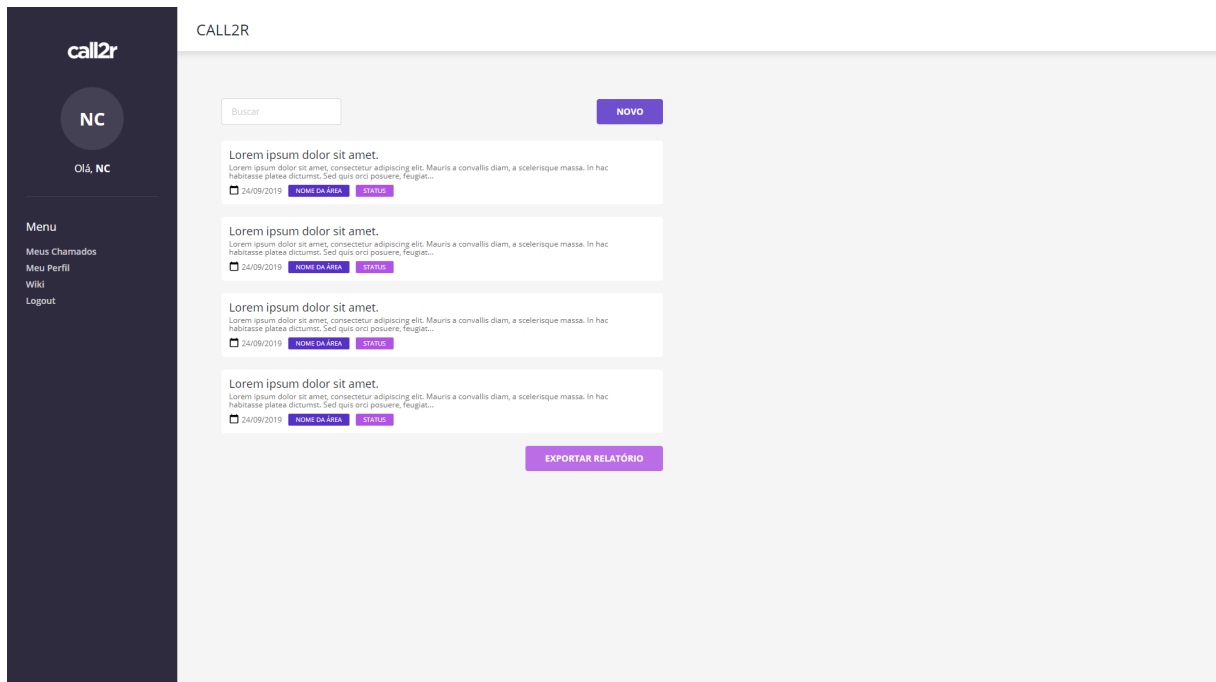
A imagem mostra uma janela modal de recuperação de senha sobreposta a uma tela de login. A janela modal, intitulada "Esqueci minha senha" com um ícone de fechar (X) no canto superior direito, contém dois campos de entrada: "CPF" e "DATA DE NASCIMENTO". Abaixo dos campos, há dois botões: "CANCELAR" em azul claro e "ENVIAR" em azul escuro. Na tela de fundo, o botão "ENVIAR" e o link "ESQUECI A SENHA" são visíveis, mas desbotados.

FONTE: Os autores (2020)

O gerenciamento de chamados é feito pela tela de chamados, apresentada na figura 4.5. Nesta tela são listados os chamados do usuário autenticado no sistema e são listadas opções laterais e em botões. Ao clicar em Novo o usuário pode criar um novo chamado, preenchendo os dados necessários como na figura 4.6. Ao criar um novo chamado, este é criado no estado aguardando suporte.

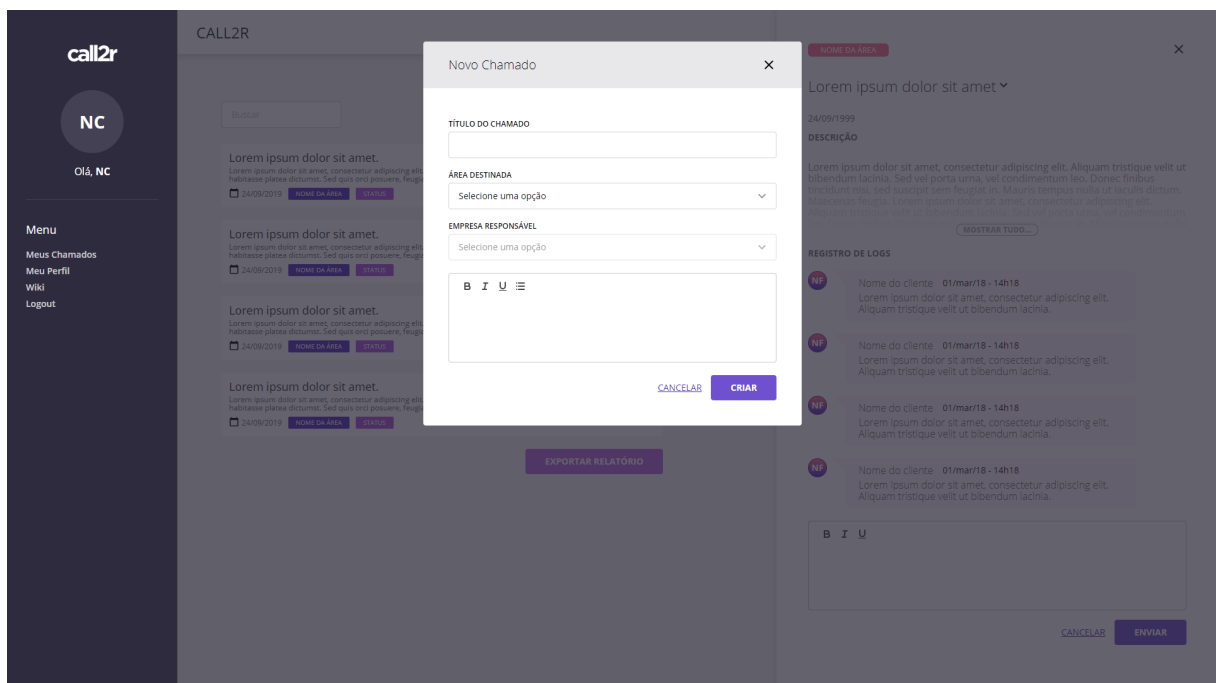
Ao clicar em um chamado, é mostrada a tela de visualização de chamado (FIGURA 4.7) na qual são mostrados os detalhes do chamado, assim como os registros desse chamado. É possível adicionar um registro ao chamado na caixa de texto e então clicando em Enviar.

Figura 4.5: TELA DE LISTA DE CHAMADOS



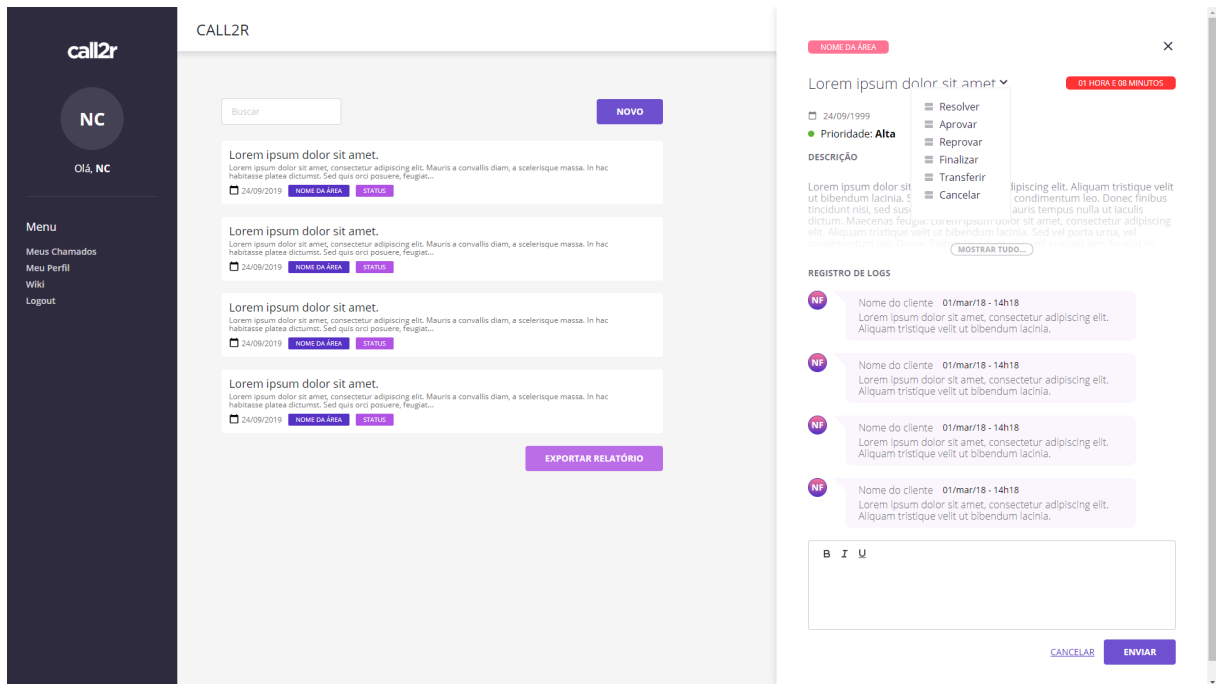
FONTE: Os autores (2020)

Figura 4.6: TELA DE CRIAÇÃO DE CHAMADOS



FONTE: Os autores (2020)

Figura 4.7: TELA DE VISUALIZAÇÃO DE CHAMADO



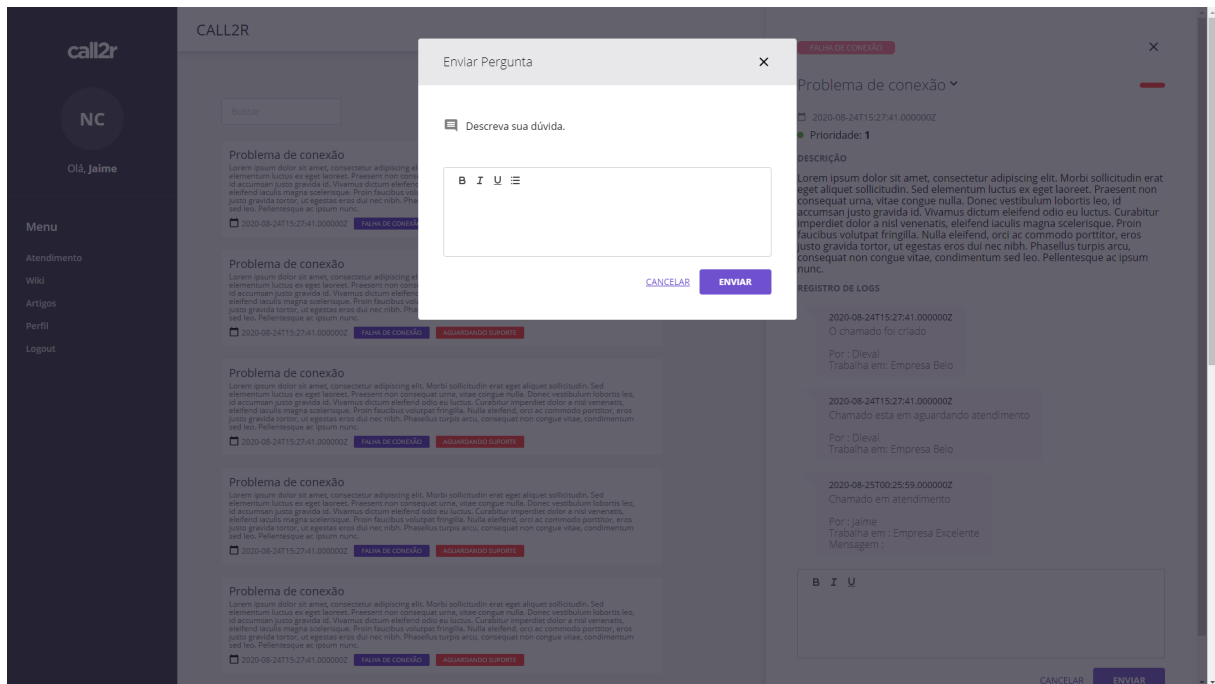
FONTE: Os autores (2020)

Os fluxos de chamados do sistema seguem o fluxograma apresentado no Apêndice H (FIGURA H.1). Os detalhamentos dos fluxos seguem nos próximos parágrafos.

Caso o usuário suporte que esteja realizando o atendimento necessite de mais informações para resolver o chamado, é possível requisitar mais informações para o usuário cliente que solicitou o chamado. Para isso, o usuário suporte adiciona um registro na caixa de texto e então clica em Encaminhar para cliente, assim o chamado retorna para o cliente e tem seu estado alterado para aguardando resposta (FIGURA 4.8). Para a resposta, o fluxo é o mesmo, mas o usuário cliente preenche o texto e então clica em Responder para que o chamado retorne para o estado em atendimento (FIGURA 4.9).

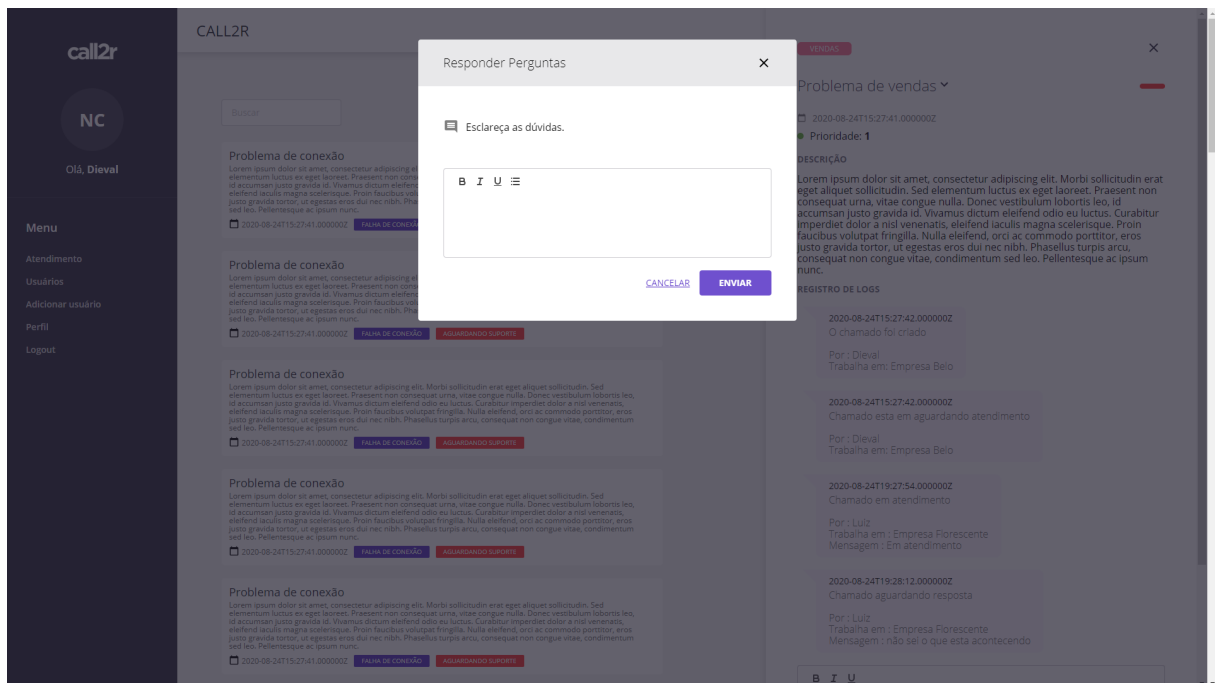
Ao visualizar um chamado e clicar na seta para aparecer as opções, dependendo do perfil do usuário e da situação do chamado, diferentes opções aparecem. As opções possíveis são: resolver, transferir, aprovar, reprovar, finalizar e cancelar. Cada fluxo é detalhado nos parágrafos seguintes.

Figura 4.8: TELA DE ENCAMINHAMENTO PARA CLIENTE



FONTE: Os autores (2020)

Figura 4.9: TELA DE RESPOSTA DO CLIENTE



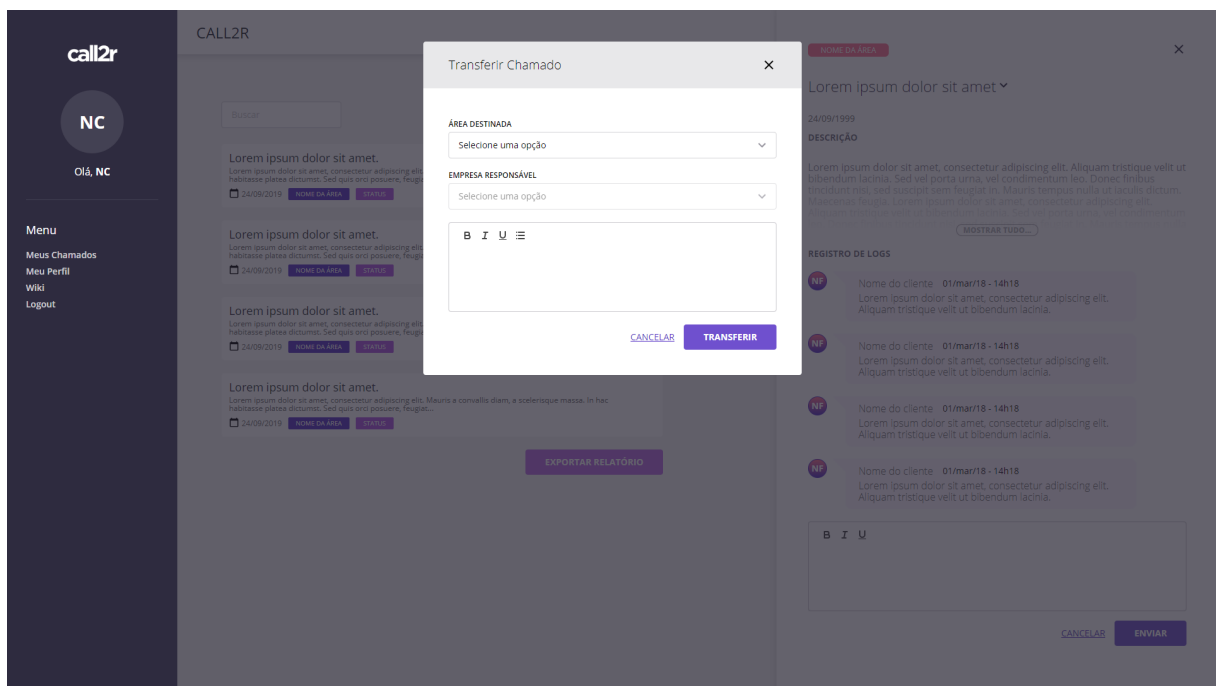
FONTE: Os autores (2020)

O usuário suporte pode visualizar um chamado, clicar na seta para abrir o menu de opções e clicar em Atender. Ao clicar nesta opção, o chamado muda de estado para em

atendimento e o responsável por aquele chamado é alterado para o usuário que realizou a ação. Também é criado um registro referente a esta atualização no chamado.

O usuário suporte também pode transferir o chamado para outra empresa responsável, caso note que o chamado foi atribuído erroneamente para a empresa suporte que trabalha. O usuário clica na seta para abrir o menu de opções e clica em Transferir e então, na nova tela que é apresentada como na figura 4.10, preenche os dados necessários e então em transferir. O sistema altera a empresa responsável pelo chamado e cria um registro dessa alteração.

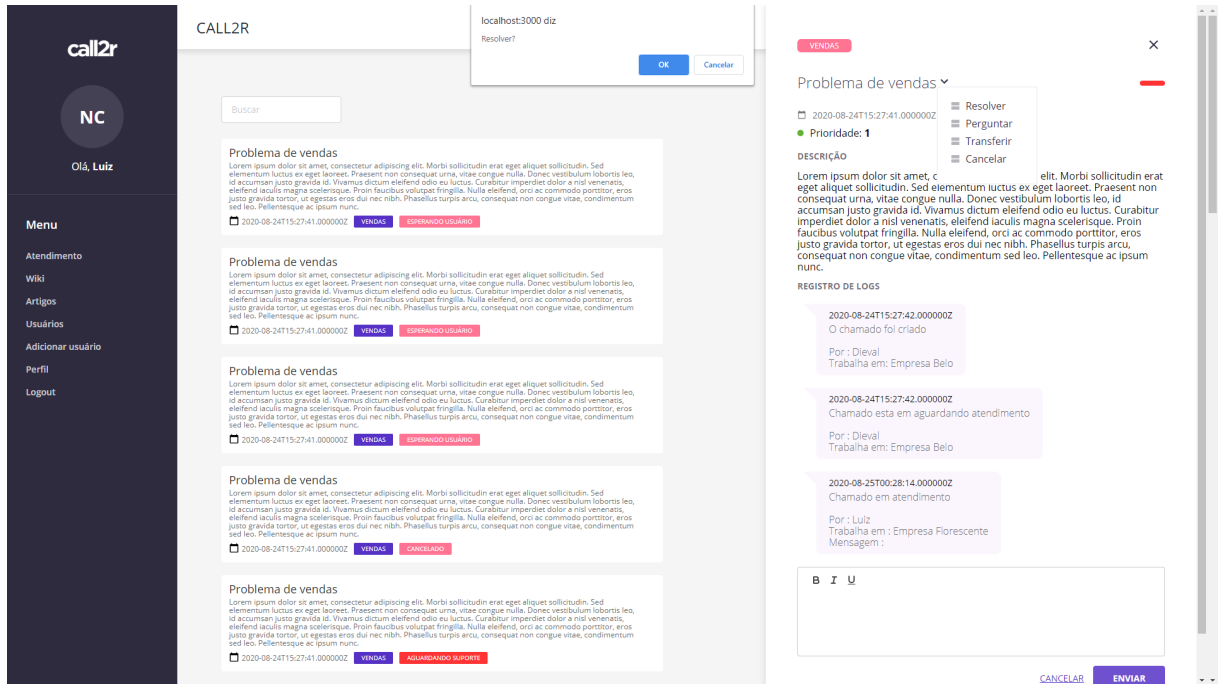
Figura 4.10: TELA DE TRANSFERÊNCIA DE CHAMADO



FONTE: Os autores (2020)

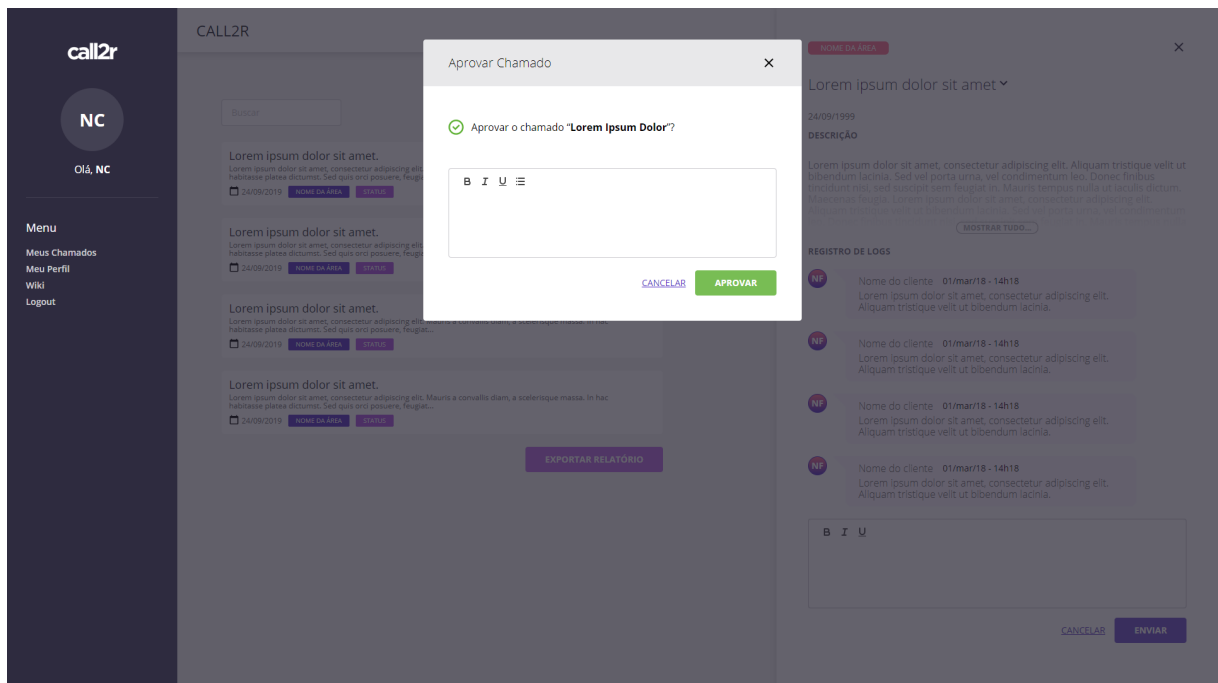
Após o atendimento, o usuário suporte clica na opção Resolver para que o chamado tenha o estado alterado para aguardando suporte e retorne para a lista de chamados do usuário cliente que solicitou este chamado (FIGURA 4.11). Depois deste ponto, o usuário cliente pode aprovar ou reprovar como na figura 4.12 e na figura 4.13, respectivamente. Em ambos os casos é criado um registro da alteração realizada no chamado.

Figura 4.11: TELA DE RESOLUÇÃO DE CHAMADO



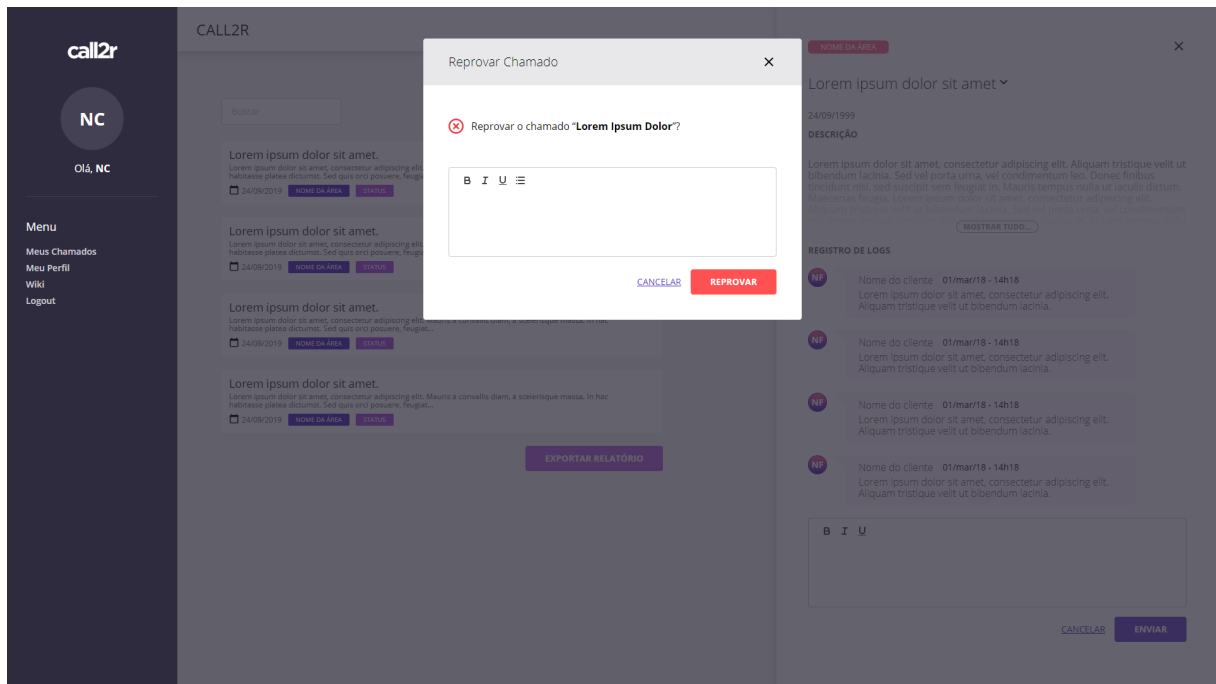
FONTE: Os autores (2020)

Figura 4.12: TELA DE APROVAÇÃO DE CHAMADO



FONTE: Os autores (2020)

Figura 4.13: TELA DE REPROVAÇÃO DE CHAMADO

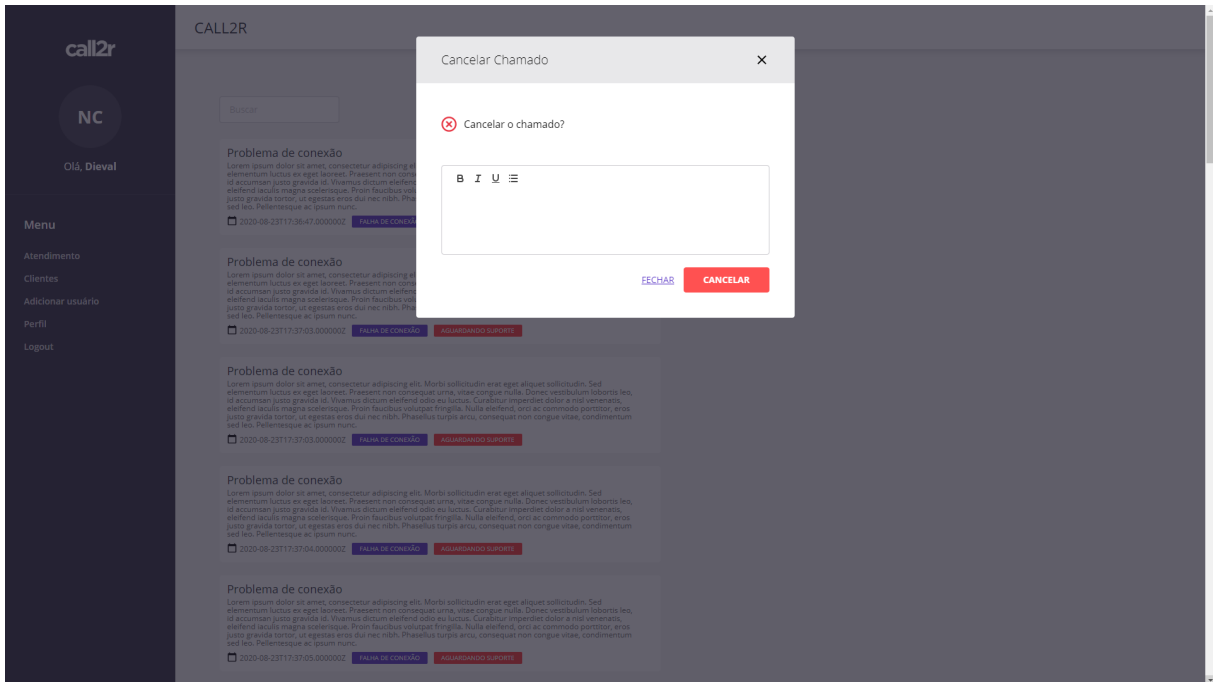


FONTE: Os autores (2020)

Nas opções também existe a opção cancelar para que o gerente suporte possa realizar o cancelamento de um chamado que está errado, foi aberto erroneamente, foi aberto duplicado ou por outro motivo. Para isso, o gerente suporte deve acessar a opção cancelar, preencher a justificativa e cancelar (FIGURA 4.14). O estado do chamado é alterado para cancelado e não há possibilidade de novas alterações.

Outra opção existente é a opção de priorizar um chamado. O usuário gerente cliente pode usar esta opção para que o chamado seja destacado na lista de chamados e para informar ao usuário suporte que estiver realizando atendimento que o chamado foi priorizado e portanto, deve ser dada uma maior importância a ele.

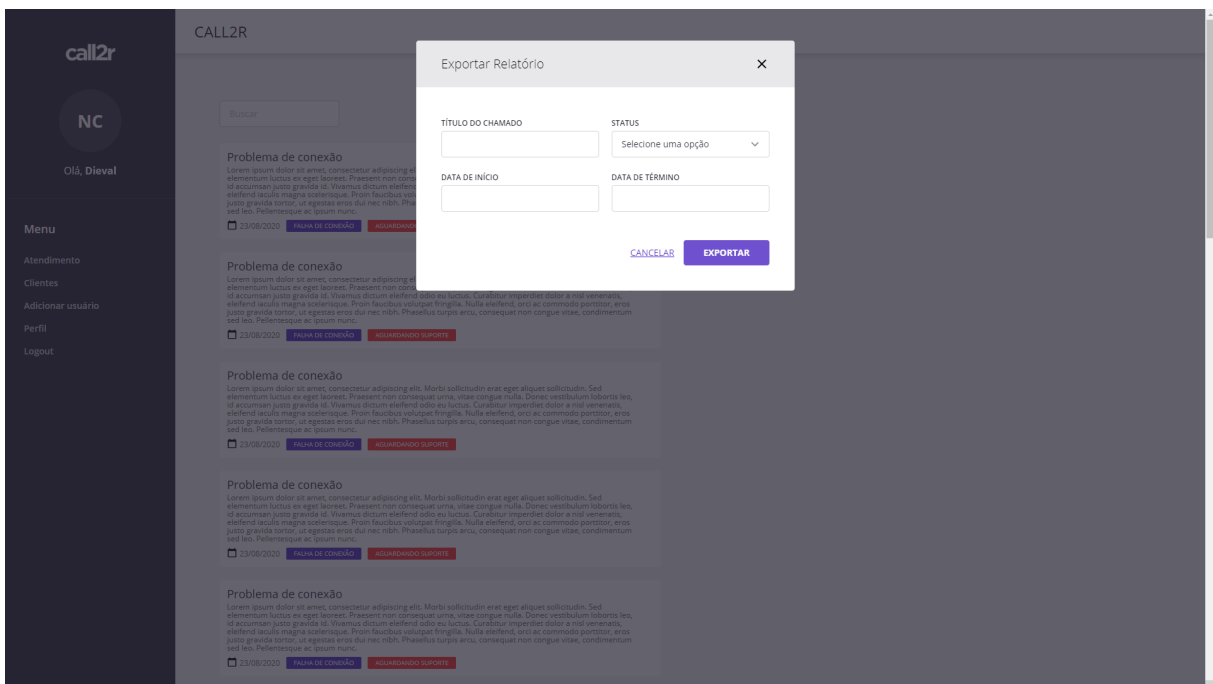
Figura 4.14: TELA DE CANCELAMENTO



FONTE: Os autores (2020)

Os usuários do tipo gerente também podem exportar relatórios do chamados no formato PDF (FIGURA 4.15), podendo selecionar um período, palavra-chave no título do chamado e/ou o estado dos chamados que serão exportados para o relatório.

Figura 4.15: TELA DE EXPORTAR RELATÓRIOS



FONTE: Os autores (2020)

No menu lateral existe a opção Meu perfil, na qual o usuário pode acessar e editar seu perfil como na figura 4.16. Nesta tela o usuário pode acessar a opção trocar senha e então alterar sua senha, digitando a senha antiga para confirmação, a senha nova e então a nova senha para confirmar (FIGURA 4.17).

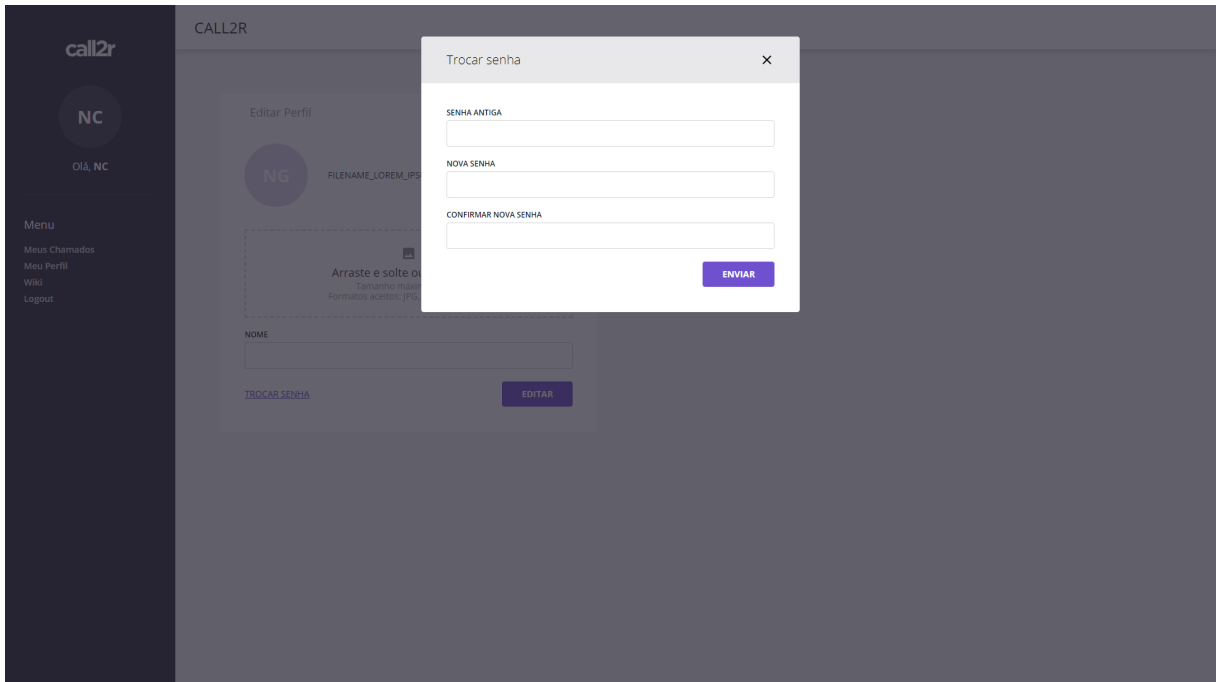
Usuários com perfil administrador e gerente podem visualizar listas de usuários pela opção no menu lateral. A diferença na listagem é que usuários do tipo administrador podem visualizar usuários do tipo gerente (FIGURA 4.18) e usuários do tipo gerente podem visualizar usuários do mesmo tipo, cliente ou suporte, e que estejam atribuídos na mesma empresa que ele (FIGURA 4.19).

Figura 4.16: TELA DE PERFIL

FONTE: Os autores (2020)

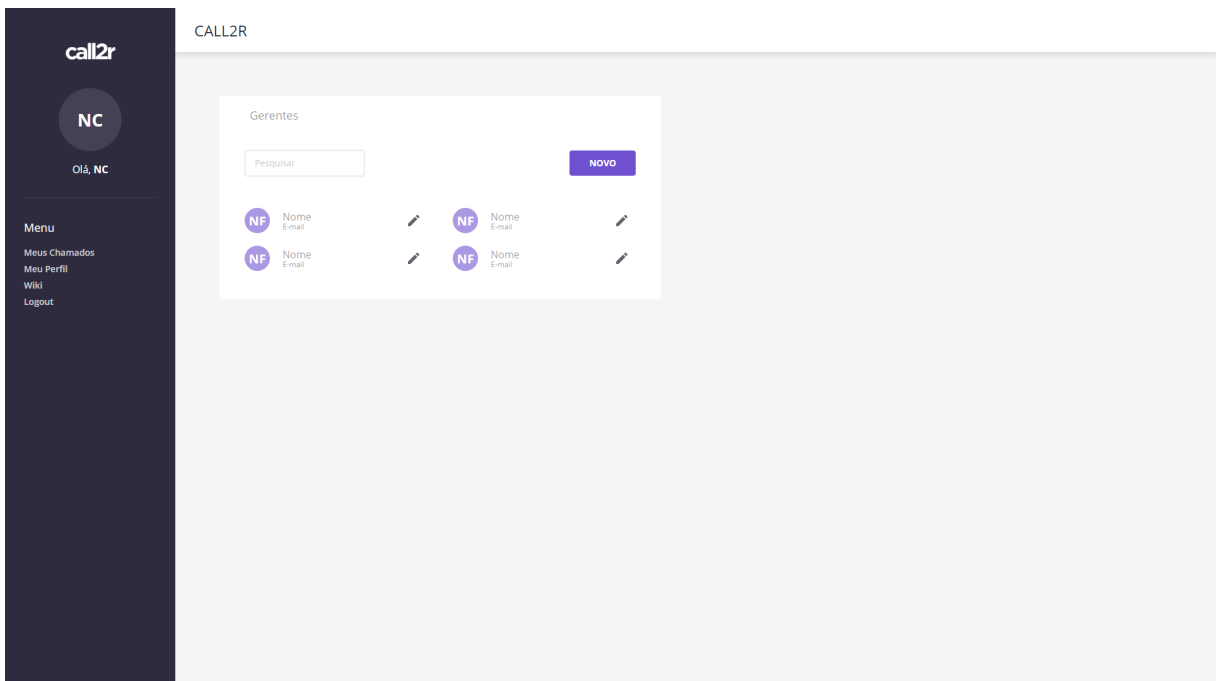
No sistema é possível criar usuários, mas apenas usuários com perfil gerente e administrador podem realizar esta ação. O usuário administrador pode criar usuários para qualquer empresa (FIGURA 4.20), mãe ou suporte, enquanto que o gerente pode criar usuários apenas para a empresa na qual está atribuído (FIGURA 4.21).

Figura 4.17: TELA DE ALTERAÇÃO DE SENHA



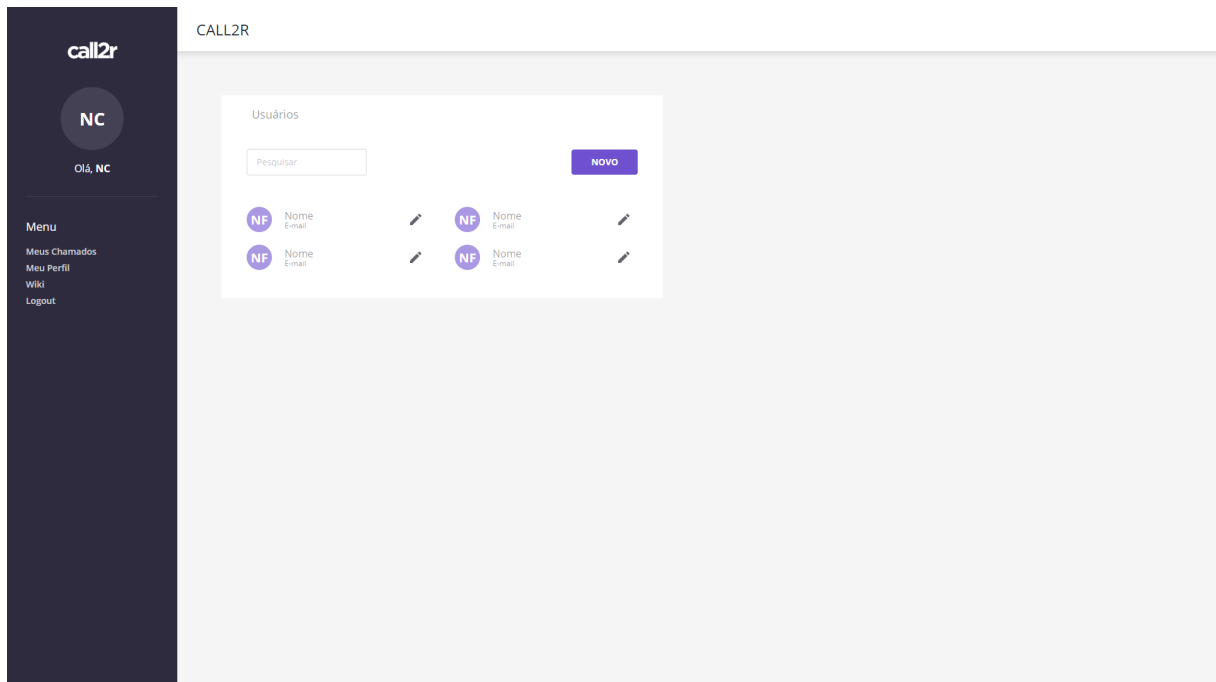
FONTE: Os autores (2020)

Figura 4.18: TELA DE LISTA DE GERENTES



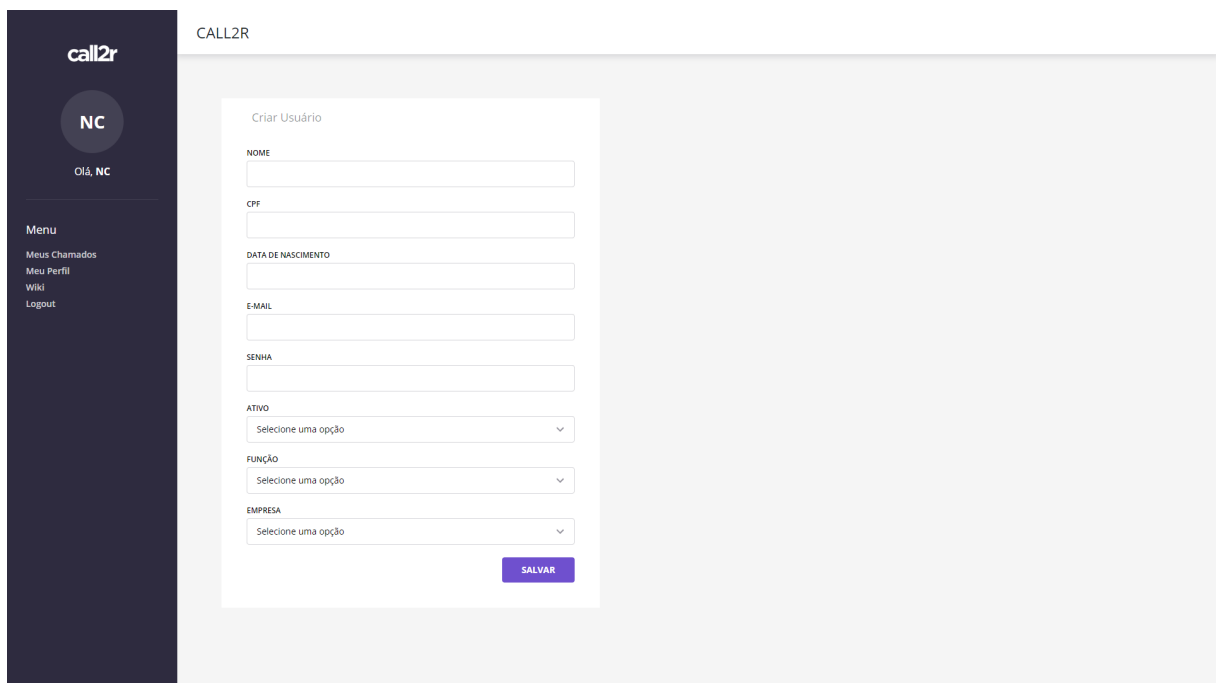
FONTE: Os autores (2020)

Figura 4.19: TELA DE LISTA DE USUÁRIOS



FONTE: Os autores (2020)

Figura 4.20: TELA DE CRIAÇÃO DE USUÁRIO POR ADMINISTRADOR



FONTE: Os autores (2020)

Figura 4.21: TELA DE CRIAÇÃO DE USUÁRIO POR GERENTE

CALL2R

call2r

NC

Olá, NC

Menu

- Meus Chamados
- Meu Perfil
- Wiki
- Logout

Criar Usuário

NOME

CPF

DATA DE NASCIMENTO

E-MAIL

SENHA

ATIVO

Selecione uma opção

FUNÇÃO

Selecione uma opção

SALVAR

FONTE: Os autores (2020)

Também é possível alterar o usuário através da opção editar. Alguns dados não são disponibilizados para edição, como CPF. Nesta mesma tela é possível ativar ou desativar um usuário, caso ele esteja ativo ou não na empresa que está atribuído (FIGURA 4.22).

Figura 4.22: TELA DE EDIÇÃO DE USUÁRIO

CALL2R

call2r

NC

Olá, NC

Menu

- Meus Chamados
- Meu Perfil
- Wiki
- Logout

Editar Usuário

NOME

E-MAIL

CPF

DATA DE NASCIMENTO

ATIVO

Sim

FUNÇÃO

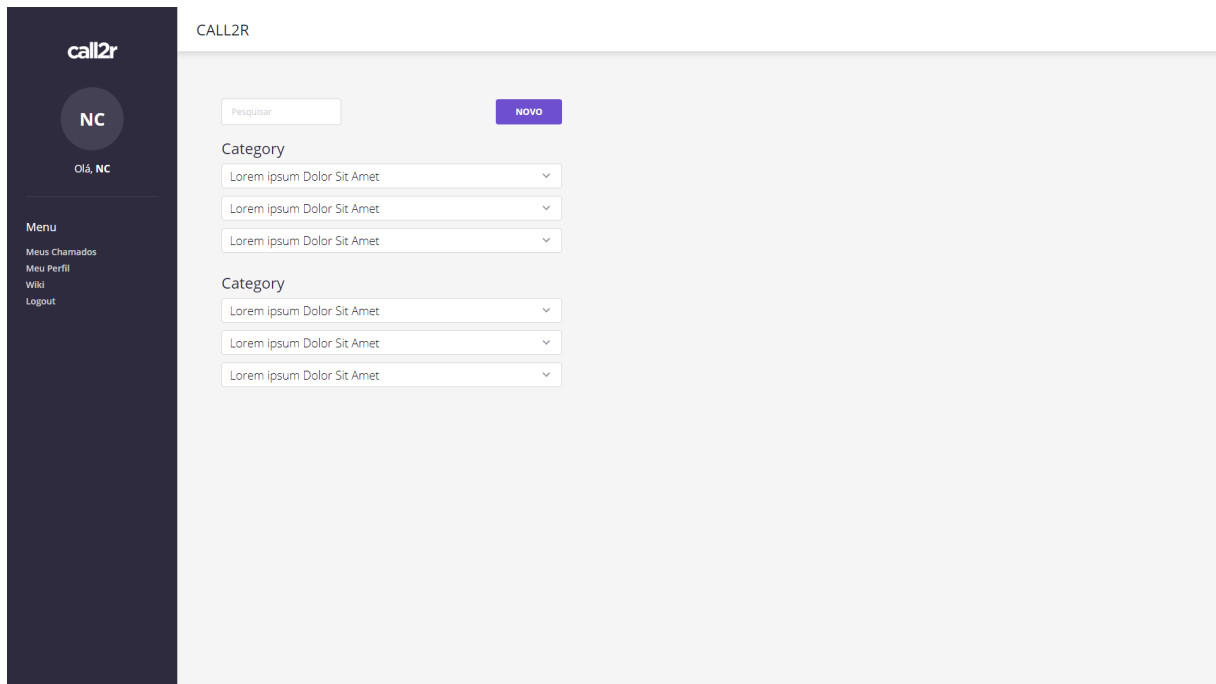
Usuário Suporte

EDITAR

FONTE: Os autores (2020)

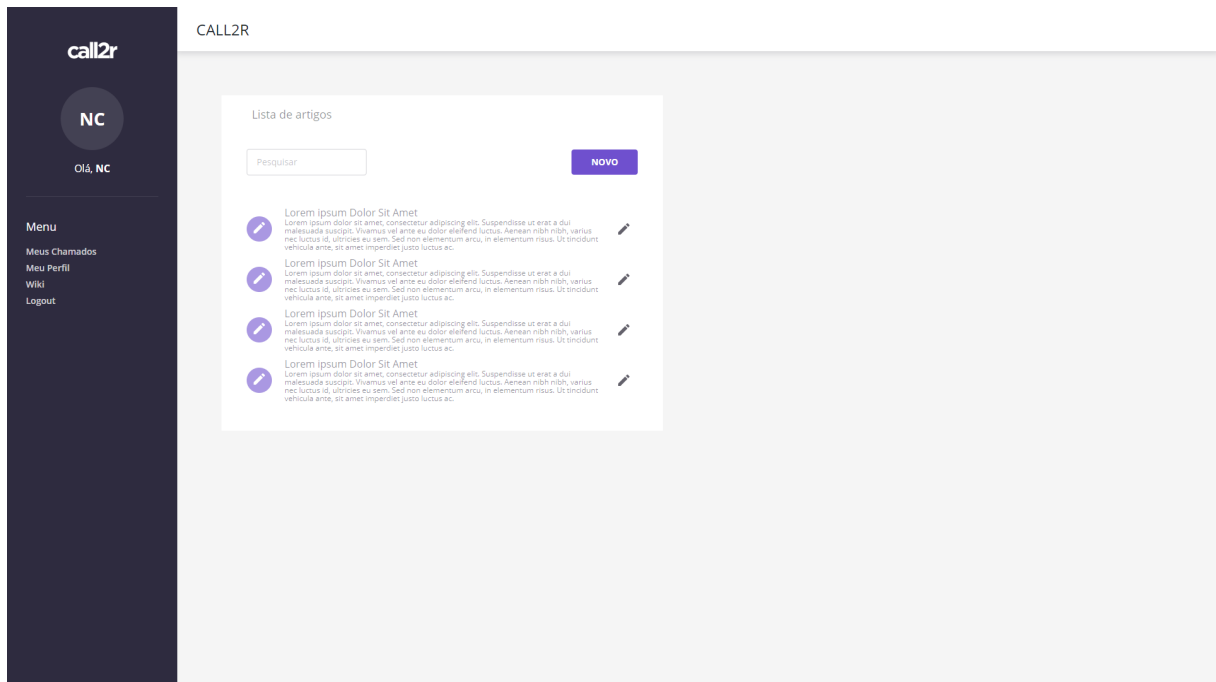
Também existe a opção *wiki* no menu lateral para usuários suporte. Na *wiki* é possível o usuário visualizar artigos separados por categorias. O objetivo da *wiki* é que os usuários suporte possam registrar os passos para resolver problemas, evitando que quando outro suporte for solucionar um mesmo problema ou problema parecido, perca tempo procurando soluções ou testando soluções que não funcionam (FIGURA 4.23). Ao clicar em uma categoria, o sistema apresenta a lista de artigos referentes àquela categoria de artigos (FIGURA 4.24).

Figura 4.23: TELA DE WIKI



FONTE: Os autores (2020)

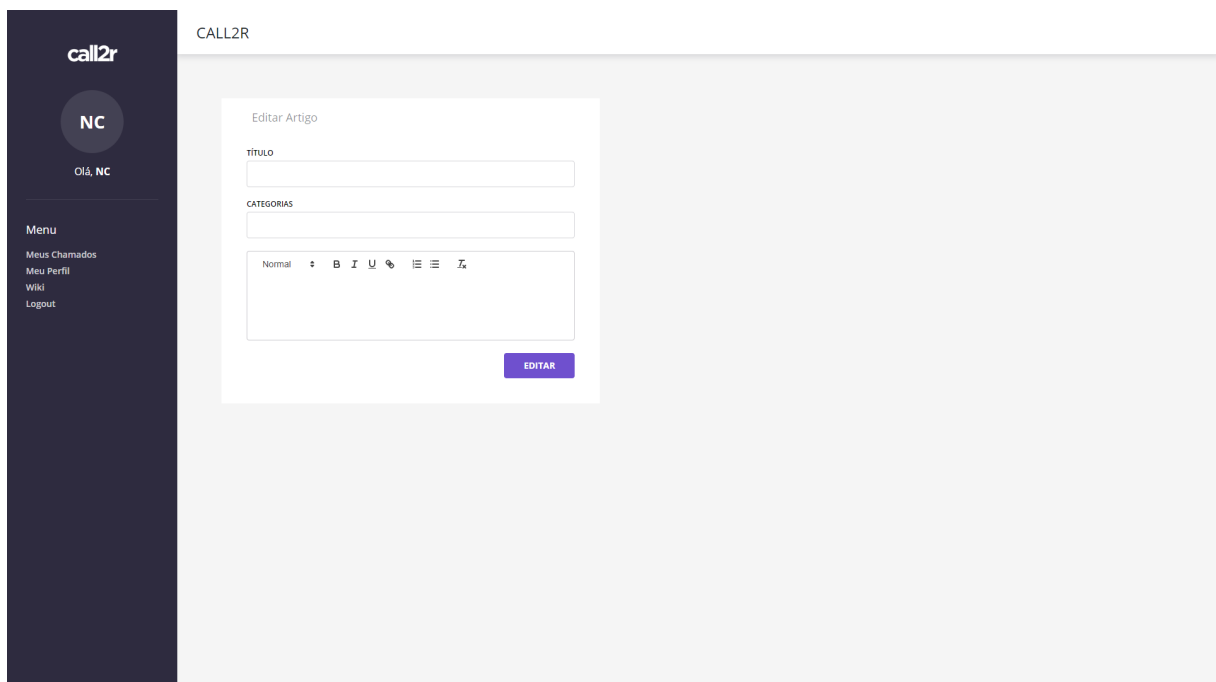
Figura 4.24: TELA DE LISTA DE ARTIGOS



FONTE: Os autores (2020)

Clicando em Novo, é apresentada a tela para inserir um novo artigo, no qual o usuário pode inserir um título, categoria e o texto referente ao artigo (FIGURA 4.25). Ao editar um artigo é apresentada a mesma tela, apenas com os dados do artigo preenchidos.

Figura 4.25: TELA DE EDIÇÃO DE ARTIGO



FONTE: Os autores (2020)

No sistema também existe o cadastro e alterações de empresas que são funções exclusivas ao usuário administrador. O sistema possui apenas uma empresa mãe, mas pode ter várias empresas suportes. Na tela de criação e edição de empresas também é possível configurar o tempo de SLA de acordo com as prioridades dos chamados e esta configuração é feita em horas (FIGURA 4.26). O tempo do SLA configurado é o tempo que o usuário suporte tem para atender e resolver um chamado, dependendo da prioridade do mesmo. O usuário administrador também pode visualizar uma lista das empresas cadastradas no sistema (FIGURA 4.27).

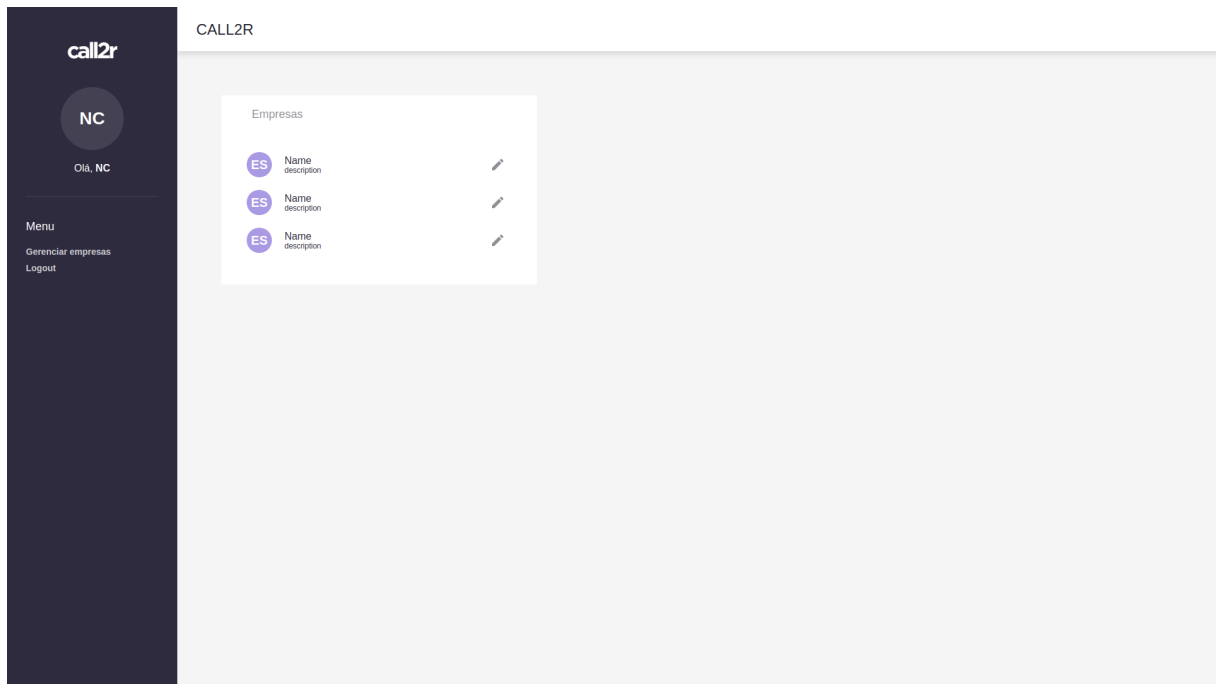
Também existe o cadastro de áreas na mesma tela. O usuário suporte pode digitar o nome da área no campo área e então clica em Adicionar área e esta área é criada. Ao digitar o nome da área, caso ela exista, o sistema apresenta os resultados, funcionando como um campo de pesquisa.

Figura 4.26: TELA DE EDIÇÃO DE EMPRESA

A imagem mostra a interface de usuário para a edição de uma empresa no sistema call2r. O cabeçalho superior à esquerda contém o logo 'call2r' e o nome de usuário 'NC' com o texto 'Olá, NC' abaixo dele. Um menu lateral à esquerda oferece opções: 'Menu', 'Meus Chamados', 'Meu Perfil', 'Wiki' e 'Logout'. O conteúdo principal da página é o formulário 'Editar Empresa', que inclui campos de entrada para: 'NOME DA EMPRESA', 'CNPJ', 'DESCRIÇÃO', e cinco configurações de SLA (BAIXÍSSIMA, BAIXA, MÉDIA, ALTA e ALTÍSSIMA PRIORIDADE). Abaixo desses campos, há uma seção 'Áreas' com um campo 'ÁREA 1' e um link 'ADICIONAR NOVA'. Um botão 'SALVAR' está localizado na base do formulário.

FONTE: Os autores (2020)

Figura 4.27: TELA DE LISTA DE EMPRESAS

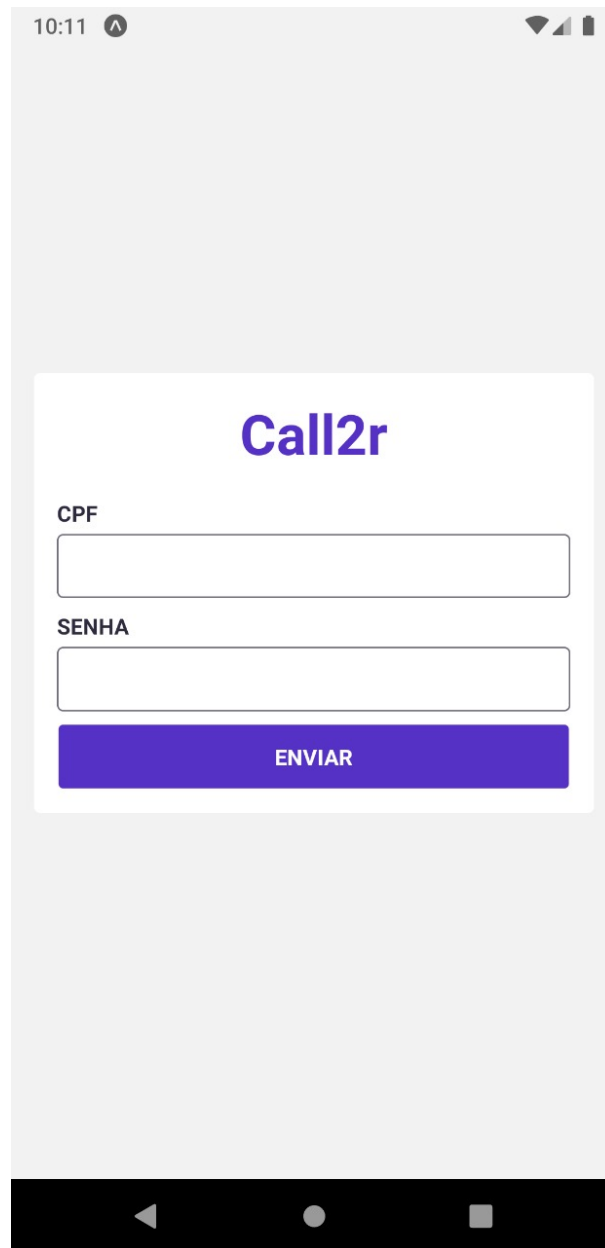


FONTE: Os autores (2020)

4.2.2 Sistema Mobile

O sistema *mobile* tem como finalidade apenas a visualização de chamados. Inicialmente é apresentado ao usuário a tela de *login* para que o usuário realize a autenticação, da mesma forma que é realizado no sistema *web*. O usuário acessa utilizando seu CPF e senha, que previamente já estavam cadastrados no sistema (FIGURA 4.28).

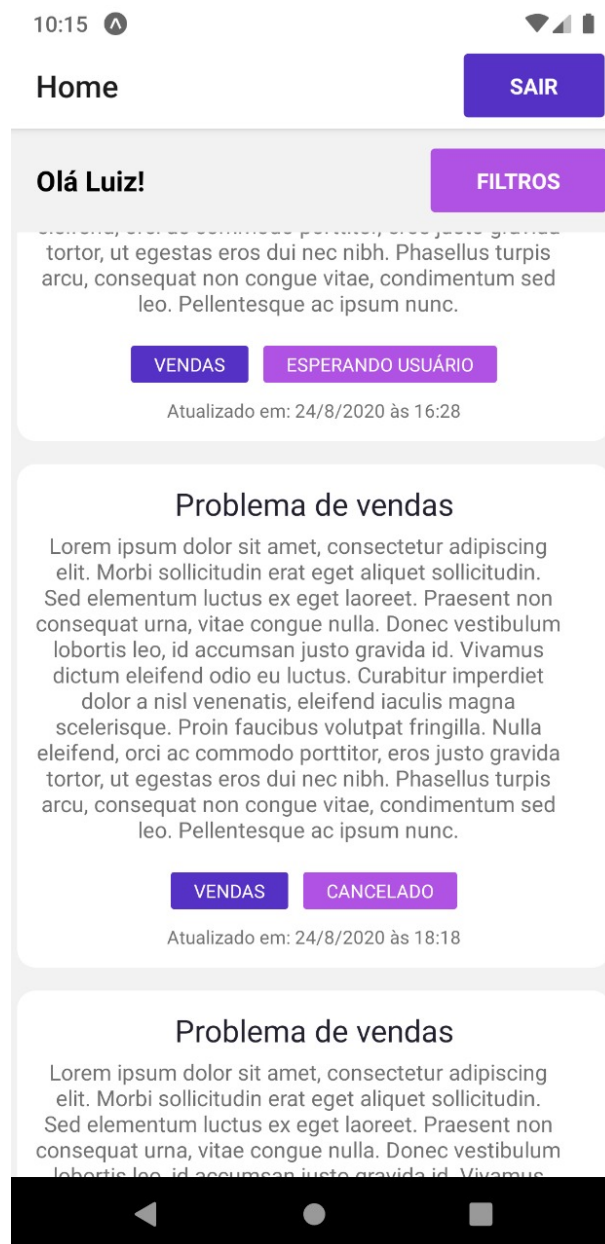
Figura 4.28: TELA DE LOGIN MOBILE

A mobile login screen for the application 'Call2r'. The screen features a white background with a purple header area containing the 'Call2r' logo. Below the logo, there are two input fields: one for 'CPF' and one for 'SENHA'. A purple button labeled 'ENVIAR' is positioned below the input fields. The screen is framed by a grey border, and the top status bar shows the time '10:11' and various icons. The bottom navigation bar is black with standard Android navigation icons.

FONTE: Os autores (2020)

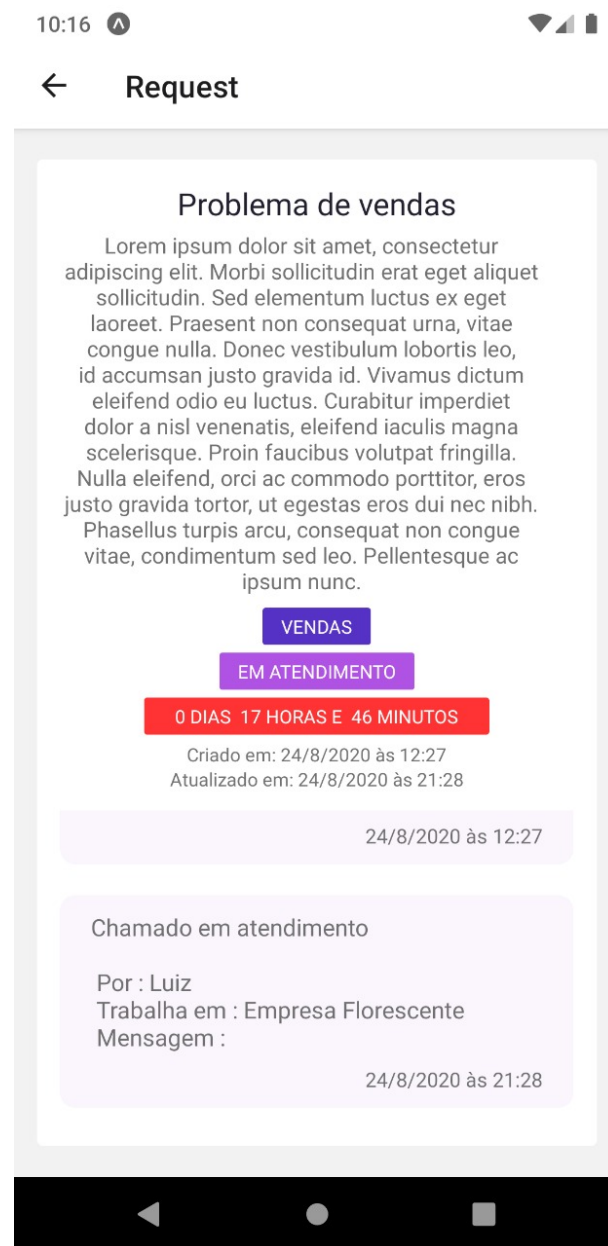
Após realizar a autenticação, é apresentado ao usuário uma tela com a lista de chamados do usuário (FIGURA 4.29). A partir desta lista o usuário pode visualizar algum chamado clicando nele e então são mostrados os detalhes do chamado, incluindo os *logs* do chamado (FIGURA 4.30).

Figura 4.29: TELA DE LISTA DE CHAMADOS MOBILE



FONTE: Os autores (2020)

Figura 4.30: TELA DE VISUALIZAR CHAMADO MOBILE



FONTE: Os autores (2020)

Na tela de lista de chamados é possível o usuário filtrar chamados de acordo com os filtros pré-definidos para então diminuir o tamanho da lista de chamados e facilitar a visualização (FIGURA 4.31).

Figura 4.31: TELA DE FILTROS MOBILE



FONTE: Os autores (2020)

4.3 CONSIDERAÇÕES À APRESENTAÇÃO DO SOFTWARE

As interfaces apresentadas neste capítulo representam todas as funções do sistema implementadas. Algumas telas sofreram alterações em comparação com os protótipos desenvolvidos anteriormente e outras telas deixaram de existir e tiveram suas funcionalidades implementadas juntas com outras telas.

As telas que sofreram alteração foram alteradas para corrigir problemas que existiam no protótipo e/ou para adaptar a tela à implementação realizada do projeto. O motivo para implementar as funcionalidades de algumas telas em outras e retirá-las do projeto foi para diminuir a quantidade de telas e tentar diminuir a complexidade do projeto, já que é um projeto grande.

O foco do projeto e a complexidade do projeto foi no *backend* da aplicação, para ter uma API completa e complexa e por isso o *frontend* é funcional e usável apenas. Um exemplo de implementação pendente para o *frontend* é a paginação nas listagens de chamados e usuários devido à complexidade e tempo para implementação. Por este motivo a equipe decidiu que era um item de baixa prioridade.

5 CONSIDERAÇÕES FINAIS

Ao final do desenvolvimento do projeto o sistema Call2R foi planejado e desenvolvido, cumprindo os objetivos propostos anteriormente. O sistema foi desenvolvido com o *backend* como API, um *frontend web* e uma aplicação *mobile*. Foi possível observar que mesmo com tempo, a equipe necessitava de um planejamento melhor para desenvolver diagramas, já que, certos diagramas foram corrigidos ao final do desenvolvimento do projeto ainda.

A equipe também obteve mais conhecimento do desenvolvimento real de uma aplicação e de novas tecnologias pois no período sem *sprints* foram estudadas tecnologias, linguagens e *frameworks* para decidir a melhor abordagem e implementação do projeto. Neste período, foi observado que a linguagem Java, escolhida inicialmente, não supriria as necessidades da equipe por causa da complexidade e, conseqüentemente, do tempo necessário para a implementação.

Durante a etapa de desenvolvimento do *backend* e do *frontend* da aplicação a equipe se preocupou em seguir as especificações de casos de uso e outros itens já estabelecidos durante a primeira etapa de desenvolvimento do projeto. Esta preocupação da equipe foi para evitar que a documentação seguisse sendo alterada, impactando no próprio desenvolvimento da aplicação e em outros capítulos da documentação.

Por causa das reuniões semanais da segunda etapa do projeto, houve certa dificuldade em todos os membros da equipe participarem das reuniões todas as semanas, nestes casos, houve disciplina da equipe para entregar os itens propostos para aquela *sprint*, mesmo o integrante responsável não poder participar da reunião. Também houveram reuniões remarcadas de segunda-feira para terça-feira ou quarta-feira, mas os itens da *sprint* entreguem na segunda-feira, fazendo com que uma *sprint* não interferisse com a outra.

5.1 RECOMENDAÇÕES PARA TRABALHOS FUTUROS

Após a conclusão da primeira etapa é claro a noção de alguns pontos que indicam a continuidade do projeto, com o objetivo de melhorar as atividades já realizadas, principalmente o trabalho de equipe e comunicação, e realizar a implementação do sistema propriamente dito.

Após a conclusão da segunda etapa do projeto, ainda foi possível observar possíveis melhoras como melhor planejamento do projeto, planejamento de tempo, organização e comunicação entre os membros da equipe.

Também é possível aumentar o escopo da aplicação *mobile* futuramente, permitindo o usuário realizar todas as funcionalidades da aplicação *web*. Esta mudança será possível de forma relativamente fácil, já que o *backend* da aplicação é uma API.

REFERÊNCIAS

ADOBE. **Adobe XD**. 2019. Disponível em:

<<https://www.adobe.com/br/products/xd.html>>. Acessado em 12 de nov. de 2019.

AMAZON. **Announcing Amazon Elastic Compute Cloud (Amazon EC2) - beta**. 2006.

Disponível em: <<https://aws.amazon.com/pt/about-aws/whats-new/2006/08/24/announcing-amazon-elastic-compute-cloud-amazon-ec2---beta/>>. Acessado em 10 de set. de 2020.

AMAZON. **Armazenamento S3 - Simple Storage Services - Amazon Web Services**. 2020a.

Disponível em: <<https://aws.amazon.com/pt/s3/>>. Acessado em 03 de jul. de 2020.

AMAZON. **Computação em nuvem com a AWS**. 2020b. Disponível em:

<https://aws.amazon.com/pt/what-is-aws/?nc2=h_q1_le_int>. Acessado em 10 de set. de 2020.

AMAZON. **Conceitos básicos do Amazon Simple Storage Service - Amazon Web Services**.

2020c. Disponível em: <https://docs.aws.amazon.com/pt_br/AmazonS3/latest/gsg/GetStartedWithS3.html>. Acessado em 03 de jul. de 2020.

AMAZON. **O que é DevOps?** 2019. Disponível em:

<<https://aws.amazon.com/pt/devops/what-is-devops/>>. Acessado em 25 de nov. de 2019.

ASTAH. **Astah UML**. 2019. Disponível em: <<http://astah.net/>>. Acessado em 08 de nov. de 2019.

ATLASSIAN. **IT metrics: 4 best practices for success**. 2020. Disponível em:

<<https://www.atlassian.com/itsm/service-request-management/it-metrics-and-reporting>>. Acessado em 25 de jul. de 2020.

AUDY, Jorge. **Scrum 360**. Rio Grande do Sul: Casa do Código, 2015.

BANKS, Alex; PORCELLO, Eve. **Learning React Functional Web Development with React and Redux**. California: O'Reilly, 2017.

BARRETO, Rafael. **Cliente quer preço ou qualidade**. 2015. Disponível em: <https://www.youtube.com/watch?v=ri7WfcqavK0&ab_channel=SebraeSP>.

Acessado em 04 de nov. de 2020.

BAY, Jeff. **Object Calisthenics**. 2008. Disponível em:

<<https://www.cs.helsinki.fi/u/luontola/tdd-2009/ext/ObjectCalisthenics.pdf>>. Acessado em 30 de jun. de 2020.

- BECK, Kent et al. **Manifesto para o desenvolvimento ágil de software**. 2001. Disponível em: <<https://www.manifestoagil.com.br/>>. Acessado em 24 de nov. de 2019.
- BITNER, Mary Jo; BOOMS, Bernard H.; TETREAUULT, Mary Stanfield. **The Service Encounter: Diagnosing Favorable and Unfavorable Incidents**. Estados Unidos: Journal of Marketing, 1990.
- BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivar. **The Unified Modeling Language User Guide**. Estados Unidos: Addison-Wesley, 2005.
- CHACON, Scott; GIT. **About**. 2005b. Disponível em: <<https://git-scm.com/about>>. Acessado em 17 de mai. de 2020.
- CHACON, Scott; GIT. **Getting Started - A Short History of Git**. 2005a. Disponível em: <<https://git-scm.com/book/en/v2/Getting-Started-A-Short-History-of-Git>>. Acessado em 17 de mai. de 2020.
- COCKBURN, Alistair. **Hexagonal Architecture**. 2005. Disponível em: <<http://archive.is/5j2NI>>. Acessado em 30 de jun. de 2020.
- COHEN, David; LINDVALL, Mikael; COSTA, Patricia. **An Introduction to Agile Methods**. 2004. Disponível em: <http://robertfeldt.net/courses/agile/cohen_2004_intro_to_agile_methods.pdf>. Acessado em 27 de out. de 2019.
- DESK, Jira Service. **Jira Service Desk**. 2020. Disponível em: <<https://www.atlassian.com/br/software/jira/service-desk/free>>. Acessado em 04 de nov. de 2020.
- DEVMEDIA. **Iniciando na Linguagem Java**. 2011. Disponível em: <<https://www.devmedia.com.br/iniciando-na-linguagem-java/21136>>. Acessado em 27 de out. de 2019.
- DOCKER. **Containerizing an application**. 2019b. Disponível em: <<https://docs.docker.com/get-started/part2/>>. Acessado em 24 de nov. de 2019.
- DOCKER. **Get Started, Part 1: Orientation and Setup**. 2019a. Disponível em: <<https://docs.docker.com/get-started/>>. Acessado em 27 de out. de 2019.
- EVANS, Eric. **Domain-Driven Design: Tackling Complexity in the Heart of Software**. Estados Unidos: Addison-Wesley Professional, 2003.
- FACEBOOK. **React**. 2019. Disponível em: <<https://pt-br.reactjs.org/>>. Acessado em 27 de out. de 2019.
- FILHO, Felício Cestari. **ITIL v3**. Rio de Janeiro: Escola Superior de Redes, 2012.
- FILHO, Rubem Melendez. **O Guia Prático do MySQL**. São Paulo: Novatec, 2011.

FOWLER, Martin. **UML Distilled: A Brief Guide to the Standard Object Modeling Language**. Estados Unidos: Addison-Wesley, 2004.

GANTTPROJECT. **GanttProject**. 2019. Disponível em:

<<https://www.ganttproject.biz/>>. Acessado em 12 de nov. de 2019.

GASETA, Edson Roberto. **Fundamentos de Governança de TI**. Rio de Janeiro: Escola Superior de Redes, 2012.

GITHUB. **About**. 2019. Disponível em: <<https://github.com/about>>. Acessado em 17 de mai. de 2020.

GITHUB. **Criar uma pull request - GitHub Docs**. 2020. Disponível em:

<<https://help.github.com/pt/github/collaborating-with-issues-and-pull-requests/creating-a-pull-request>>. Acessado em 03 de jul. de 2020.

GITHUB. **Mastering Issues . GitHub Guides**. 2014. Disponível em:

<<https://guides.github.com/features/issues/>>. Acessado em 03 de jul. de 2020.

GOSLING, James; JOY, Bill; STEELE, Guy; BRACHA, Gilad. Boston: Addison-Wesley, 2005.

GOSLING, James; JOY, Bill; STEELE, Guy; BRACHA, Gilad et al. **The Java Language Specification Java SE 10 Edition**. California: Oracle, 2018.

HEROKU. **Dynos: the heart of the Heroku plataforma**. 2019b. Disponível em:

<<https://www.heroku.com/dynos>>. Acessado em 10 de mai. de 2020.

HEROKU. **What is Heroku?** 2019a. Disponível em:

<<https://www.heroku.com/about>>. Acessado em 10 de mai. de 2020.

IBGE. **Pesquisa Anual de Serviços**. 2015. Disponível em:

<https://biblioteca.ibge.gov.br/visualizacao/periodicos/150/pas_2015_v17_informativo.pdf>. Acessado em 21 de nov. de 2019.

IBM CLOUD EDUCATION. **PaaS (Plataform-as-a-Service)**. 2019. Disponível em:

<<https://www.ibm.com/cloud/learn/paas>>. Acessado em 09 de mai. de 2020.

IEEE. **IEEE Standard Glossary of Software Engineering Terminology**. Estados Unidos: IEEE, 1990.

ISACA. **What is the purpose of COBIT 5 and who is using it?** 2019. Disponível em:

<https://support.isaca.org/app/answers/detail/a_id/712/~/what-is-the-purpose-of-cobit-5-and-who-is-using-it%3F>. Acessado em 25 de jul. de 2020.

LARMAN, Craig. **Utilizando UML e Padrões**. Rio Grande do Sul: Bookman, 2000.

LOVELOCK, Christophe; WRIGHT, Laur. **Serviços: Marketing e Gestão**. São Paulo: Saraiva, 2006.

MANAGER, Desk. **Desk Manager**. 2020. Disponível em: <<https://deskmanager.com.br/>>. Acessado em 04 de nov. de 2020.

MARRONE, Mauricio; KOLBE, Lutz. Impact of IT Service Management Frameworks on the IT Organization. **Business Information Systems Engineering**, v. 3, p. 5–18, 2011. Acessado em 04 de nov. de 2020.

MARTIN, Robert Cecil. **The Principles of OOD**. 2005. Disponível em: <<http://butunclebob.com/ArticleS.UncleBob.PrinciplesOfOod>>. Acessado em 30 de jun. de 2020.

MICROSOFT. **What is TypeScript?** 2020. Disponível em: <<https://www.typescriptlang.org/>>. Acessado em 17 de set. de 2020.

MYSQL. **MySQL Documentation**. 2018. Disponível em: <<https://dev.mysql.com/doc/refman/8.0/en/introduction.html>>. Acessado em 24 de nov. de 2019.

NEVES, Pedro; RUAS, Rui. **O Guia Prático do MySQL**. Portugal: Edições Centro Atlântico, 2001.

NGINX. **Nginx**. 2020. Disponível em: <<https://nginx.org/en/>>. Acessado em 03 de jul. de 2020.

OGC, OFFICE OF GOVERNMENT COMMERCE. **Service Delivery**. Londres: The Stationary Office, 2001.

ORACLE. **Java**. 2019. Disponível em: <https://www.java.com/pt_BR/download/faq/techinfo.xml>. Acessado em 24 de nov. de 2019.

PAULA, Gilles De. **Plano de Ação – O passo a passo da ideia à concretização de seus objetivos!** 2016. Disponível em: <<https://www.treasy.com.br/blog/plano-de-acao/#O-que-e-um-Plano-de-Acao>>. Acessado em 24 de nov. de 2019.

PEREIRA, Ruben; SILVA, Miguel Mira da. A maturity model for implementing ITIL v3. **Proceedings - 2010 6th World Congress on Services, Services-1 2010**, p. 399–406, 2010. Acessado em 04 de nov. de 2020.

PERIARD, Gustavo. **O Ciclo PDCA e a melhoria contínua**. 2011. Disponível em: <<http://www.sobreadministracao.com/o-ciclo-pdca-deming-e-a-melhoria-continua/>>. Acessado em 18 de jan. de 2020.

PETENATE, Marcelo. **WBS: entenda como usar uma Estrutura Analítica de Projeto**. 2019. Disponível em: <<https://www.escolaedti.com.br/wbs-entenda-como-usar-uma-estrutura-analitica-de-projeto>>. Acessado em 24 de nov. de 2019.

PHP. **História do PHP**. 2020a. Disponível em:

<https://www.php.net/manual/pt_BR/history.php.php>. Acessado em 26 de jul. de 2020.

PHP. **PHP**. 2020b. Disponível em: <<https://www.php.net/>>. Acessado em 25 de jun. de 2020.

PHP-FIG. **PSR-4: Autoloader**. 2013. Disponível em:

<<https://www.php-fig.org/psr/psr-4/>>. Acessado em 30 de jun. de 2020.

POSTGRESQL. **A Brief History of PostgreSQL**. 2020a. Disponível em:

<<https://www.postgresql.org/docs/current/history.html>>. Acessado em 10 de mai. de 2020.

POSTGRESQL. **What is PostgreSQL**. 2020b. Disponível em:

<<https://www.postgresql.org/about/>>. Acessado em 10 de mai. de 2020.

PRESSMAN, Roger. **Engenharia de Software**. São Paulo: Makron Books, 1995.

RIBEIRO, Rafael Dias; CUNHA E SOUSA RIBEIRO, Horácio da. **Métodos Ágeis em Gerenciamento de Projetos**. Rio de Janeiro: Horácio da Cunha e Sousa Ribeiro, 2015.

RODRIGUEZ, Gabriel Lopez. **Sinergia**. 2013. Disponível em:

<<https://sourceforge.net/projects/sinergia/>>. Acessado em 04 de nov. de 2020.

SANTOS, Aline Regina; COSTA, Jane Iara Pereira da; SANDER, Josué Anexandre. **As Falhas no Setor de Serviços: um estudo exploratório**. 2008. Disponível em: <https://www.aedb.br/seget/arquivos/artigos08/579_seget08_falhas.pdf>.

Acessado em 21 de jul. de 2020.

SCHWABER, Ken; SUTHERLAND, Jeff. **Guia do Scrum**. 2017. Disponível em:

<<https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-Portuguese-Brazilian.pdf>>. Acessado em 24 de nov. de 2019.

SERVICENOW. **Pricing for ServiceNow IT Service Management**. 2020. Disponível em:

<<https://www.servicenow.com.br/lpgp/pricing-itsm.html>>. Acessado em 04 de nov. de 2020.

SILVA, Alberto Manuel Rodrigues da; VIDEIRA, Carlos Alberto Escaleira. **UML, Metodologias e Ferramentas CASE**. Portugal: Edições Centro Atlântico, 2001.

SYMFONY. **Six good reasons to use Symfony**. 2020c. Disponível em:

<<https://symfony.com/six-good-reasons>>. Acessado em 25 de jun. de 2020.

SYMFONY. **Symfony**. 2020a. Disponível em: <<https://symfony.com/>>. Acessado em 25 de jun. de 2020.

SYMFONY. **Symfony Community**. 2020b. Disponível em:

<<https://symfony.com/community>>. Acessado em 26 de jul. de 2020.

TAKEUCHI, Hirotaka; NONAKA, Ikujiro. **The New New Product Development Game**. Estados Unidos: Harvard Business Review, 1986.

TECNICON. **Metodologia Scrum para a gestão de processos ágeis na indústria**. 2019.

Disponível em: <https://www.tecnicon.com.br/blog/411-Metodologia_Scrum_para_a_gestao_de_processos_ageis_na_industria>. Acessado em 24 de nov. de 2019.

TRELLO. **Trello**. 2018. Disponível em: <<https://trello.com/>>. Acessado em 12 de nov. de 2019.

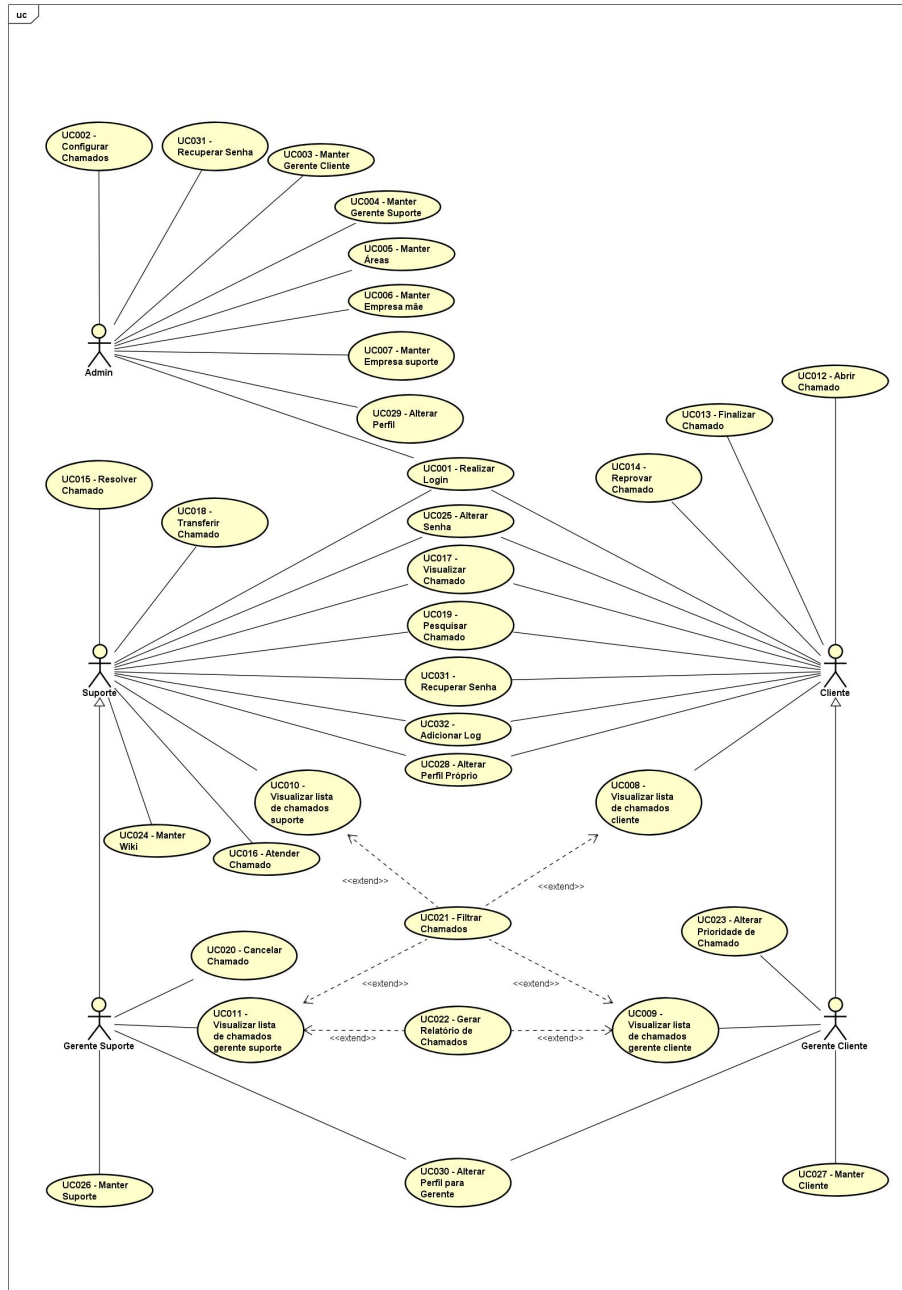
W3TECHS. **Usage Statistics and Market Share of PHP for Websites**. 2020. Disponível em: <<https://w3techs.com/technologies/details/pl-php>>. Acessado em 25 de jun. de 2020.

W3TECHS. **Usage statistics of server-side programming languages for websites**. 2020.

Disponível em: <https://w3techs.com/technologies/overview/programming_language>. Acessado em 21 de jul. de 2020.

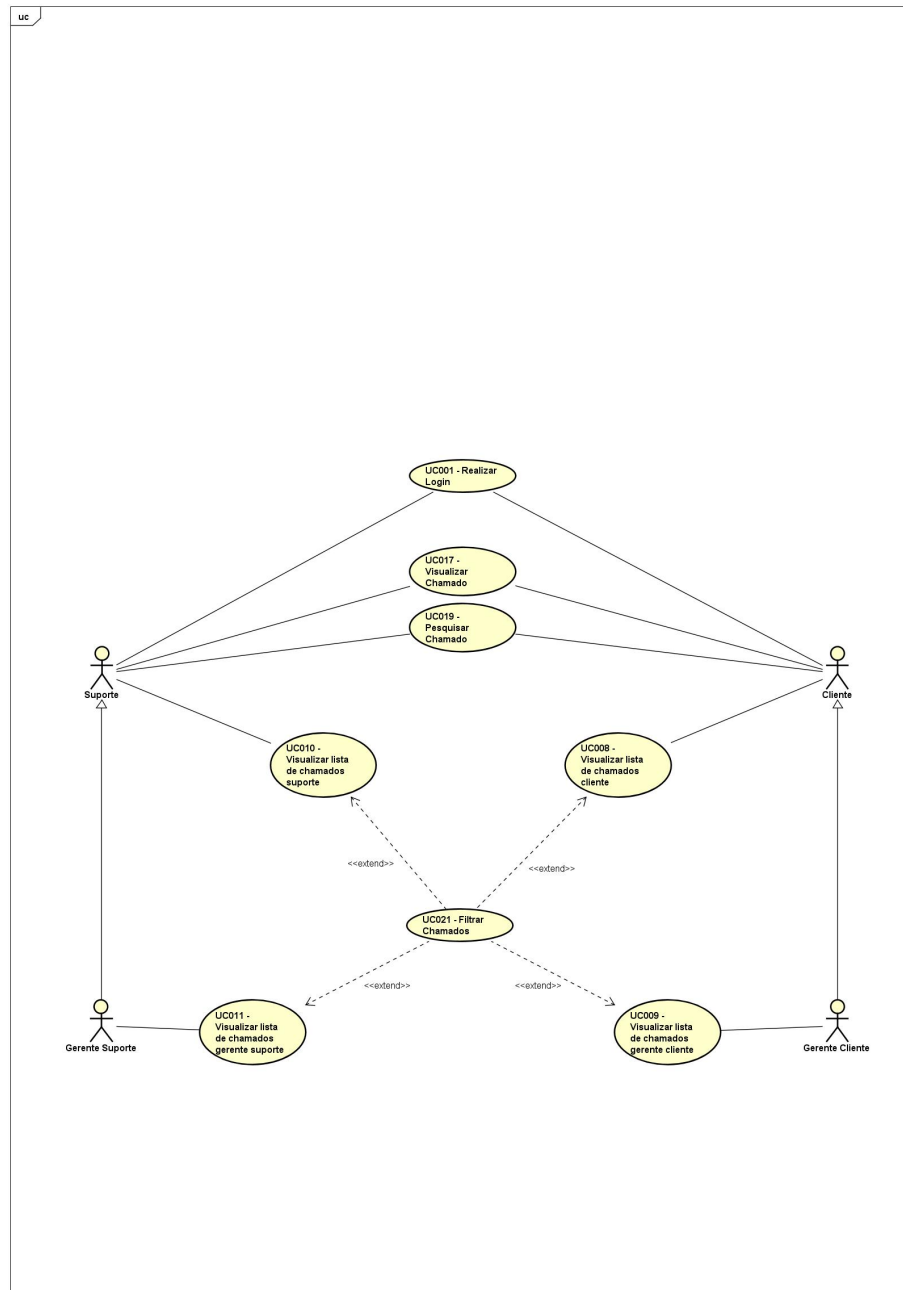
APÊNDICE A – DIAGRAMA DE CASOS DE USO

Figura A.1: DIAGRAMA DE CASOS DE USO



Fonte: Os autores (2019)

Figura A.2: DIAGRAMA DE CASOS DE USO MOBILE

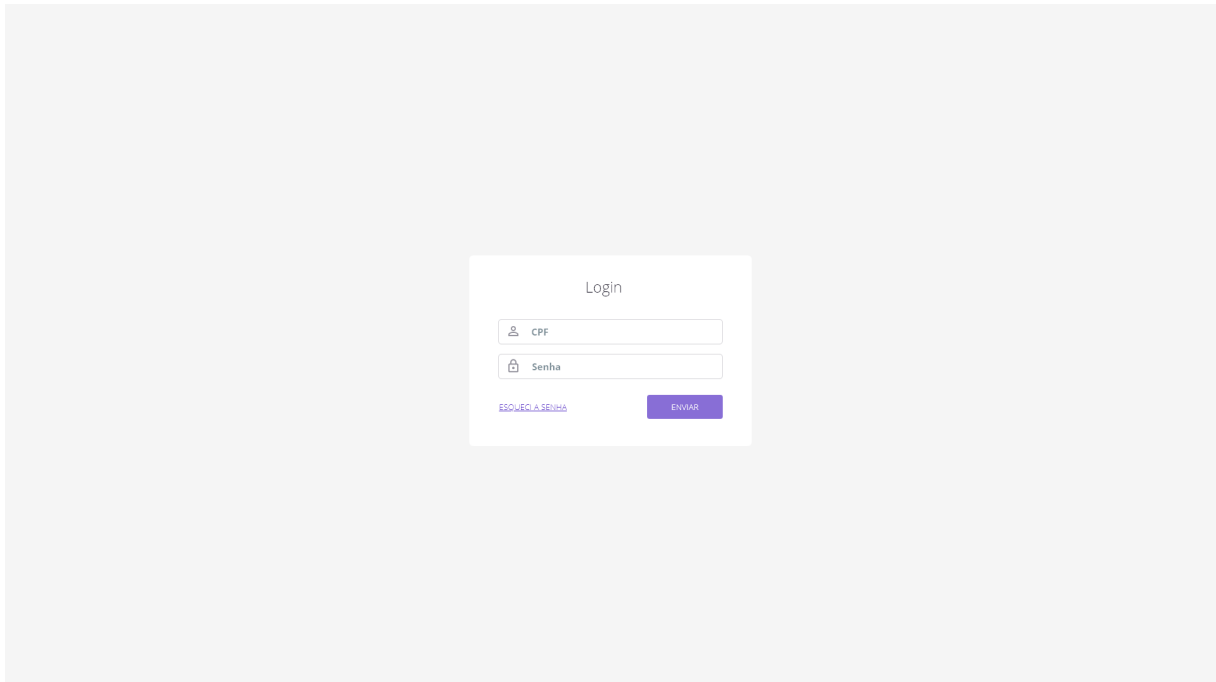


Fonte: Os autores (2020)

APÊNDICE B – ESPECIFICAÇÕES DE CASOS DE USO

UC001 - Realizar Login

Figura B.1: TELA DE LOGIN



FONTE: Os autores (2019)

Descrição: O ator pode logar no sistema, informando os dados de login (CPF e senha) para ter acesso à tela inicial.

Atores: Cliente, Gerente Cliente, Suporte, Gerente Suporte e Admin.

Pré-Condição: O ator deve estar pré-cadastrado.

Pós-Condição: O ator tem acesso à tela inicial e ao sistema.

Fluxo de Eventos Principal:

- P1.** O ator tem acesso à tela de login;
- P2.** O ator preenche o campo do CPF e senha;
- P3.** O ator clica no botão "Login";
- P4.** O sistema verifica se o login é válido (E1, E2);
- P5.** O sistema redireciona para a tela principal;
- P6.** O caso de uso é encerrado;

Fluxos Alternativos: Não se aplica.

Fluxos de Exceção:

E1: O ator não está cadastrado:

1. O sistema retorna uma mensagem de erro informando que o CPF digitado não está cadastrado;
2. O caso de uso é encerrado;

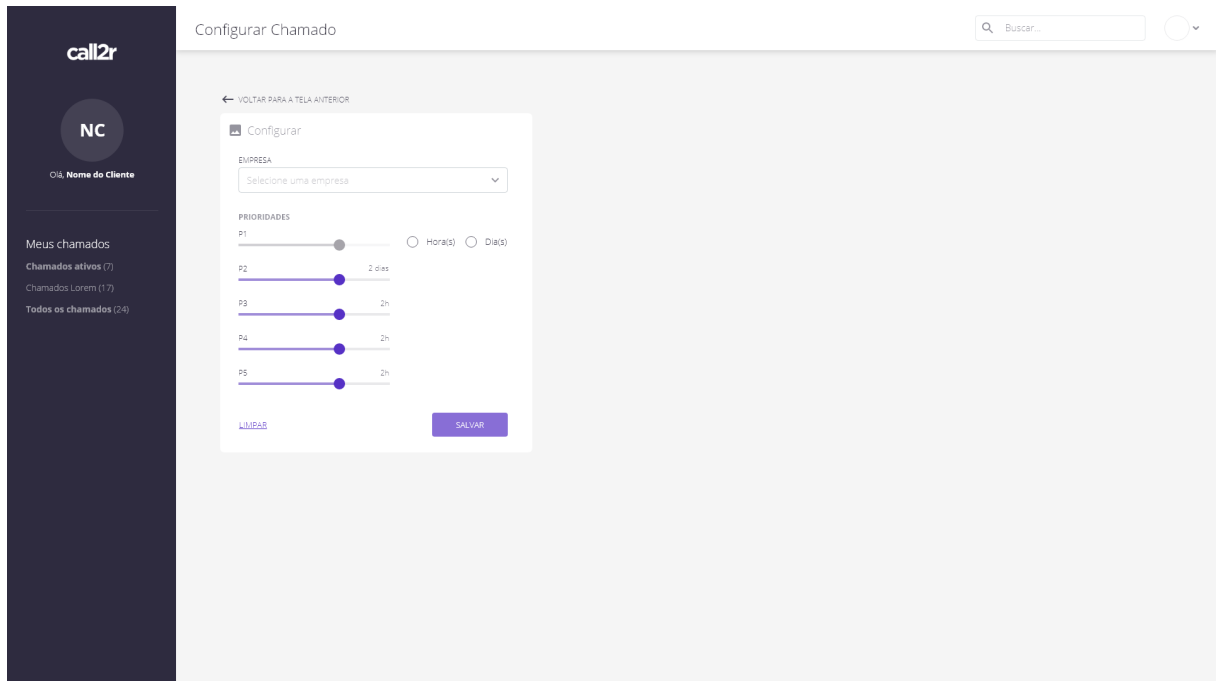
E2: Senha incorreta:

1. O sistema retorna uma mensagem de erro informando que a senha está incorreta;
2. O caso de uso é encerrado;

Regras de Negócio: O sistema libera um acesso diferente para cada tipo de ator: Admin, Gerente Cliente, Cliente, Gerente Suporte ou Suporte.

UC002 - Configurar Chamados

Figura B.2: TELA DE CONFIGURAÇÃO DE CHAMADOS



FONTE: Os autores (2019)

Descrição: O ator pode, através da edição do cadastro da empresa, criar, alterar, excluir e visualizar configurações de chamados para cada empresa.

Atores: Admin.

Pré-Condição: O ator deve estar logado como Admin e devem existir empresas suporte válidas cadastradas.

Pós-Condição: O ator poderá ter criado, visualizado, alterado ou excluído uma configuração de chamado.

Fluxo de Eventos Principal:

- P1.** O ator acessa a opção de visualização de configurações;
- P2.** O sistema apresenta ao ator uma lista com as configurações de chamados;
- P3.** O ator pode visualizar as configurações (A1, A2, A3, A4);
- P4.** O caso de uso é encerrado;

Fluxos Alternativos:

A1: Criar configuração de chamado:

1. O usuário clica em novo para criar uma nova configuração;

2. O sistema apresenta uma tela com campos para serem preenchidos;
3. O ator preenche os campos;
4. O ator seleciona a empresa para qual a regra proposta será estabelecida;
5. O ator clica em Salvar para salvar a configuração (E1, E2);
6. O sistema salva as alterações;
7. O caso de uso é encerrado;

A2: Visualizar configuração de chamado:

1. O ator clica em uma configuração de chamado existente;
2. O sistema apresenta uma tela com os dados da configuração;
3. O caso de uso é encerrado;

A3: Alterar configuração de chamado:

1. O ator clica em uma configuração de chamado existente;
2. O sistema apresenta uma tela com os dados da configuração;
3. O ator realiza as alterações;
4. O ator clica em Salvar (E1, E2);
5. O sistema salva os dados;
6. O caso de uso é encerrado;

A4: Excluir configuração de chamado:

1. O ator clica em uma configuração de chamado existente;
2. O sistema apresenta uma tela com os dados da configuração;
3. O ator clica em excluir para excluir a configuração de chamado;
4. O sistema abre uma tela de confirmação;
5. O ator confirma a exclusão (E3);
6. O sistema exclui a configuração;
7. O caso de uso é encerrado;

Fluxos de Exceção:

E1: Validação de dados de preenchimento:

1. O sistema verifica se os dados já existem;
2. O sistema alerta ao usuário que os dados já existem;

3. O caso de uso é encerrado;

E2: Configuração ou campos vazios:

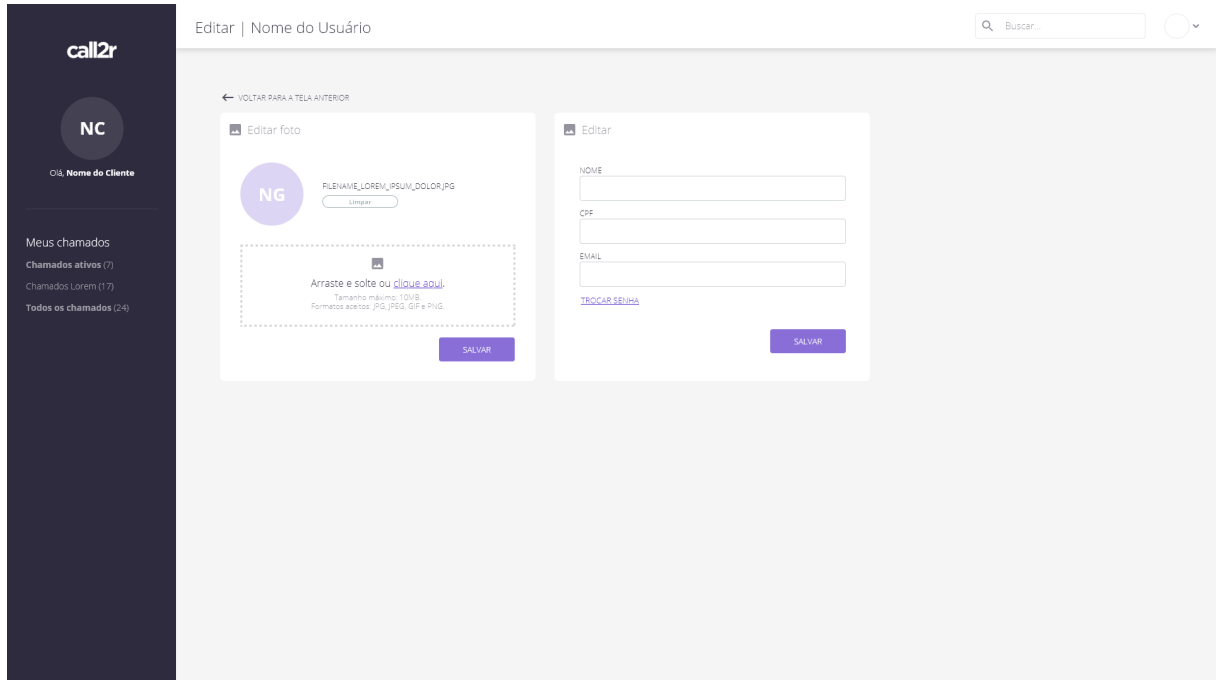
1. O sistema retorna uma mensagem de erro informando que existem campos vazios;
2. O caso de uso é encerrado;

E3: Validação para verificar a vinculação:

1. O sistema verifica se existe alguma empresa suporte utilizando a configuração;
2. O sistema informa ao usuário que é necessário desvincular a empresa vinculada;
3. O caso de uso é encerrado;

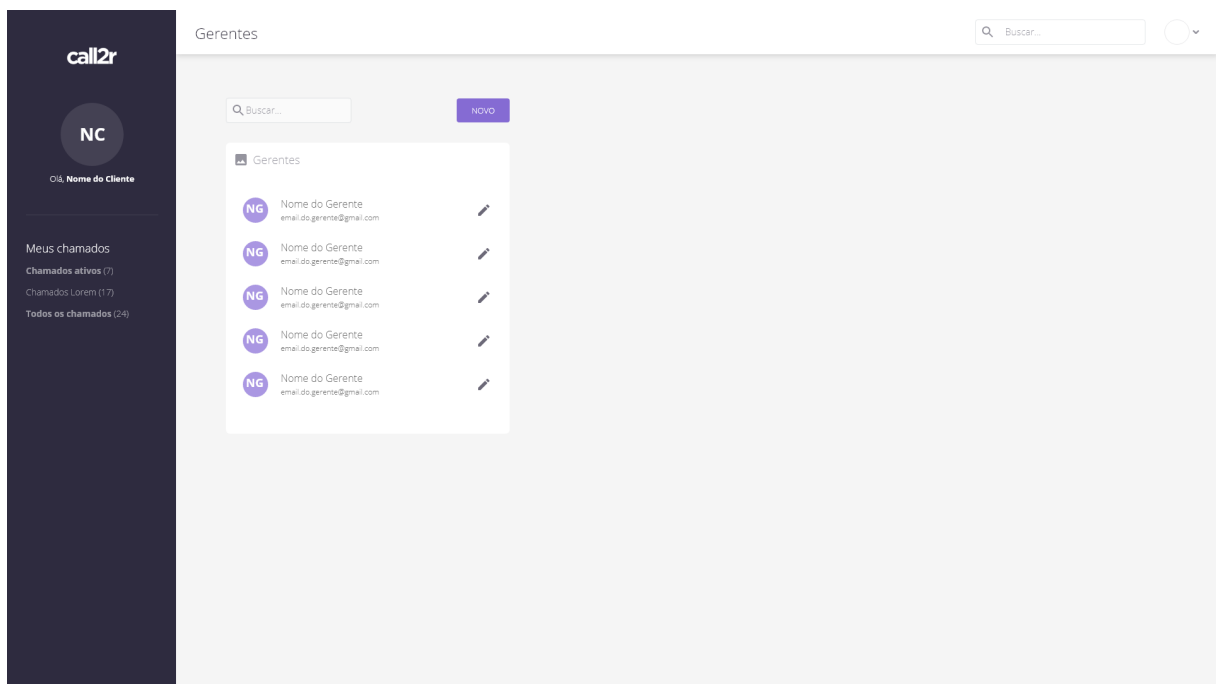
UC003 - Manter Gerente Cliente

Figura B.3: TELA DE EDIÇÃO DE GERENTE



FONTE: Os autores (2019)

Figura B.4: TELA DE LISTA DE GERENTES



FONTE: OS AUTORES (2019)

Descrição: O ator pode visualizar a lista de gerentes de cliente e pode criar, alterar ou excluir gerentes de cliente.

Atores: Admin.

Pré-Condição: O ator deve estar logado como Admin.

Pós-Condição: O ator poderá ter criado, visualizado, alterado, excluído ou pesquisado gerentes cliente.

Fluxo de Eventos Principal:

- P1.** O ator seleciona a opção lista de gerente cliente;
- P2.** O sistema carrega uma lista de gerentes cliente;
- P3.** O ator pode visualizar os gerentes (A1, A2, A3, A4, A5);
- P4.** O caso de uso é encerrado;

Fluxos Alternativos:

A1: Criar gerente cliente:

- 1. O ator clica em Novo;
- 2. O sistema apresenta uma tela de cadastro de gerente cliente;
- 3. O ator preenche o cadastro com as informações do gerente cliente;
- 4. O ator clica em Salvar (E1, E2);
- 5. O sistema cadastra o gerente cliente;
- 6. O caso de uso é encerrado;

A2: Visualizar gerente cliente:

- 1. O ator clica em um gerente cliente existente;
- 2. O sistema apresenta uma tela com os dados do gerente cliente;
- 3. O caso de uso é encerrado;

A3: Alterar gerente cliente:

- 1. O ator clica em Editar em um gerente cliente existente;
- 2. O sistema apresenta uma tela de alteração de gerente cliente;
- 3. O ator pode alterar os dados do gerente cliente;
- 4. O ator clica em Salvar (E1, E2);
- 5. O sistema abre uma tela de confirmação;
- 6. O ator confirma;

7. O caso de uso é encerrado;

A4: Desativar gerente cliente:

1. O ator clica em Desativar em um gerente cliente existente;
2. O sistema abre uma tela de confirmação;
3. O ator confirma a desativação;
4. O sistema desativa o gerente cliente selecionado;
5. O caso de uso é encerrado;

A5: Ativar gerente cliente:

1. O ator visualiza um gerente cliente;
2. O ator clica em Ativar;
3. O sistema ativa o gerente cliente;
4. O caso de uso é encerrado;

A6: Pesquisa gerente cliente:

1. O ator preenche um campo de pesquisa;
2. O sistema pesquisa os gerentes cliente correspondentes;
3. O sistema apresenta os gerentes cliente;
4. O caso de uso é encerrado;

Fluxos de Exceção:

E1: Gerente cliente já existe:

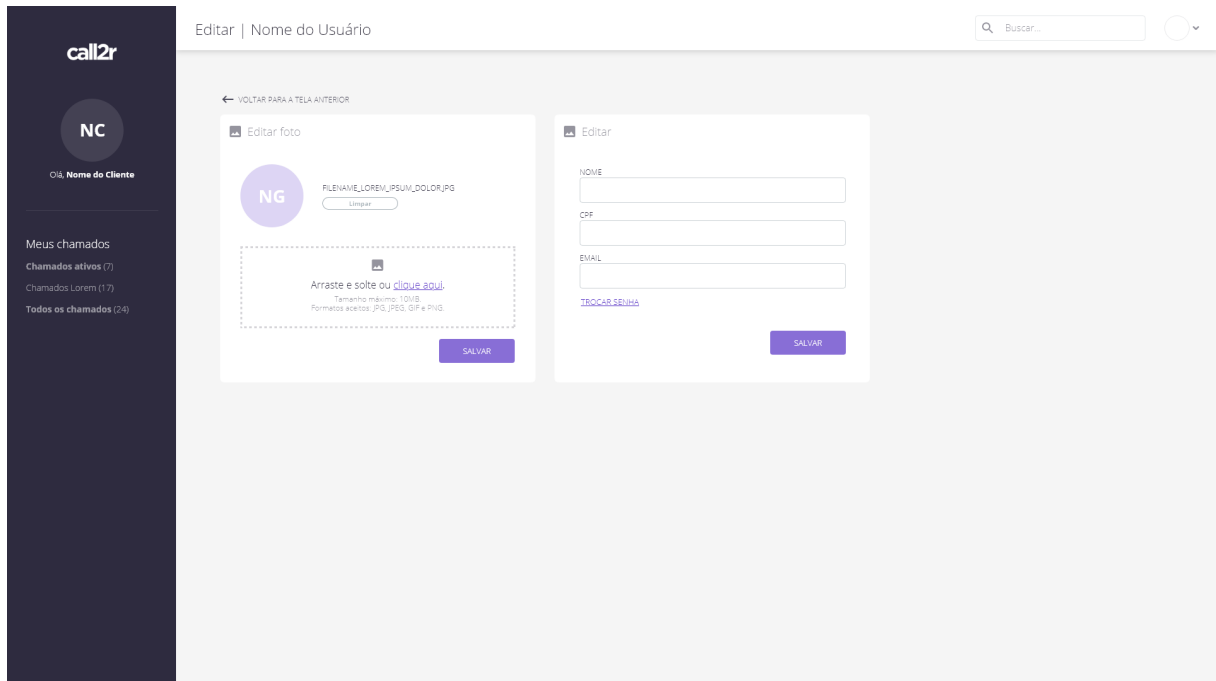
1. O sistema retorna uma mensagem de erro informando que o gerente cliente já existe;
2. O caso de uso é encerrado;

E2: Gerente cliente ou campos vazios:

1. O sistema retorna uma mensagem de erro informando que existem campos vazios;
2. O caso de uso é encerrado;

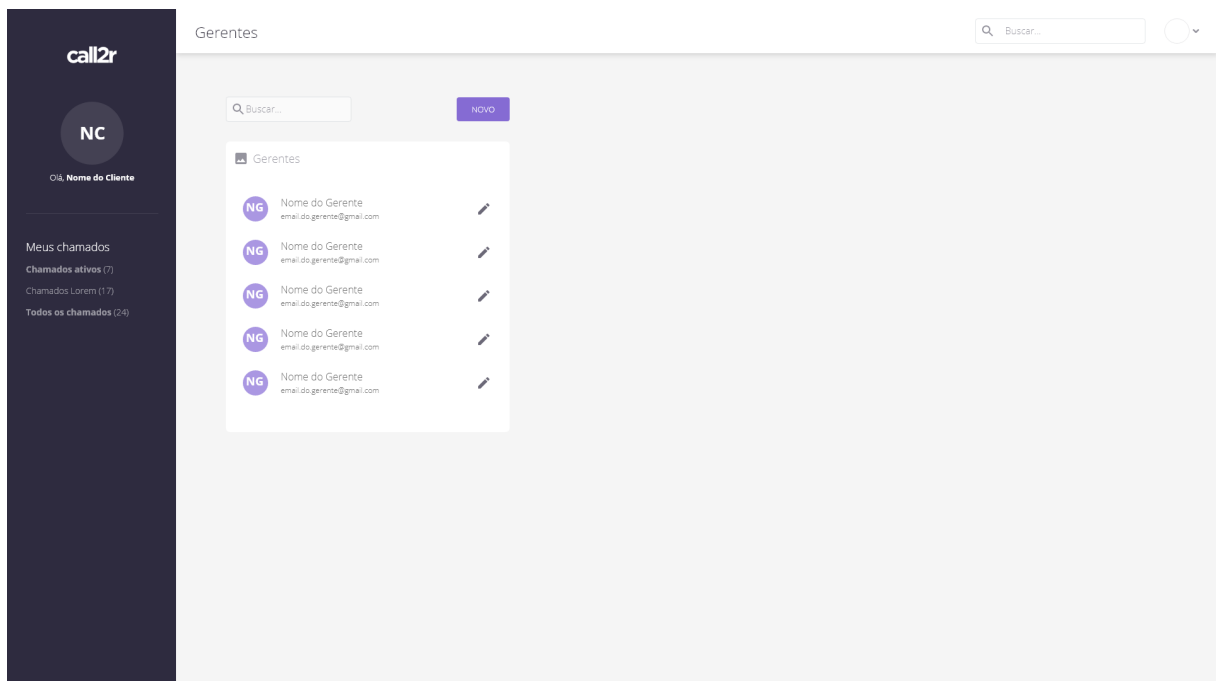
UC004 - Manter Gerente Suporte

Figura B.5: TELA DE EDIÇÃO DE GERENTE



FONTE: Os autores (2019)

Figura B.6: TELA DE LISTA DE GERENTES



FONTE: OS AUTORES (2019)

Descrição: O ator pode visualizar a lista de gerentes de suporte e pode criar, alterar ou excluir gerentes de suporte.

Atores: Admin.

Pré-Condição: O ator deve estar logado como Admin.

Pós-Condição: O ator poderá ter criado, visualizado, alterado, excluído ou pesquisado gerentes suporte.

Fluxo de Eventos Principal:

- P1.** O ator seleciona a opção lista de gerente suporte;
- P2.** O sistema carrega uma lista de gerentes suporte;
- P3.** O ator pode visualizar os gerentes (A1, A2, A3, A4);
- P4.** O caso de uso é encerrado;

Fluxos Alternativos:

A1: Criar gerente suporte:

- 1. O ator clica em Novo;
- 2. O sistema apresenta uma tela de cadastro de gerente suporte;
- 3. O ator preenche o cadastro com as informações do gerente suporte;
- 4. O ator clica em Salvar (E1, E2);
- 5. O sistema cadastra o gerente suporte;
- 6. O caso de uso é encerrado;

A2: Visualizar gerente suporte:

- 1. O ator clica em um gerente suporte existente;
- 2. O sistema apresenta uma tela com os dados do gerente suporte;
- 3. O caso de uso é encerrado;

A3: Alterar gerente suporte:

- 1. O ator clica em Editar em um gerente suporte existente;
- 2. O sistema apresenta uma tela de alteração de gerente suporte;
- 3. O ator pode alterar os dados do gerente suporte;
- 4. O ator clica em Salvar (E1, E2);
- 5. O sistema abre uma tela de confirmação;
- 6. O ator confirma;

7. O caso de uso é encerrado;

A4: Desativar gerente suporte:

1. O ator clica em Desativar em um gerente suporte existente;
2. O sistema abre uma tela de confirmação;
3. O ator confirma a desativação;
4. O sistema desativa o gerente suporte selecionado;
5. O caso de uso é encerrado;

A5: Ativar gerente suporte:

1. O ator visualiza um gerente suporte;
2. O ator clica em Ativar;
3. O sistema ativa o gerente suporte;
4. O caso de uso é encerrado;

A6: Pesquisa gerente suporte:

1. O ator preenche o campo de pesquisa;
2. O sistema pesquisa os gerentes suporte correspondentes;
3. O sistema apresenta os gerentes suporte;
4. O caso de uso é encerrado;

Fluxos de Exceção:

E1: Gerente suporte já existe:

1. O sistema retorna uma mensagem de erro informando que o gerente suporte já existe;
2. O caso de uso é encerrado;

E2: Gerente suporte ou campos vazios:

1. O sistema retorna uma mensagem de erro informando que existem campos vazios;
2. O caso de uso é encerrado;

UC005 - Manter áreas

Figura B.7: TELA DE EDIÇÃO DE ÁREAS

The screenshot shows a web interface for managing areas. On the left is a dark sidebar with the 'call2r' logo, a user profile 'NC', and a list of call statistics. The main content area is titled 'Dados da Empresa Mãe' and contains two forms. The first form, 'Editar empresa mãe', has fields for 'NOME DA EMPRESA', 'CNPJ', 'RAZÃO SOCIAL', 'NOME FICTÍCIO', 'ENDEREÇO', and 'DESCRIÇÃO', with a 'SALVAR' button at the bottom. The second form, 'Lista de áreas', has a search field 'Nome da área', a list of three 'Nome da área' entries, a 'Digite o nome da área...' input, and buttons for 'NOVA', 'SALVAR', and 'CANCELAR'.

FONTE: Os autores (2019)

Descrição: O ator pode criar, alterar ou excluir áreas.

Atores: Admin.

Pré-Condição: O ator deve estar logado como Admin.

Pós-Condição: O ator poderá ter criado, alterado ou excluído áreas.

Fluxo de Eventos Principal:

- P1.** O ator abre a tela de áreas;
- P2.** O sistema apresenta uma lista de áreas;
- P3.** O ator pode visualizar as áreas (A1, A2, A3);
- P4.** O caso de uso é encerrado;

Fluxos Alternativos:

A1: Criar área:

- 1. O ator clica em Novo;
- 2. O sistema apresenta uma tela de cadastro;
- 3. O ator preenche os campos da área;
- 4. O ator clica em Salvar (E1, E2);

5. O sistema cria a área;
6. O caso de uso é encerrado;

A2: Alterar área:

1. O ator clica em Editar na área existente;
2. O apresenta a tela com os dados da área;
3. O ator altera os dados;
4. O ator clica em Salvar (E1, E2);
5. O sistema apresenta uma tela de confirmação;
6. O ator confirma as alterações;
7. O sistema salva as alterações;
8. O caso de uso é encerrado;

A3: Excluir área:

1. O ator clica em Excluir na área existente;
2. O sistema apresenta uma tela de confirmação;
3. O ator confirma a exclusão (E3);
4. O sistema exclui a área;
5. O caso de uso é encerrado;

Fluxos de Exceção:**E1:** Nome da área já existe:

1. O sistema retorna uma mensagem de erro informando que a área já existe;
2. O caso de uso é encerrado;

E2: Área ou campos vazios:

1. O sistema retorna uma mensagem de erro informando que existem campos vazios;
2. O caso de uso é encerrado;

E3: Área com dependência:

1. O sistema retorna uma mensagem de erro informando que existem dependências;
2. O caso de uso é encerrado;

Regras de Negócio: As áreas se relacionam com as empresas e o sistema não permite que dados que dependam da área sejam apagados. As empresas podem ter mais de uma área.

UC006 - Manter Empresa Mãe

Figura B.8: TELA DE EDIÇÃO DA EMPRESA MÃE

FONTE: Os autores (2019)

Descrição: O ator pode cadastrar, visualizar ou alterar detalhes da empresa mãe.

Atores: Admin.

Pré-Condição: O ator deve estar logado como Admin.

Pós-Condição: O ator poderá ter cadastrado, visualizado ou alterado dados da empresa mãe.

Fluxo de Eventos Principal:

- P1.** O ator abre a tela de empresa mãe;
- P2.** O sistema apresenta uma tela com as informações já cadastradas (A1, A2);
- P3.** O caso de uso é encerrado;

Fluxos Alternativos:

A1: Cadastro inicial da empresa mãe:

1. O sistema apresenta uma tela de cadastro de empresa mãe;
2. O ator preenche as informações de cadastro da empresa mãe;
3. O ator clica em Salvar (E1);
4. O sistema salva o cadastro;
5. O caso de uso é encerrado;

A2: Alterar empresa mãe:

1. O sistema apresenta uma tela com os dados da empresa mãe;
2. O ator altera os dados;
3. O ator clica em Salvar (E1);
4. O sistema apresenta uma tela de confirmação;
5. O ator confirma;
6. O sistema salva as alterações;
7. O caso de uso é encerrado;

Fluxos de Exceção:**E1:** Empresa mãe ou campos vazios:

1. O sistema retorna uma mensagem de erro informando que existem campos vazios;
2. O caso de uso é encerrado;

UC007 - Manter Empresa Suporte

Figura B.9: TELA DE EDIÇÃO DA EMPRESA SUPORTE

call2r

Editar | Nome da Empresa

Buscar...

VOLTAR PARA A TELA ANTERIOR

Editar empresa

NOME DA EMPRESA

CNPJ

DESCRIÇÃO

ÁREA

Selecionar uma opção

SALVAR

Lista de subáreas

Nome da subárea

Nome da subárea

Nome da subárea

Nome da subárea

Nome da subárea

Digite o nome da subárea...

CANCELAR

NOVA

SALVAR

Meus chamados

Chamados ativos (7)

Chamados Lorem (17)

Todos os chamados (24)

Olá, Nome do Cliente

FONTE: OS AUTORES (2019)

Figura B.10: TELA DE LISTA DE EMPRESAS SUPORTE

call2r

Empresas Suporte

Buscar...

Empresas

ES Nome da Empresa de Suporte
Descrição do que a empresa de suporte faz

ES Nome da Empresa de Suporte
Descrição do que a empresa de suporte faz

ES Nome da Empresa de Suporte
Descrição do que a empresa de suporte faz

ES Nome da Empresa de Suporte
Descrição do que a empresa de suporte faz

ES Nome da Empresa de Suporte
Descrição do que a empresa de suporte faz

Meus chamados

Chamados ativos (7)

Chamados Lorem (17)

Todos os chamados (24)

Olá, Nome do Cliente

Fonte: Os autores (2019)

Descrição: O ator pode visualizar, criar, alterar ou excluir uma empresa suporte.

Atores: Admin.

Pré-Condição: O ator deve estar logado como Admin e é necessário uma empresa mãe cadastrada.

Pós-Condição: O ator poderá ter criado, visualizado, alterado ou excluído empresas suporte.

Fluxo de Eventos Principal:

- P1.** O ator abre a tela de empresas suporte;
- P2.** O sistema apresenta uma lista de empresas suporte;
- P3.** O ator pode visualizar as empresas suporte (A1, A2, A3, A4);
- P4.** O caso de uso é encerrado;

Fluxos Alternativos:

A1: Criar empresa suporte:

- 1. O ator clica em Novo;
- 2. O sistema apresenta uma tela de cadastro de empresa suporte;
- 3. O ator preenche as informações da empresa suporte;
- 4. O ator clica em Salvar (E1, E2);
- 5. O sistema salva as informações da empresa suporte;
- 6. O caso de uso é encerrado;

A2: Visualizar empresa suporte:

- 1. O ator clica em uma empresa suporte existente;
- 2. O sistema apresenta uma tela com os dados da empresa suporte selecionada;
- 3. O caso de uso é encerrado;

A3: Alterar empresa suporte:

- 1. O ator clica em Editar em uma empresa suporte existente;
- 2. O sistema apresenta uma tela com os dados da empresa suporte selecionada;
- 3. O ator altera os dados da empresa;
- 4. O ator clica em Salvar (E1, E2);
- 5. O sistema apresenta uma tela de confirmação;
- 6. O ator confirma;
- 7. O sistema salva as alterações na empresa;

8. O caso de uso é encerrado;

A4: Excluir empresa suporte:

1. O ator clica em Excluir em uma empresa suporte existente;
2. O sistema apresenta uma tela de confirmação;
3. O ator confirma (E3);
4. O sistema exclui a empresa;
5. O caso de uso é encerrado;

Fluxos de Exceção:

E1: Empresa suporte já existe:

1. O sistema retorna uma mensagem de erro informando que a empresa suporte já existe;
2. O caso de uso é encerrado;

E2: Empresa suporte ou campos vazios:

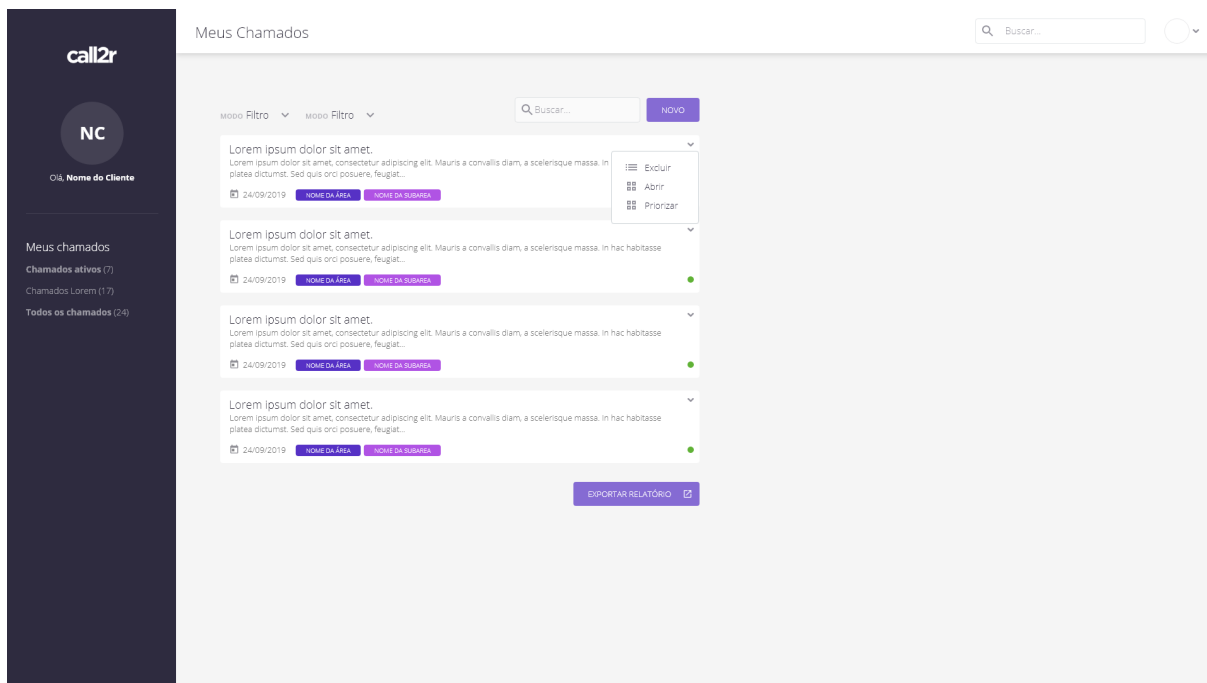
1. O sistema retorna uma mensagem de erro informando que existem campos vazios;
2. O caso de uso é encerrado;

E3: Itens vinculados a empresa suporte:

1. O sistema retorna uma mensagem de erro informando que existem itens vinculados a empresa suporte e que ela não pode ser excluída;
2. O caso de uso é encerrado;

UC008 - Visualizar Lista de Chamados Cliente

Figura B.11: TELA DE LISTA DE CHAMADOS



FONTE: OS AUTORES (2019)

Descrição: O ator pode visualizar a lista de chamados.

Atores: Cliente.

Pré-Condição: O ator deve estar logado como cliente.

Pós-Condição: O ator terá acesso a lista de chamados.

Fluxo de Eventos Principal:

- P1.** O ator acessa a tela inicial;
- P2.** O sistema apresenta uma lista de chamados do usuário;
- P3.** O ator pode visualizar a lista de chamados (A1, A2);
- P4.** O caso de uso é encerrado;

Fluxos Alternativos:

A1: Chamados ativos:

1. O sistema apresenta uma lista de chamados de seu usuário com estado "Aguardando suporte", "Em atendimento" e "Aguardando cliente";
2. O ator pode visualizar a lista de chamados;
3. O caso de uso é encerrado;

A2: Todos os chamados:

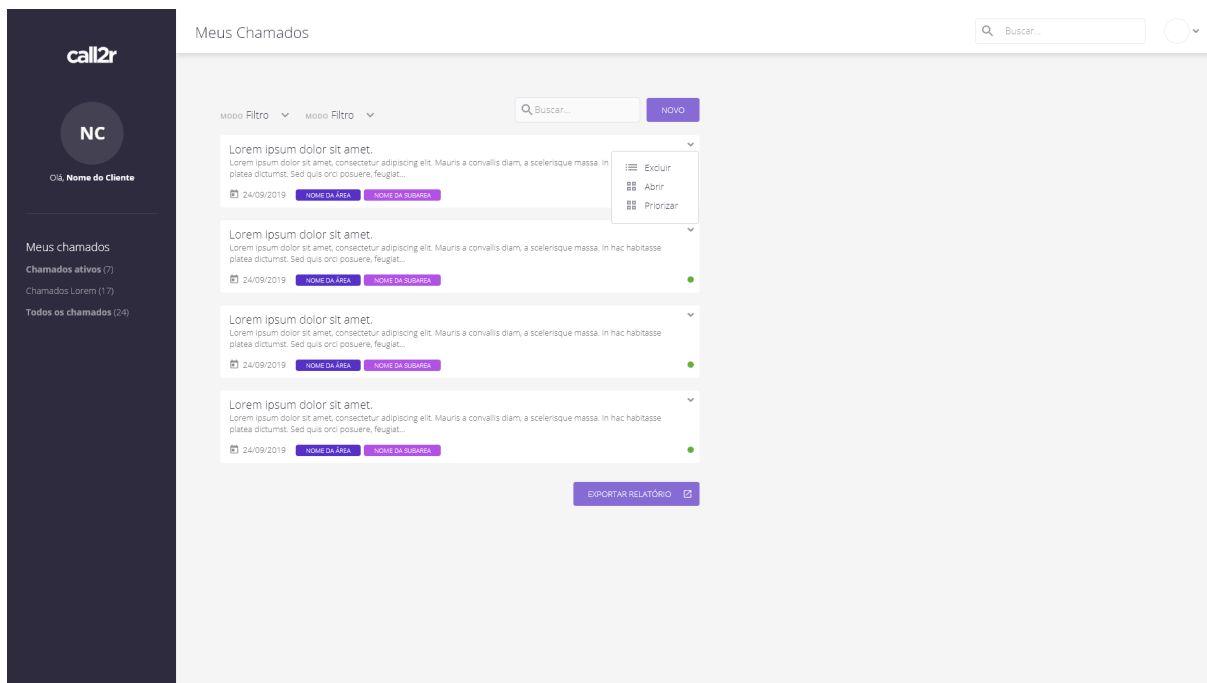
1. O sistema apresenta uma lista com chamados do seu usuário;
2. O ator pode visualizar a lista de chamados;
3. O caso de uso é encerrado;

Fluxos de Exceção: Não se aplica.

Regras de Negócio: O usuário do tipo Cliente só tem acesso a chamados criados por ele.

UC009 - Visualizar Lista de Chamados Gerente Cliente

Figura B.12: TELA DE LISTA DE CHAMADOS



FONTE: OS AUTORES (2019)

Descrição: O ator pode visualizar a lista de chamados.

Atores: Gerente Cliente.

Pré-Condição: O ator deve estar logado como gerente cliente.

Pós-Condição: O ator terá acesso a lista de chamados.

Fluxo de Eventos Principal:

- P1.** O ator acessa a tela inicial;
- P2.** O sistema apresenta uma lista de chamados do usuário;
- P3.** O ator pode visualizar a lista de chamados (A1, A2, A3);
- P4.** O caso de uso é encerrado;

Fluxos Alternativos:

A1: Chamados ativos:

1. O sistema apresenta uma lista de chamados dos usuários Cliente e Gerente cliente com estado "Aguardando suporte", "Em atendimento" e "Aguardando cliente";
2. O ator pode visualizar a lista de chamados;
3. O caso de uso é encerrado;

A2: Chamados gerente:

1. O sistema apresenta uma lista de chamados do ator;
2. O ator pode visualizar a lista de chamados;
3. O caso de uso é encerrado;

A3: Todos os chamados:

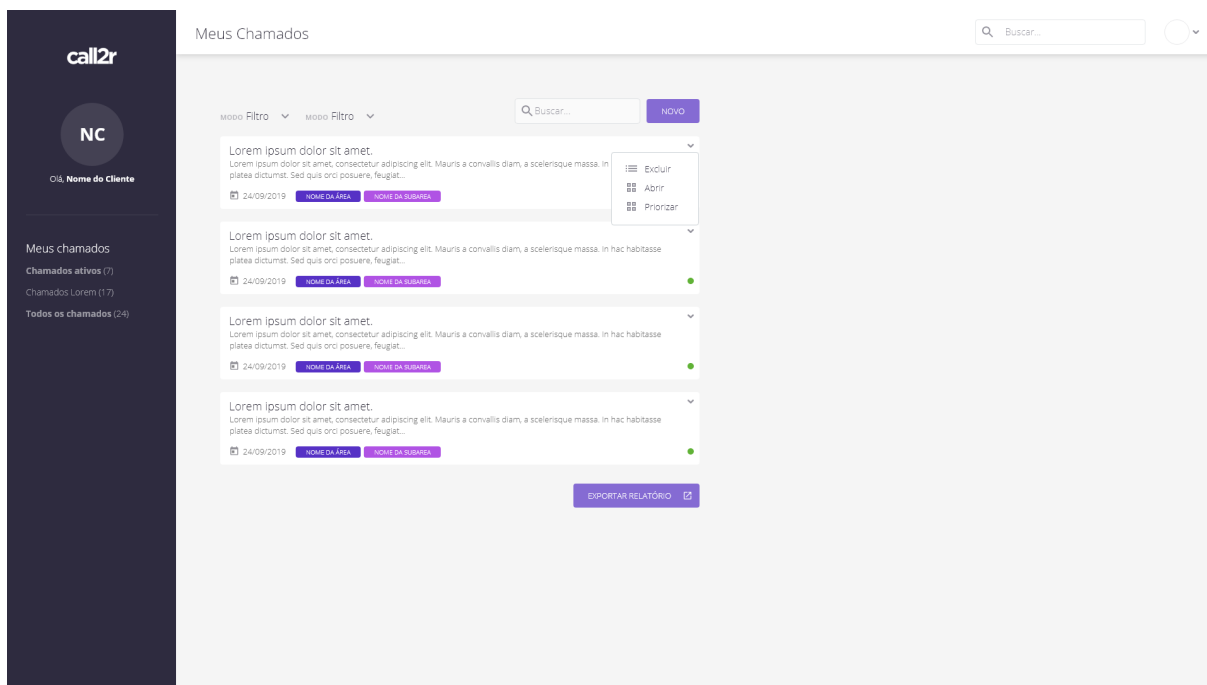
1. O sistema apresenta uma lista com chamados dos usuários Cliente e Gerente cliente;
2. O ator pode visualizar a lista de chamados;
3. O caso de uso é encerrado;

Fluxos de Exceção: Não se aplica.

Regras de Negócio: O usuário do tipo Gerente Cliente tem acesso a todos os chamados criados pelos usuários do tipo Cliente e Gerente Cliente.

UC010 - Visualizar Lista de Chamados Suporte

Figura B.13: TELA DE LISTA DE CHAMADOS



FONTE: OS AUTORES (2019)

Descrição: O ator pode visualizar a lista de chamados.

Atores: Suporte.

Pré-Condição: O ator deve estar logado como cliente.

Pós-Condição: O ator terá acesso a lista de chamados.

Fluxo de Eventos Principal:

- P1.** O ator acessa a tela inicial;
- P2.** O sistema apresenta uma lista de chamados do usuário;
- P3.** O ator pode visualizar a lista de chamados (A1, A2, A3);
- P4.** O caso de uso é encerrado;

Fluxos Alternativos:

A1: Chamados ativos:

1. O sistema apresenta uma lista de chamados de seu usuário com estado "Em atendimento" e "Aguardando cliente";
2. O ator pode visualizar a lista de chamados;
3. O caso de uso é encerrado;

A2: Chamados pendentes:

1. O sistema apresenta uma lista de chamados com estado "Aguardando suporte";
2. O ator pode visualizar a lista de chamados;
3. O caso de uso é encerrado;

A3: Todos os chamados:

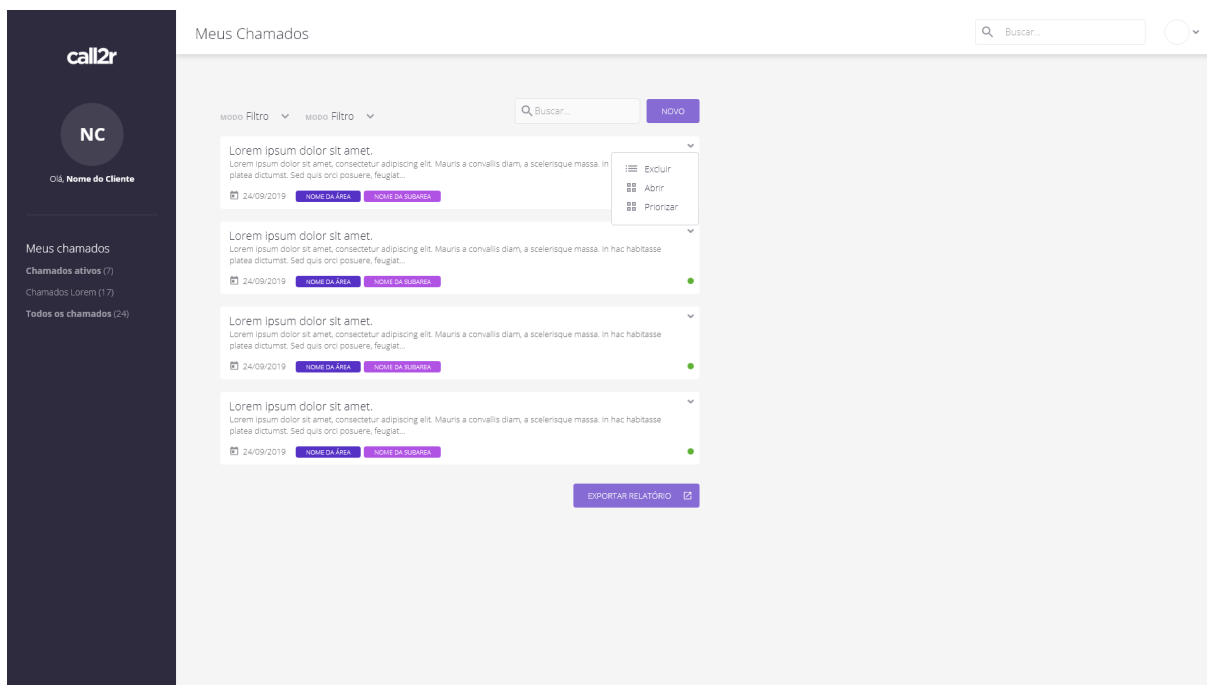
1. O sistema apresenta uma lista com chamados do seu usuário;
2. O ator pode visualizar a lista de chamados;
3. O caso de uso é encerrado;

Fluxos de Exceção: Não se aplica.

Regras de Negócio: O usuário do tipo Suporte só tem acesso a chamados no estado "Aguardando suporte" e chamados no estado "Em atendimento" pelo seu usuário.

UC011 - Visualizar Lista de Chamados Gerente Suporte

Figura B.14: TELA DE LISTA DE CHAMADOS



FONTE: Os autores (2019)

Descrição: O ator pode visualizar a lista de chamados.

Atores: Gerente Suporte.

Pré-Condição: O ator deve estar logado como gerente cliente.

Pós-Condição: O ator terá acesso a lista de chamados.

Fluxo de Eventos Principal:

- P1.** O ator acessa a tela inicial;
- P2.** O sistema apresenta uma lista de chamados do usuário;
- P3.** O ator pode visualizar a lista de chamados (A1, A2, A3, A4);
- P4.** O caso de uso é encerrado;

Fluxos Alternativos:

A1: Chamados ativos:

1. O sistema apresenta uma lista de chamados dos usuários Suporte com estado "Em atendimento" e "Aguardando cliente";
2. O ator pode visualizar a lista de chamados;
3. O caso de uso é encerrado;

A2: Chamados pendentes:

1. O sistema apresenta uma lista de chamados com estado "Aguardando suporte";
2. O ator pode visualizar a lista de chamados;
3. O caso de uso é encerrado;

A3: Chamados gerente:

1. O sistema apresenta uma lista de chamados do ator;
2. O ator pode visualizar a lista de chamados;
3. O caso de uso é encerrado;

A4: Todos os chamados:

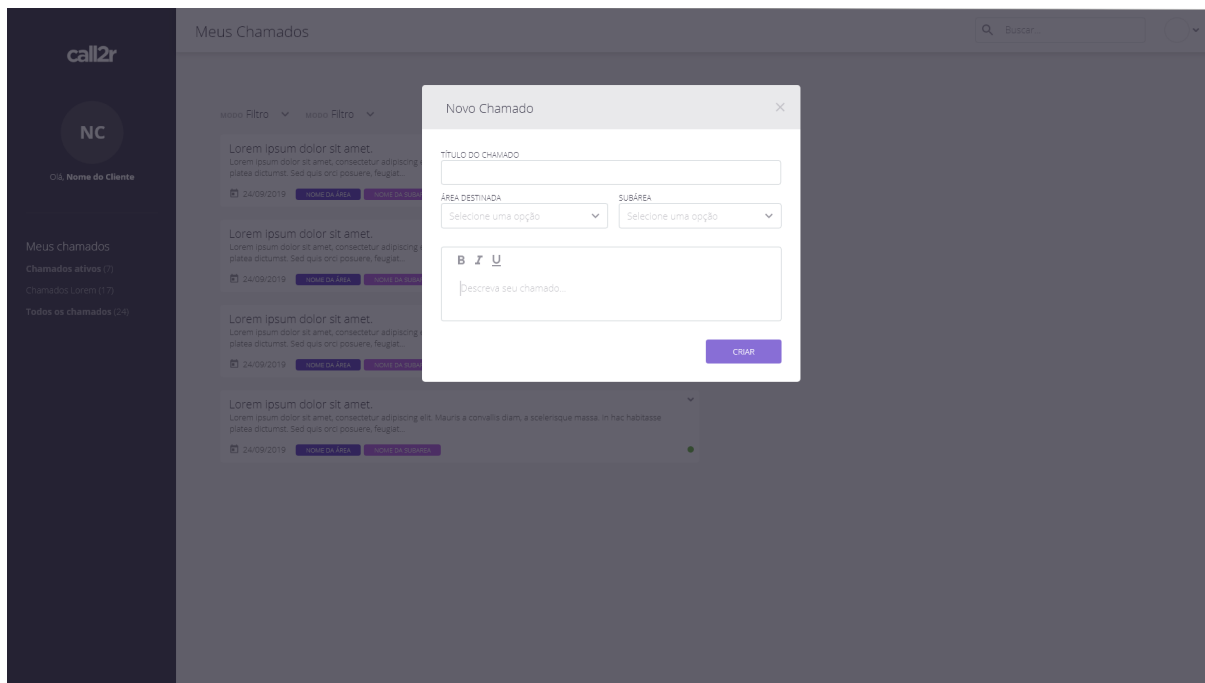
1. O sistema apresenta uma lista com chamados com estado "Em atendimento"ou "Aguardando atendimento";
2. O ator pode visualizar a lista de chamados;
3. O caso de uso é encerrado;

Fluxos de Exceção: Não se aplica.

Regras de Negócio: O usuário do tipo Gerente Suporte tem acesso a todos os chamados no estado "Em atendimento"e "Aguardando suporte".

UC012 - Abrir Chamado

Figura B.15: TELA DE ABERTURA DE NOVO CHAMADO



FONTE: Os autores (2019)

Descrição: O ator pode abrir um chamado, informando a empresa responsável, descrição, área e categoria.

Atores: Gerente cliente e cliente.

Pré-Condição: O ator deve estar logado como cliente ou gerente cliente.

Pós-Condição: O ator terá criado um novo chamado.

Fluxo de Eventos Principal:

- P1.** O ator clica em Novo na tela inicial;
- P2.** O sistema apresenta uma tela de criação de chamado;
- P3.** O ator preenche os dados do chamado;
- P4.** O ator clica em Salvar (E1);
- P5.** O sistema salva os dados do chamado;
- P6.** O sistema retorna ao ator os dados do chamado criado;
- P7.** O caso de uso é encerrado;

Fluxos Alternativos: Não se aplica.

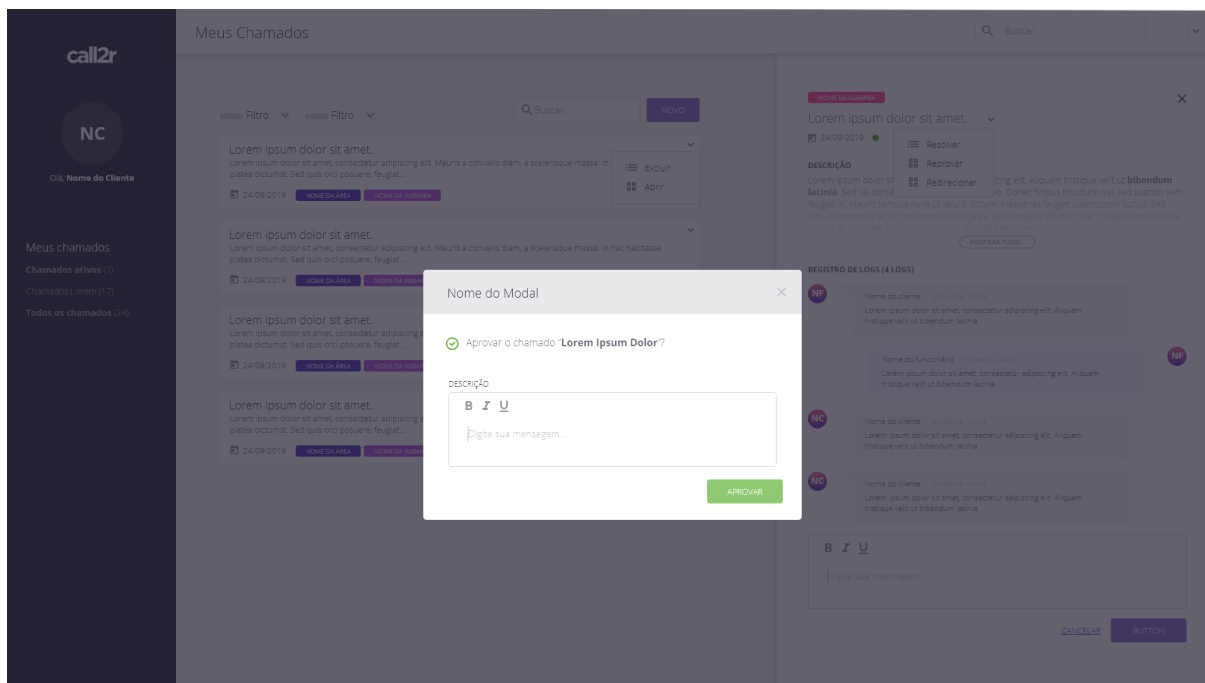
Fluxos de Exceção:

E1: Chamado vazio ou campos vazios:

1. O sistema retorna uma mensagem de erro informando que existem campos vazios;
2. O caso de uso é encerrado;

UC013 - Finalizar Chamado

Figura B.16: TELA DE APROVAÇÃO DE CHAMADO



FONTE: OS AUTORES (2019)

Descrição: O ator pode finalizar um chamado, indicando que o atendimento realizado foi aprovado.

Atores: Gerente cliente e cliente.

Pré-Condição: O ator deve estar logado como cliente ou gerente cliente e o chamado deve estar no estado "Aguardando cliente".

Pós-Condição: O chamado selecionado é aprovado e finalizado.

Fluxo de Eventos Principal:

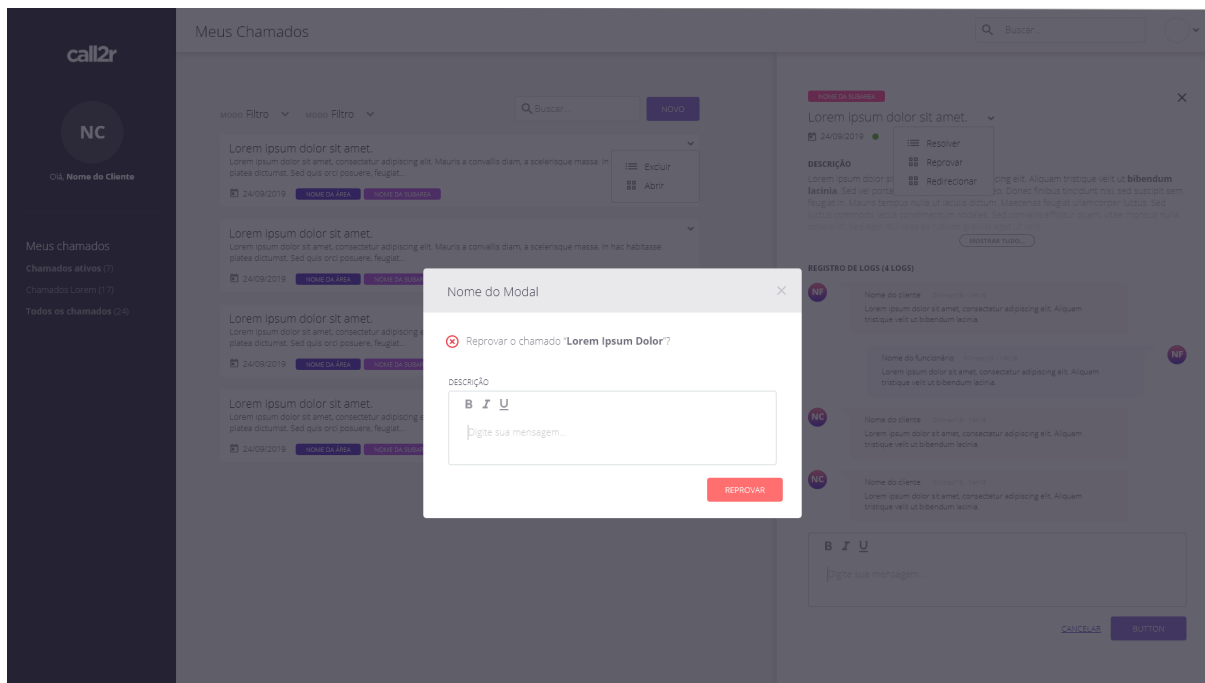
- P1.** O ator visualiza um chamado com estado "Aguardando cliente";
- P2.** O ator clica em Aprovar chamado;
- P3.** O sistema altera o estado do chamado para "Aprovado" e salva;
- P4.** O sistema retorna uma mensagem informando que o chamado foi aprovado;
- P5.** O caso de uso é encerrado;

Fluxos Alternativos: Não se aplica.

Fluxos de Exceção: Não se aplica.

UC014 - Reprovar Chamado

Figura B.17: TELA DE REPROVAÇÃO DE CHAMADO



FONTE: OS AUTORES (2019)

Descrição: O ator pode reprovar um chamado, indicando que o atendimento realizado não foi aprovado.

Atores: Gerente cliente e cliente.

Pré-Condição: O ator deve estar logado como cliente ou gerente cliente e o chamado deve estar no estado "Aguardando cliente".

Pós-Condição: O chamado selecionado é reprovalido e tem estado alterado para "Aguardando suporte".

Fluxo de Eventos Principal:

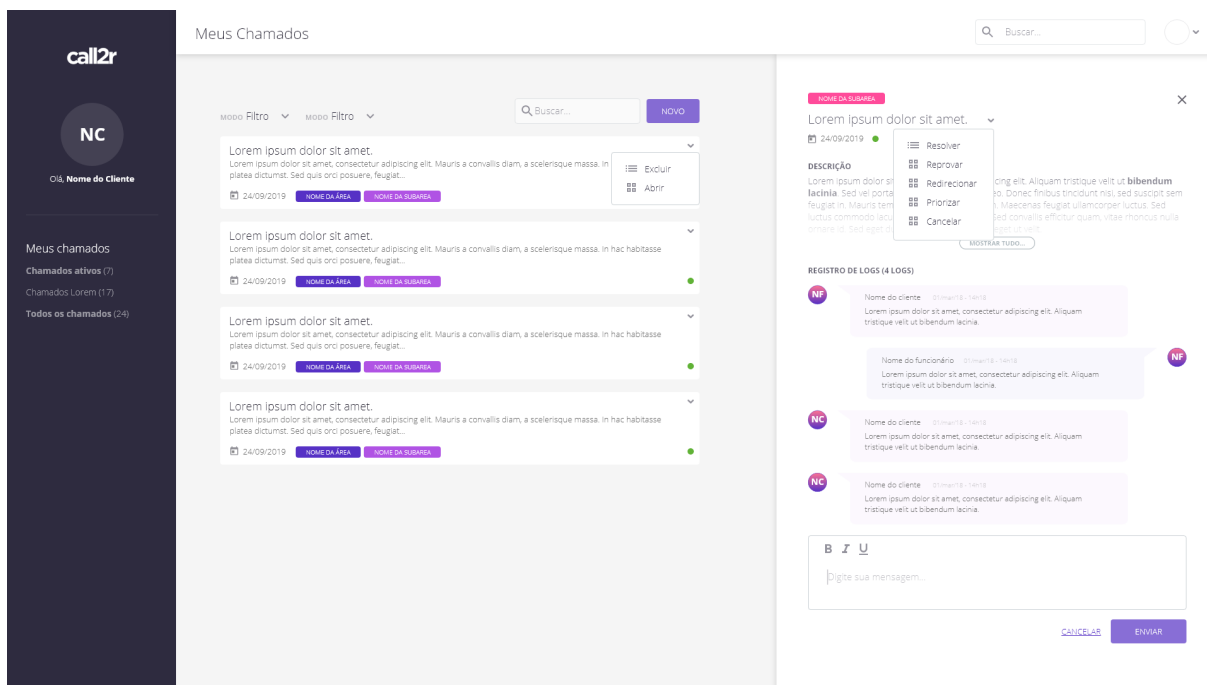
- P1.** O ator visualiza um chamado com estado "Aguardando cliente";
- P2.** O ator clica em Reprovar chamado;
- P3.** O sistema altera o estado do chamado para "Aguardando suporte" e salva;
- P4.** O sistema retorna uma mensagem informando que o chamado retornou para a lista de chamados do suporte;
- P5.** O caso de uso é encerrado;

Fluxos Alternativos: Não se aplica.

Fluxos de Exceção: Não se aplica.

UC015 - Resolver Chamado

Figura B.18: TELA DE RESOLUÇÃO DE CHAMADO



FONTE: OS AUTORES (2019)

Descrição: O ator pode resolver um chamado, indicando que o atendimento foi realizado.

Atores: Gerente suporte e suporte.

Pré-Condição: O ator deve estar logado como suporte ou gerente suporte e o chamado deve estar no estado "Em atendimento".

Pós-Condição: O chamado selecionado é resolvido e alterado para "Aguardando cliente".

Fluxo de Eventos Principal:

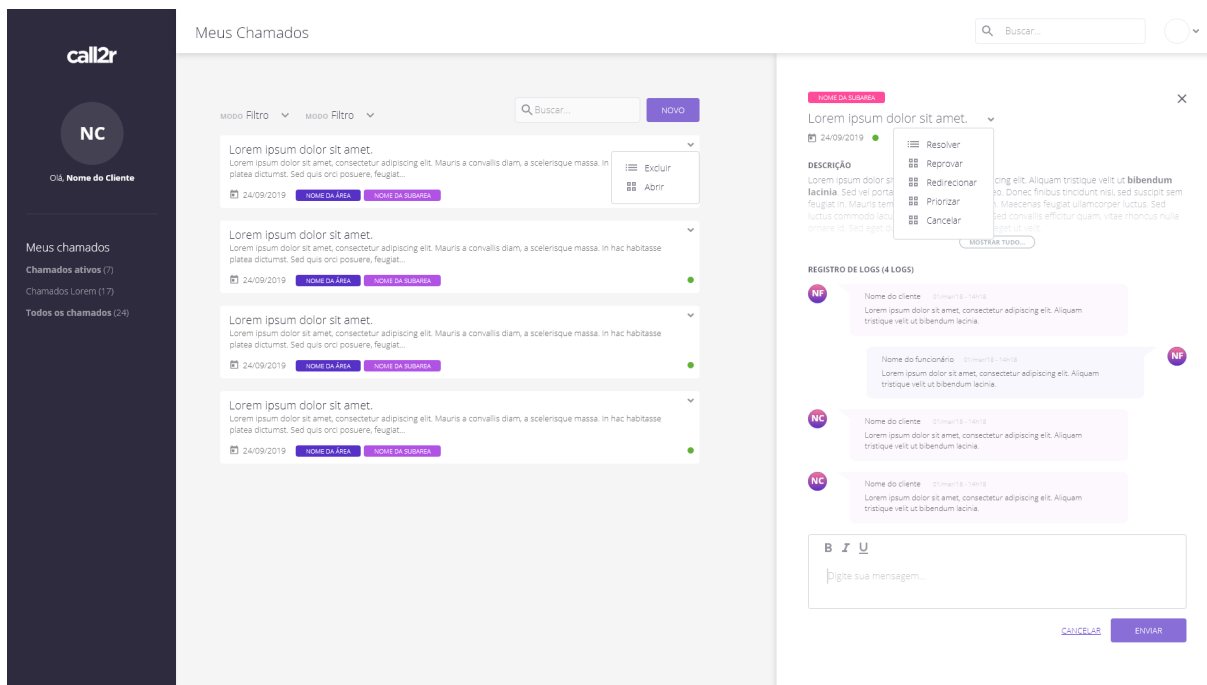
- P1.** O ator visualiza um chamado com estado "Em atendimento";
- P2.** O ator clica em Resolver chamado;
- P3.** O sistema altera o estado do chamado para "Aguardando cliente" e salva;
- P4.** O sistema informa ao solicitante do chamado que o chamado está no estado "Aguardando cliente";
- P5.** O sistema retorna uma mensagem informando que o chamado foi resolvido;
- P6.** O caso de uso é encerrado;

Fluxos Alternativos: Não se aplica.

Fluxos de Exceção: Não se aplica.

UC016 - Atender Chamado

Figura B.19: TELA DE ATENDIMENTO DE CHAMADO



FONTE: OS AUTORES (2019)

Descrição: O ator pode atender um chamado, indicando que o atendimento está sendo realizado.

Atores: Gerente suporte e suporte.

Pré-Condição: O ator deve estar logado como suporte ou gerente suporte e o chamado deve estar no estado "Aguardando suporte".

Pós-Condição: O chamado selecionado é alterado para "Em atendimento" e o responsável é alterado para o usuário que está atendendo.

Fluxo de Eventos Principal:

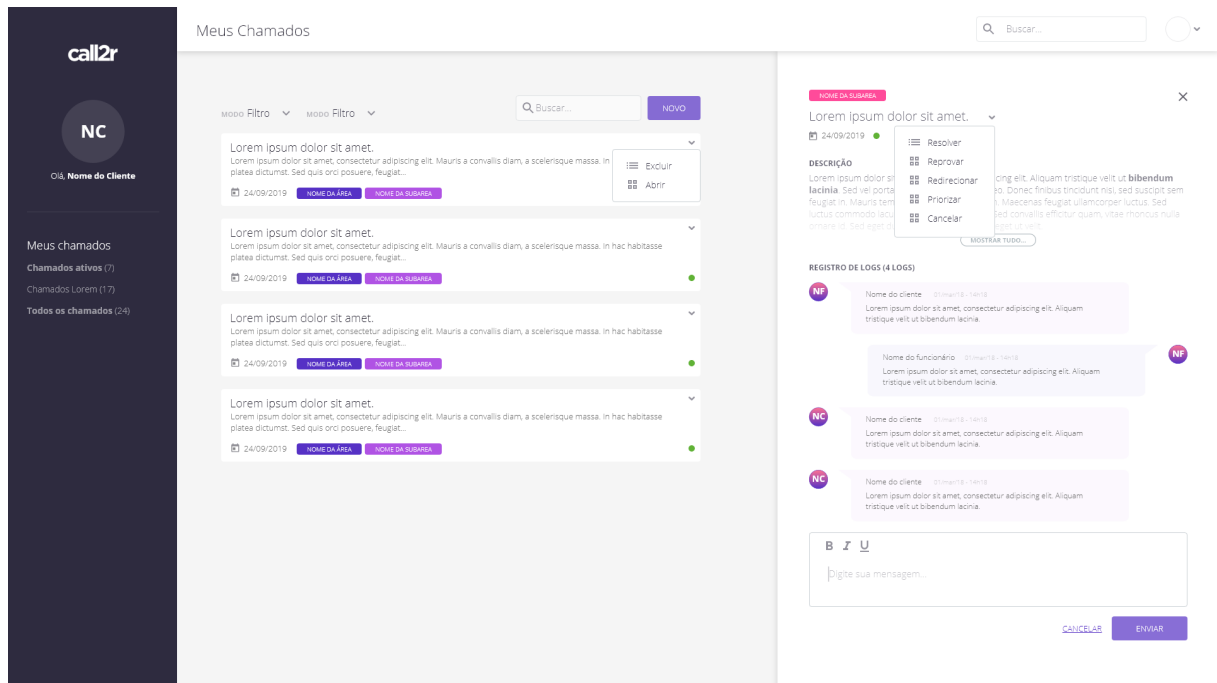
- P1.** O ator visualiza um chamado com estado "Aguardando suporte" ou "Reprovado";
- P2.** O ator clica em Atender chamado;
- P3.** O sistema altera o estado do chamado para "Em atendimento" e altera o responsável para o usuário que está atendendo o chamado e salva;
- P4.** O caso de uso é encerrado;

Fluxos Alternativos: Não se aplica.

Fluxos de Exceção: Não se aplica.

UC017 - Visualizar Chamado

Figura B.20: TELA DE DETALHES DO CHAMADO



FONTE: OS AUTORES (2019)

Descrição: O ator pode visualizar os dados de um chamado.

Atores: Gerente cliente, cliente, gerente suporte e suporte.

Pré-Condição: O ator deve estar logado como cliente, gerente cliente, suporte ou gerente suporte e deve existir um chamado válido.

Pós-Condição: O ator poderá ter visualizado os dados do chamado.

Fluxo de Eventos Principal:

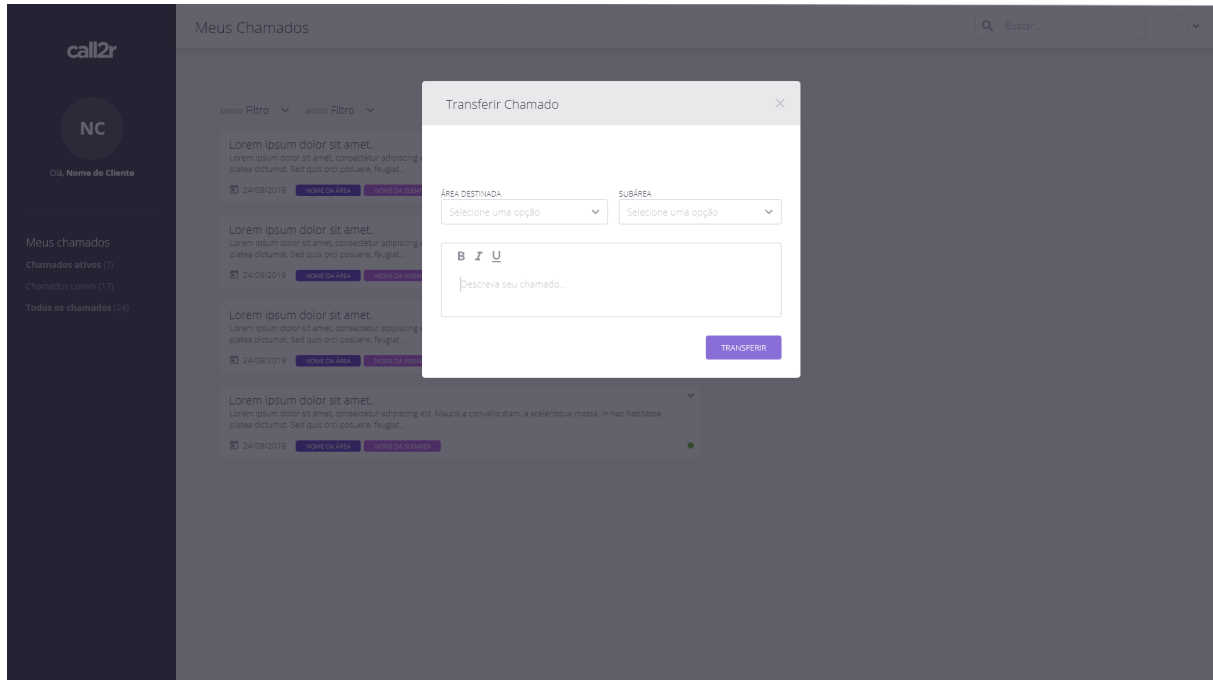
- P1.** O ator clica em um chamado na lista de chamados;
- P2.** O sistema apresenta uma tela com os dados do chamado;
- P3.** O ator pode visualizar os dados do chamado;
- P4.** O caso de uso é encerrado;

Fluxos Alternativos: Não se aplica.

Fluxos de Exceção: Não se aplica.

UC018 - Transferir Chamado

Figura B.21: TELA DE TRANSFERÊNCIA DE CHAMADO



FONTE: Os autores (2019)

Descrição: O ator pode transferir um chamado para outra empresa suporte.

Atores: Suporte.

Pré-Condição: O ator deve estar logado como suporte e deve existir um chamado válido que ele esta atendendo.

Pós-Condição: O ator poderá ter transferido um chamado para outra empresa suporte.

Fluxo de Eventos Principal:

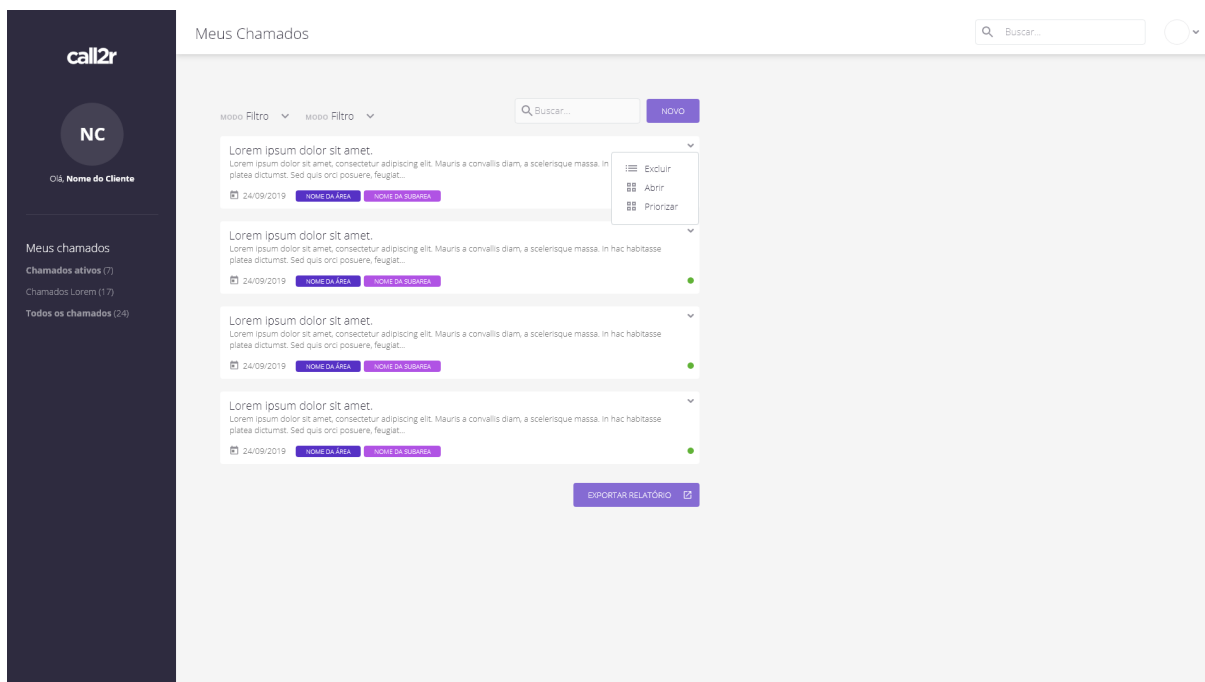
- P1.** O ator visualiza um chamado;
- P2.** O ator clica em Transferir chamado;
- P3.** O sistema apresenta uma tela para selecionar a empresa suporte destino;
- P4.** O ator seleciona a empresa suporte destino;
- P5.** O sistema apresenta uma tela de confirmação;
- P6.** O ator confirma;
- P7.** O sistema transfere o chamado para a empresa suporte destino;
- P8.** O caso de uso é encerrado;

Fluxos Alternativos: Não se aplica.

Fluxos de Exceção: Não se aplica.

UC019 - Pesquisar Chamado

Figura B.22: TELA DE PESQUISA DE CHAMADOS



FONTE: Os autores (2019)

Descrição: O ator pode pesquisar por um chamado.

Atores: Gerente cliente, cliente, gerente suporte e suporte.

Pré-Condição: O ator deve estar logado como gerente cliente, cliente, gerente suporte ou suporte.

Pós-Condição: O ator tem acesso a uma nova lista de chamados de acordo com a pesquisa.

Fluxo de Eventos Principal:

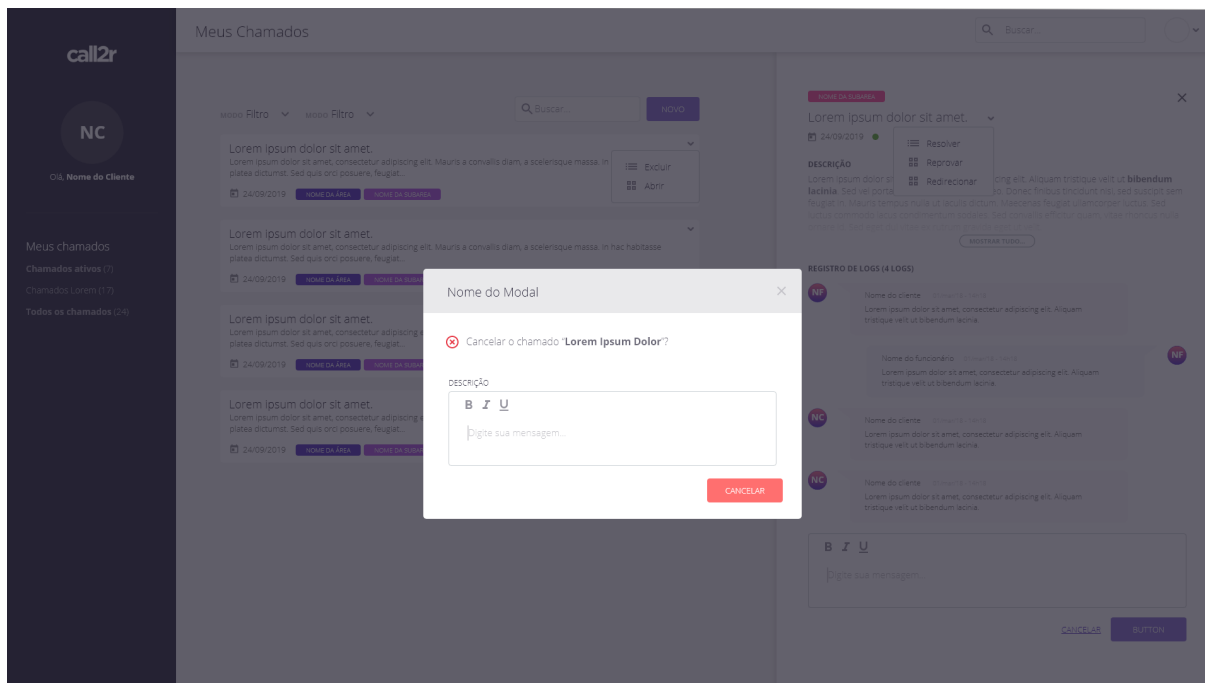
- P1. O ator preenche o campo de pesquisa;
- P2. O ator clica em pesquisar;
- P3. O sistema pesquisa os chamados correspondentes;
- P4. O sistema apresenta os chamados;
- P5. O ator pode visualizar a nova lista de chamados;
- P6. O caso de uso é encerrado;

Fluxos Alternativos: Não se aplica.

Fluxos de Exceção: Não se aplica.

UC020 - Cancelar Chamado

Figura B.23: TELA DE CANCELAMENTO DE CHAMADOS



FONTE: Os autores (2019)

Descrição: O ator pode cancelar chamados, apresentando uma justificativa para tal.

Atores: Gerente suporte.

Pré-Condição: O ator deve estar logado como gerente suporte e deve existir um chamado válido.

Pós-Condição: O chamado é cancelado e o solicitante é informado sobre o cancelamento.

Fluxo de Eventos Principal:

- P1.** O ator visualiza o chamado;
- P2.** O ator clica em Cancelar chamado;
- P3.** O sistema apresenta uma tela de confirmação;
- P4.** O ator preenche a justificativa e confirma o cancelamento (E1);
- P5.** O sistema informa ao ator que o chamado foi cancelado;
- P6.** O sistema informa ao solicitante sobre o cancelamento do chamado;
- P7.** O caso de uso é encerrado;

Fluxos Alternativos: Não se aplica.

Fluxos de Exceção:

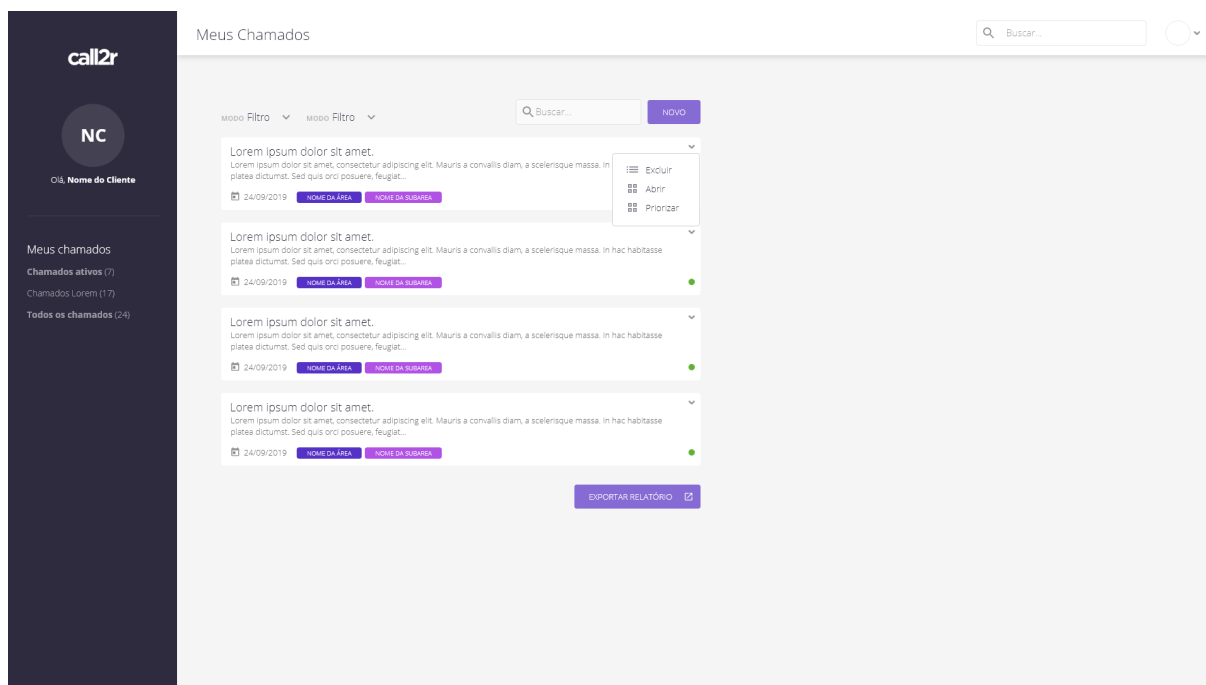
E1: Justificativa vazia:

1. O sistema retorna uma mensagem de erro informando que existem campos vazios;
2. O caso de uso é encerrado;

Regras de Negócio: O chamado só pode ser cancelado se estiver no estado Aguardando suporte ou Em atendimento.

UC021 - Filtrar Chamados

Figura B.24: TELA COM OPÇÕES DE FILTROS



FONTE: Os autores (2019)

Descrição: O ator pode filtrar os chamados na lista de chamados.

Atores: Gerente cliente, cliente, gerente suporte e suporte.

Pré-Condição: O ator deve estar logado como cliente, gerente cliente, suporte ou gerente suporte.

Pós-Condição: O ator poderá ter filtrado a lista de chamados de acordo com as opções selecionadas.

Fluxo de Eventos Principal:

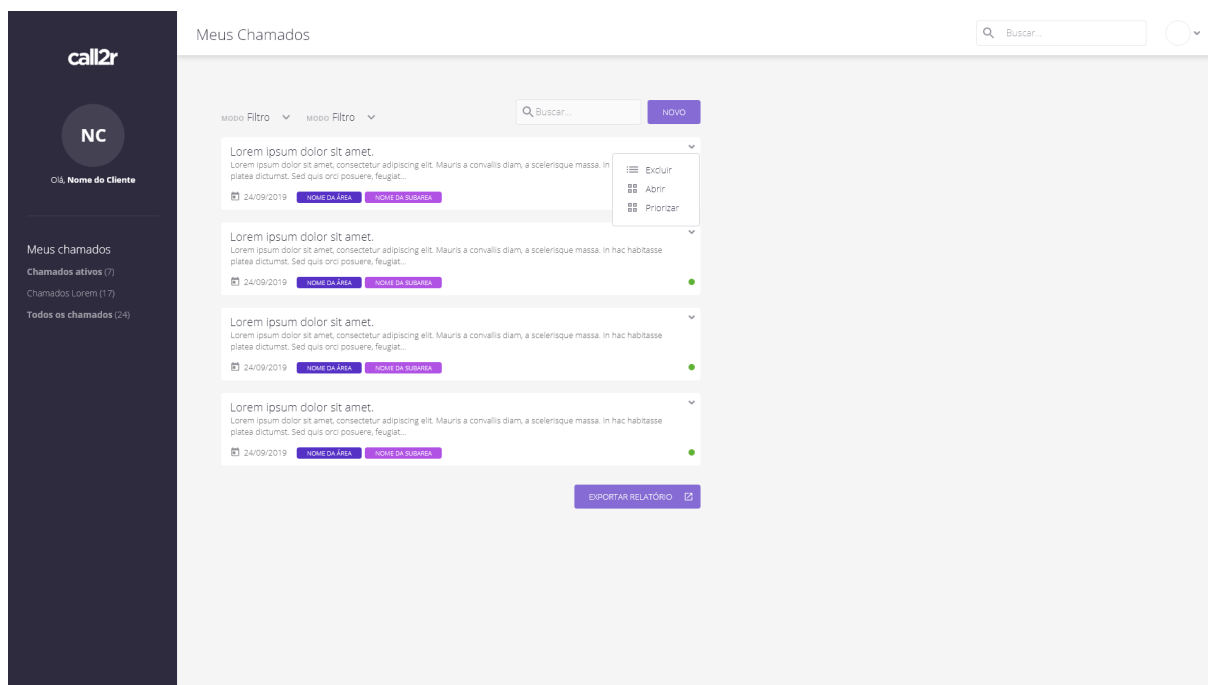
- P1.** O ator clica em filtros na lista de chamados;
- P2.** O sistema apresenta as opções de filtro para o ator;
- P3.** O ator seleciona as opções de filtro desejadas;
- P4.** O sistema filtra a lista de chamados de acordo com as opções selecionadas pelo cliente;
- P5.** O sistema apresenta a nova lista de chamados;
- P6.** O ator pode visualizar a nova lista de chamados;
- P7.** O caso de uso é encerrado;

Fluxos Alternativos: Não se aplica.

Fluxos de Exceção: Não se aplica.

UC022 - Gerar Relatório de Chamados

Figura B.25: TELA PARA EXPORTAR RELATÓRIO



FONTE: Os autores (2019)

Descrição: O ator pode gerar relatórios relacionados ao histórico, podendo salvá-los ou imprimi-los.

Atores: Gerente cliente e gerente suporte.

Pré-Condição: O ator deve estar logado como gerente cliente ou gerente suporte.

Pós-Condição: É gerado um relatório relacionado ao histórico de chamados.

Fluxo de Eventos Principal:

- P1.** O ator acessa a lista de chamados;
- P2.** O ator filtra a lista de chamados da forma que desejar;
- P3.** O ator clica em Exportar Relatório;
- P4.** O sistema apresenta uma tela de opção para o ator com as opções Salvar e Imprimir;
- P5.** O ator seleciona a opção (A1, A2);
- P6.** O sistema realiza a ação escolhida;
- P7.** O sistema retorna a lista de chamados;
- P8.** O caso de uso é encerrado;

Fluxos Alternativos:**A1:** Salvar relatório:

1. O ator seleciona o formato que deseja salvar o arquivo;
2. O ator clica em Salvar;
3. O sistema salva o arquivo no formato especificado;
4. O caso de uso é encerrado;

A2: Imprimir relatório:

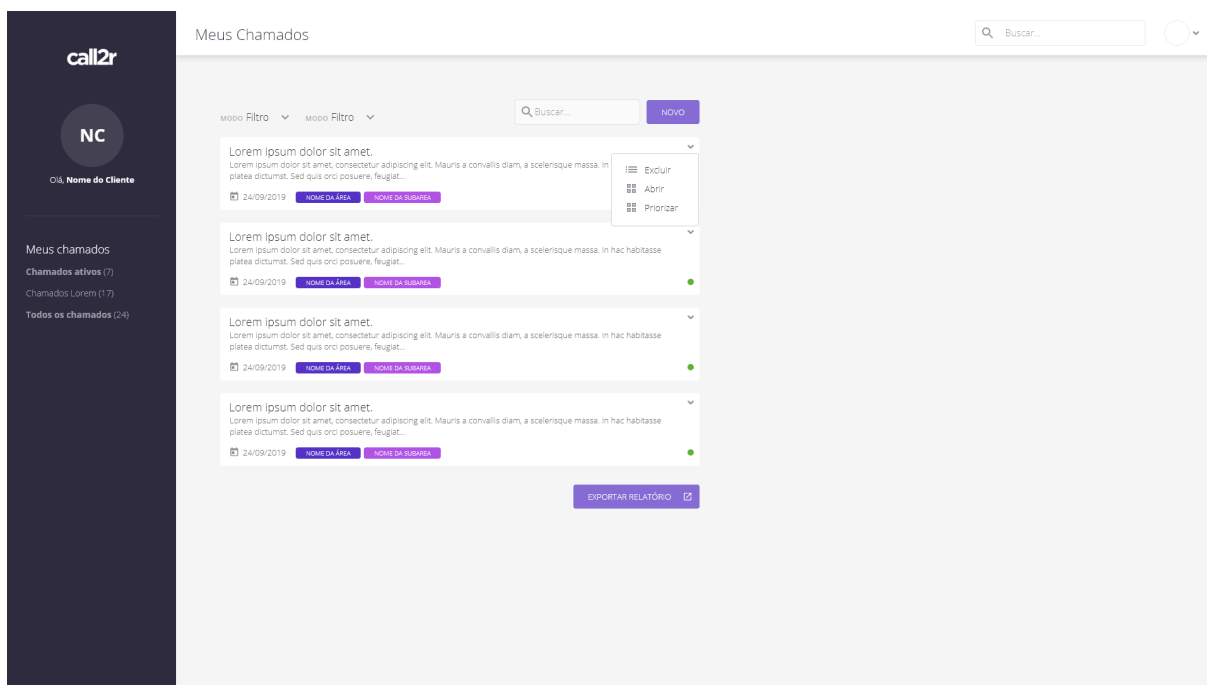
1. O ator clica em Imprimir;
2. O sistema imprime o relatório;
3. O caso de uso é encerrado;

Fluxos de Exceção: Não se aplica.

Regras de Negócio: O formato da impressão é pré-determinado como PDF.

UC023 - Alterar Prioridade de Chamado

Figura B.26: TELA PARA PRIORIZAR CHAMADO



FONTE: Os autores (2019)

Descrição: O ator pode alterar a prioridade de chamados, fazendo com que sua relevância seja alterada.

Atores: Gerente cliente.

Pré-Condição: O ator deve estar logado como gerente cliente e deve existir um chamado válido.

Pós-Condição: O chamado conterá uma *flag* visual informando a priorização.

Fluxo de Eventos Principal:

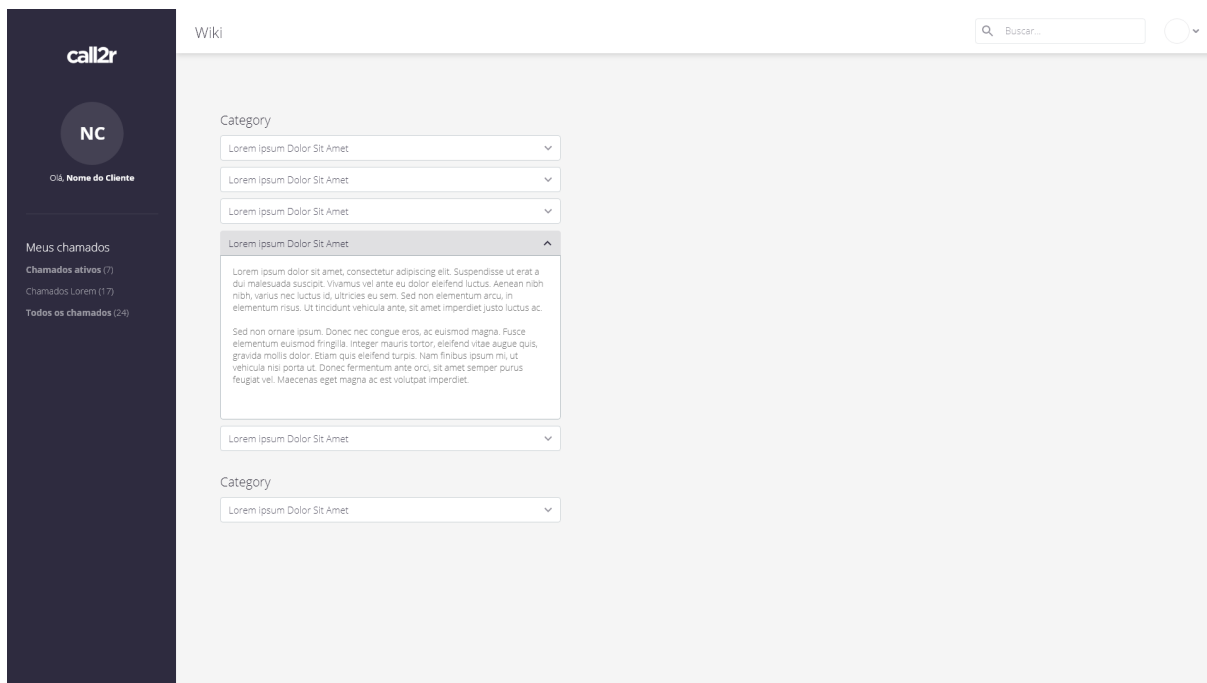
- P1.** O ator visualiza um chamado;
- P2.** O ator clica no botão priorizar chamado;
- P3.** O sistema apresenta uma tela de confirmação;
- P4.** O ator confirma;
- P5.** O sistema aplica uma flag visual na lista de chamados para este chamado;
- P6.** O sistema informa ao ator solicitante do chamado sobre a alteração de prioridade do chamado;
- P7.** O caso de uso é encerrado;

Fluxos Alternativos: Não se aplica.

Fluxos de Exceção: Não se aplica.

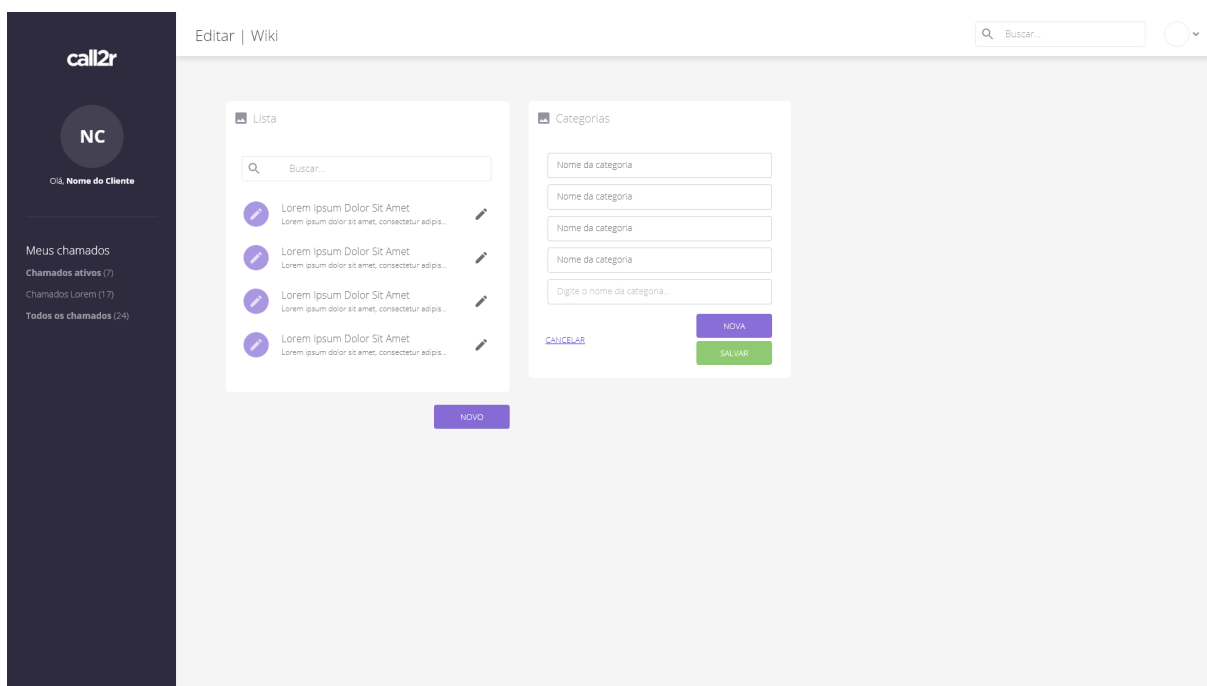
UC024 - Manter Wiki

Figura B.27: TELA DA WIKI



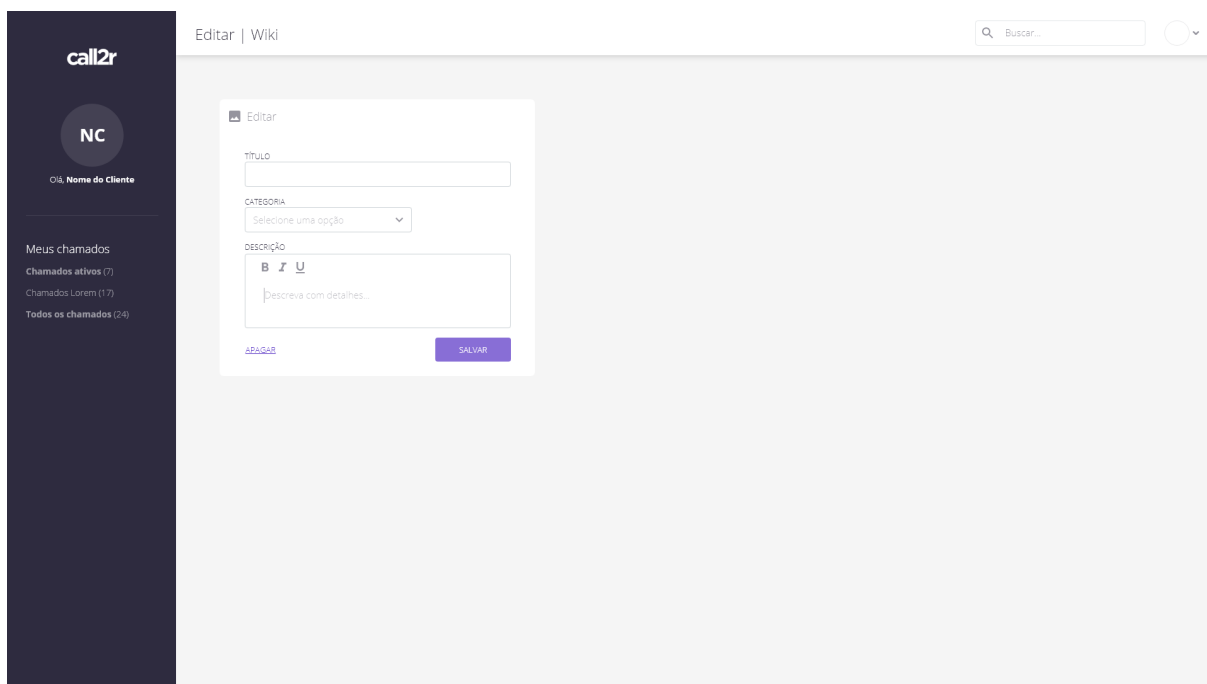
FONTE: Os autores (2019)

Figura B.28: TELA DE LISTA DE EDIÇÃO DA WIKI



FONTE: Os autores (2019)

Figura B.29: TELA DE DETALHES DE EDIÇÃO DA WIKI



FONTE: Os autores (2019)

Descrição: O ator pode criar, editar, excluir ou pesquisar artigos para documentar problemas.

Atores: Gerente suporte e suporte.

Pré-Condição: O ator deve estar logado como gerente suporte ou suporte.

Pós-Condição: O ator poderá ter criado, visualizado, alterado, excluído ou pesquisado um artigo na wiki.

Fluxo de Eventos Principal:

- P1.** O ator acessa a opção da wiki;
- P2.** O sistema apresenta uma lista de artigos;
- P3.** O ator pode visualizar os artigos em uma lista (A1, A2, A3, A4, A5);
- P4.** O caso de uso é encerrado;

Fluxos Alternativos:

A1: Criar artigo:

1. O ator clica em Novo;
2. O sistema apresenta uma tela de cadastro de artigo;
3. O ator preenche com o título e conteúdo do artigo;
4. O ator clica em Salvar (E1, E2);

5. O sistema cadastra o artigo;

6. O caso de uso é encerrado;

A2: Visualizar artigo:

1. O ator clica em um artigo na lista de artigos;

2. O sistema apresenta uma tela com cabeçalho, descrição e arquivos anexos;

3. O caso de uso é encerrado;

A3: Alterar artigo:

1. O ator clica em Editar em um artigo existente;

2. O sistema apresenta uma tela de alteração de artigo;

3. O ator altera o artigo;

4. O ator clica em Salvar (E1, E2);

5. O caso de uso é encerrado;

A4: Desativar artigo:

1. O ator clica em Desativar em um artigo existente;

2. O sistema apresenta uma tela de confirmação;

3. O ator confirma;

4. O sistema desativa o artigo;

5. O caso de uso é encerrado;

A5: Ativar artigo:

1. O ator visualiza um artigo;

2. O ator clica em Ativar;

3. O sistema ativa o artigo;

4. O caso de uso é encerrado;

A6: Pesquisar artigo:

1. O ator preenche o campo de pesquisa;

2. O ator clica em Pesquisar;

3. O sistema pesquisa os artigos correspondentes;

4. O sistema apresenta os artigos;

5. O caso de uso é encerrado;

Fluxos de Exceção:**E1:** Título já existente:

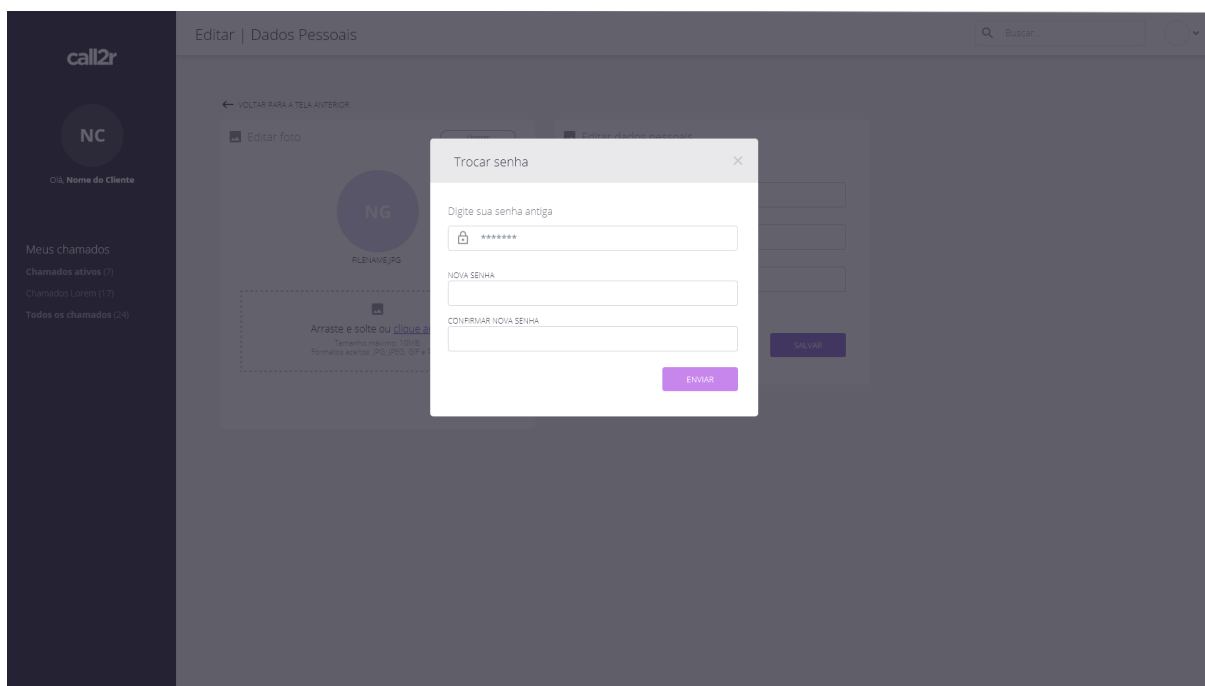
1. O sistema retorna uma mensagem de erro informando que já existe um artigo com o mesmo título;
2. O caso de uso é encerrado;

E2: Artigo vazio:

1. O sistema retorna uma mensagem de erro informando que existem campos vazios;
2. O caso de uso é encerrado;

UC025 - Alterar Senha

Figura B.30: TELA DE ALTERAÇÃO DE SENHA



FONTE: Os autores (2019)

Descrição: O ator pode atualizar a senha de acesso.

Atores: Gerente Cliente, cliente, gerente suporte e suporte.

Pré-Condição: O ator deve estar cadastrado no sistema.

Pós-Condição: A senha do ator é alterada.

Fluxo de Eventos Principal:

- P1.** O ator seleciona a opção de trocar senha;
- P2.** O sistema apresenta uma tela com os campos senha atual, nova senha e confirmação da nova senha;
- P3.** O ator preenche os campos;
- P4.** O ator clica em Alterar (E1, E2, E3);
- P5.** O sistema atualiza a senha do ator;
- P6.** O caso de uso é encerrado;

Fluxos Alternativos: Não se aplica.

Fluxos de Exceção:

- E1:** Senha atual errada:

1. O sistema retorna uma mensagem de erro informando que a senha está incorreta;
2. O caso de uso é encerrado;

E2: Confirmação da senha e senha nova diferentes:

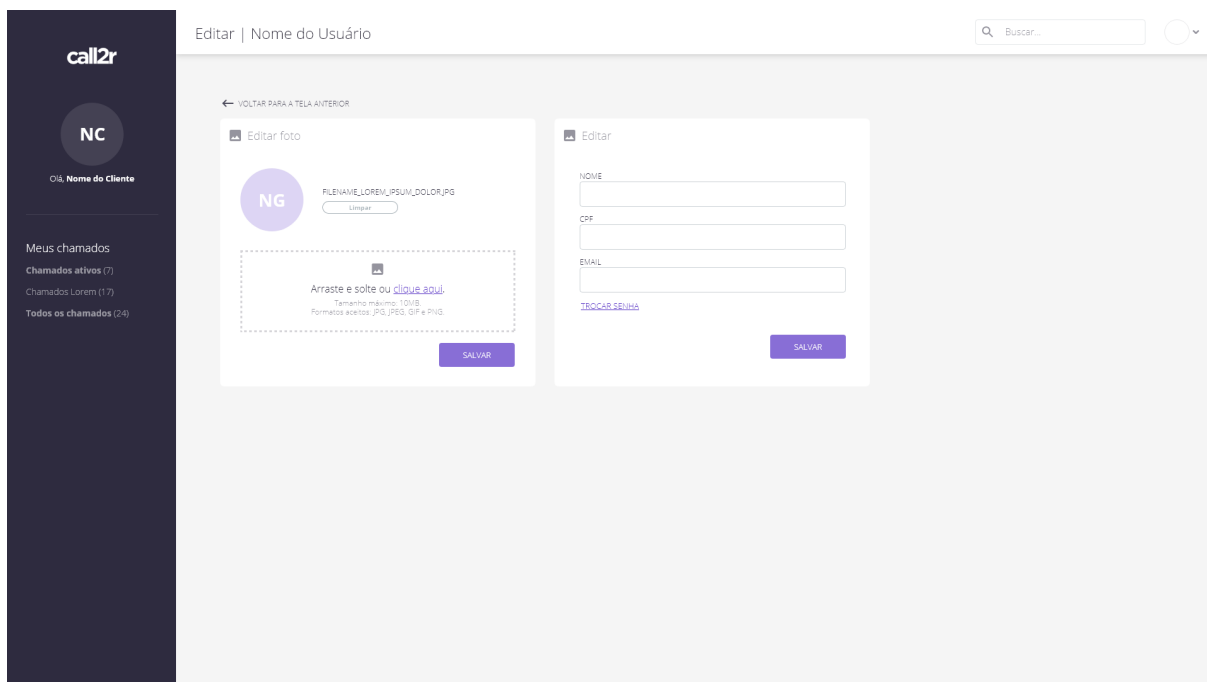
1. O sistema retorna uma mensagem de erro informando que a nova senha e a confirmação estão diferentes;
2. O caso de uso é encerrado;

E3: Nova senha igual a anterior:

1. O sistema retorna uma mensagem de erro informando que a nova senha e a anterior são iguais;
2. O caso de uso é encerrado;

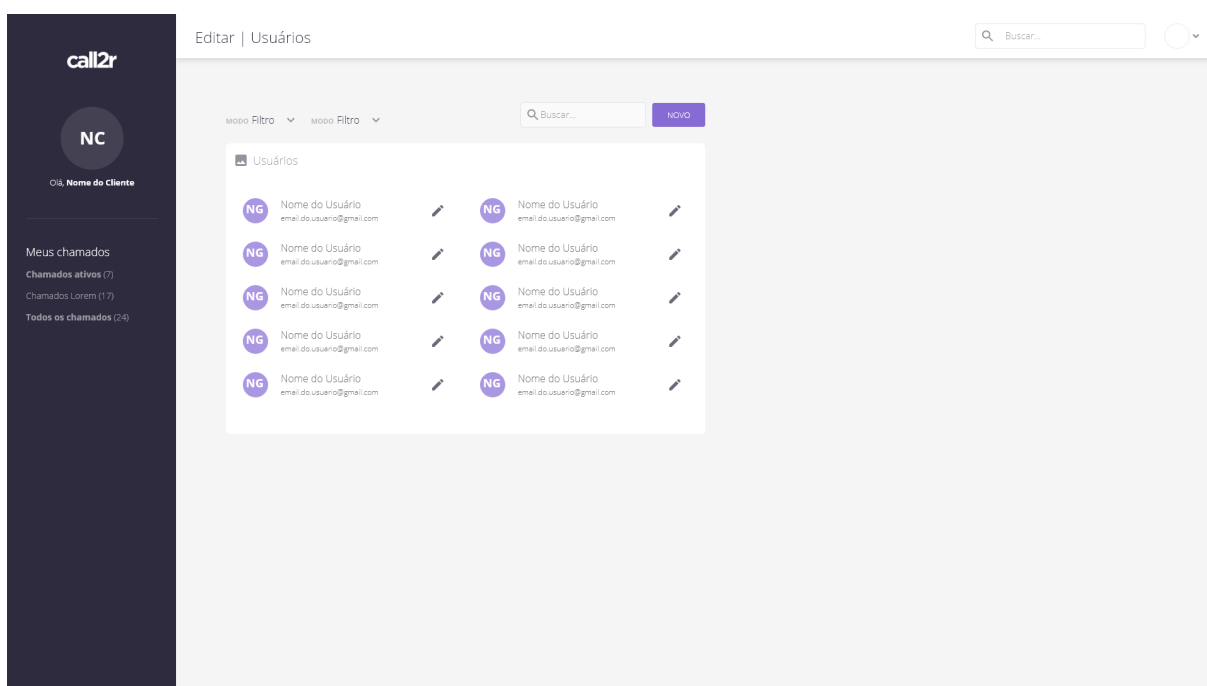
UC026 - Manter Suporte

Figura B.31: TELA DE EDIÇÃO DE USUÁRIO



FONTE: Os autores (2019)

Figura B.32: TELA DE LISTA DE USUÁRIOS



FONTE: Os autores (2019)

Descrição: O ator pode criar, visualizar, alterar ou excluir um usuário suporte.

Atores: Gerente suporte.

Pré-Condição: O ator deve estar logado como gerente suporte.

Pós-Condição: O ator poderá ter criado, alterado, excluído, visualizado ou pesquisado usuários do tipo suporte.

Fluxo de Eventos Principal:

- P1.** O ator seleciona a opção lista de suportes;
- P2.** O sistema carrega uma lista de usuários suportes;
- P3.** O ator pode visualizar os usuários suporte (A1, A2, A3, A4, A5, A6);
- P4.** O caso de uso é encerrado;

Fluxos Alternativos:

A1: Criar suporte:

- 1. O ator clica em Novo;
- 2. O sistema apresenta uma tela de cadastro de suporte;
- 3. O ator preenche os dados do suporte;
- 4. O ator clica em Salvar (E1, E2);
- 5. O sistema cadastra o suporte;
- 6. O caso de uso é encerrado;

A2: Visualizar suporte:

- 1. O ator clica em um suporte na lista de suportes;
- 2. O sistema apresenta uma tela com os dados do suporte selecionado;
- 3. O ator pode visualizar os dados;
- 4. O caso de uso é encerrado;

A3: Alterar suporte:

- 1. O ator clica em Editar em um suporte existente;
- 2. O sistema apresenta uma tela de alteração dos dados do suporte;
- 3. O ator pode alterar os dados do suporte;
- 4. O ator clica em Salvar (E1, E2);
- 5. O sistema apresenta uma tela de confirmação das alterações;
- 6. O ator confirma;

7. O sistema salva as alterações;
8. O caso de uso é encerrado;

A4: Desativar suporte:

1. O ator clica em Desativar em um suporte existente;
2. O sistema apresenta uma tela de confirmação da desativação;
3. O ator confirma a desativação;
4. O sistema desativa o suporte;
5. O caso de uso é encerrado;

A5: Ativar suporte:

1. O ator visualiza um suporte;
2. O ator clica em Ativar;
3. O sistema ativa o suporte;
4. O caso de uso é encerrado;

A6: Pesquisar suporte:

1. O ator preenche o campo de pesquisa;
2. O ator clica em Pesquisar;
3. O sistema pesquisa os suportes correspondentes;
4. O sistema apresenta os suportes;
5. O caso de uso é encerrado;

Fluxos de Exceção:**E1:** Suporte já existe:

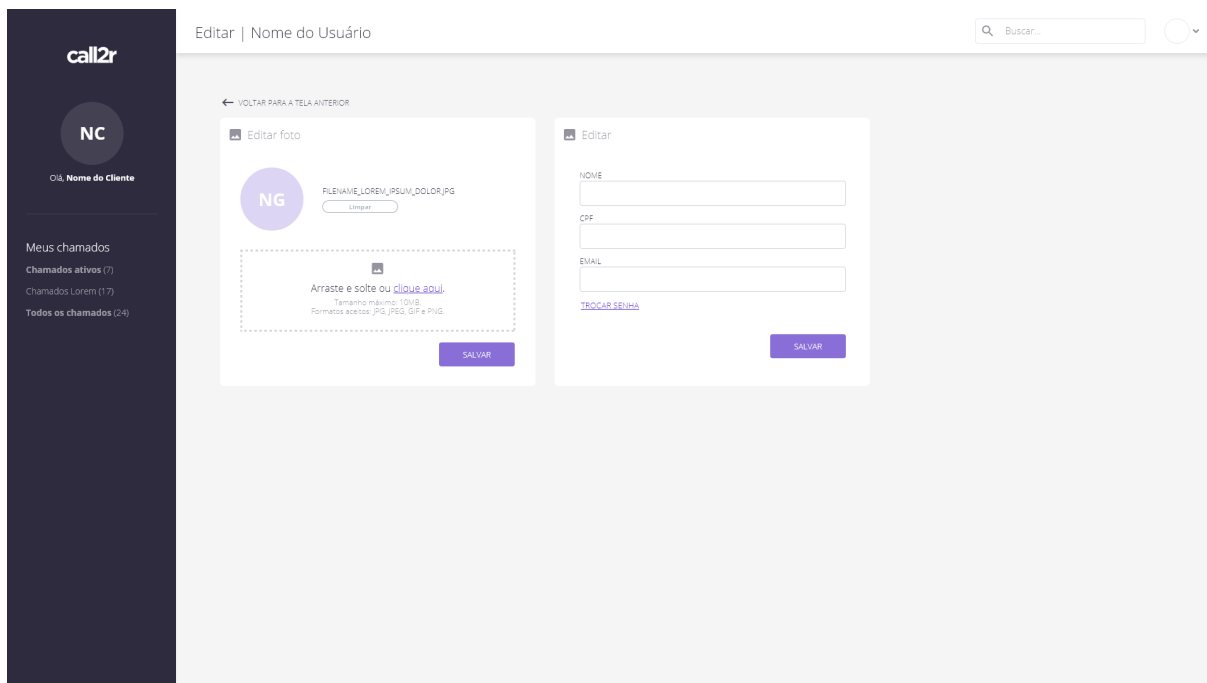
1. O sistema retorna uma mensagem de erro informando que o usuário suporte já existe;
2. O caso de uso é encerrado;

E2: Suporte vazio ou campos vazios:

1. O sistema retorna uma mensagem de erro informando que existem campos vazios;
2. O caso de uso é encerrado;

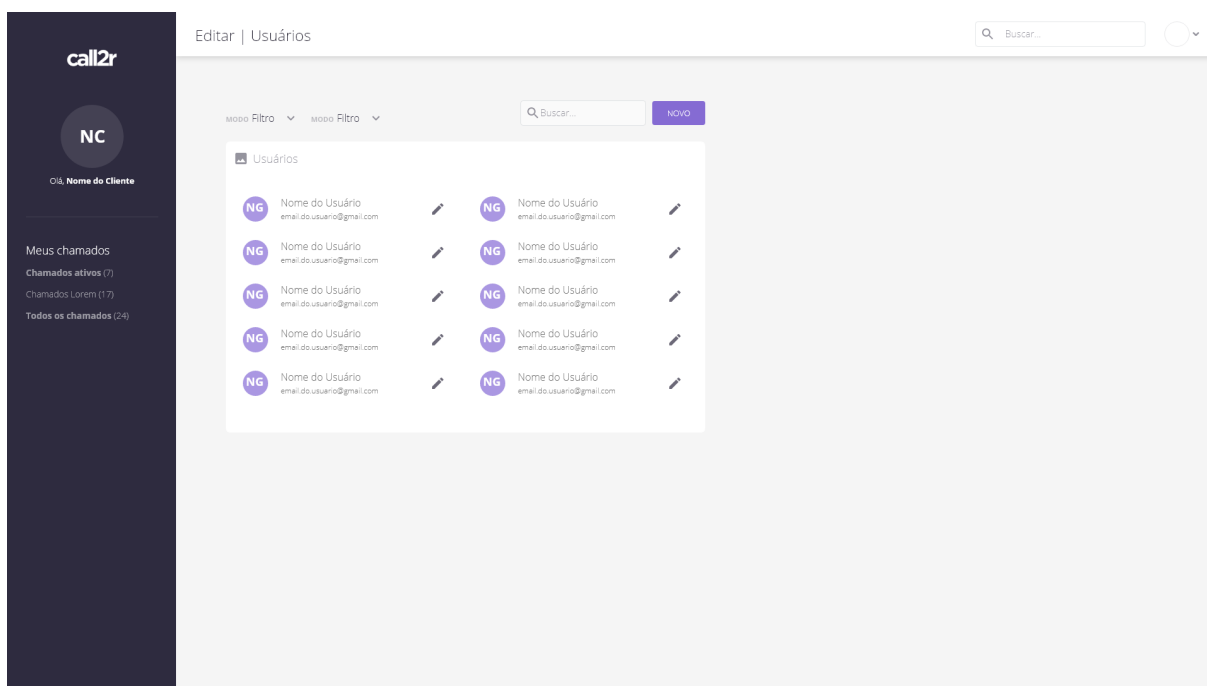
UC027 - Manter Cliente

Figura B.33: TELA DE EDIÇÃO DE USUÁRIO



FONTE: Os autores (2019)

Figura B.34: TELA DE LISTA DE USUÁRIOS



FONTE: Os autores (2019)

Descrição: O ator pode criar, visualizar, alterar ou excluir um usuário cliente.

Atores: Gerente cliente.

Pré-Condição: O ator deve estar logado como gerente cliente.

Pós-Condição: O ator poderá ter criado, alterado, excluído, visualizado ou pesquisado usuários do tipo cliente.

Fluxo de Eventos Principal:

- P1.** O ator seleciona a opção lista de clientes;
- P2.** O sistema carrega uma lista de usuários clientes;
- P3.** O ator pode visualizar os usuários cliente (A1, A2, A3, A4, A5, A6);
- P4.** O caso de uso é encerrado;

Fluxos Alternativos:

A1: Criar cliente:

- 1. O ator clica em Novo;
- 2. O sistema apresenta uma tela de cadastro de cliente;
- 3. O ator preenche os dados do cliente;
- 4. O ator clica em Salvar (E1, E2);
- 5. O sistema cadastra o cliente;
- 6. O caso de uso é encerrado;

A2: Visualizar cliente:

- 1. O ator clica em um cliente na lista de clientes;
- 2. O sistema apresenta uma tela com os dados do cliente selecionado;
- 3. O ator pode visualizar os dados;
- 4. O caso de uso é encerrado;

A3: Alterar cliente:

- 1. O ator clica em Editar em um cliente existente;
- 2. O sistema apresenta uma tela de alteração dos dados do cliente;
- 3. O ator pode alterar os dados do cliente;
- 4. O ator clica em Salvar (E1, E2);
- 5. O sistema apresenta uma tela de confirmação das alterações;
- 6. O ator confirma;

7. O sistema salva as alterações;
8. O caso de uso é encerrado;

A4: Desativar cliente:

1. O ator clica em Desativar em um cliente existente;
2. O sistema apresenta uma tela de confirmação da desativação;
3. O ator confirma a desativação;
4. O sistema desativa o cliente;
5. O caso de uso é encerrado;

A5: Ativar cliente:

1. O ator visualiza um cliente;
2. O ator clica em Ativar;
3. O sistema ativa o cliente;
4. O caso de uso é encerrado;

A6: Pesquisar cliente:

1. O ator preenche o campo de pesquisa;
2. O ator clica em Pesquisar;
3. O sistema pesquisa os clientes correspondentes;
4. O sistema apresenta os clientes;
5. O caso de uso é encerrado;

Fluxos de Exceção:**E1:** Cliente já existe:

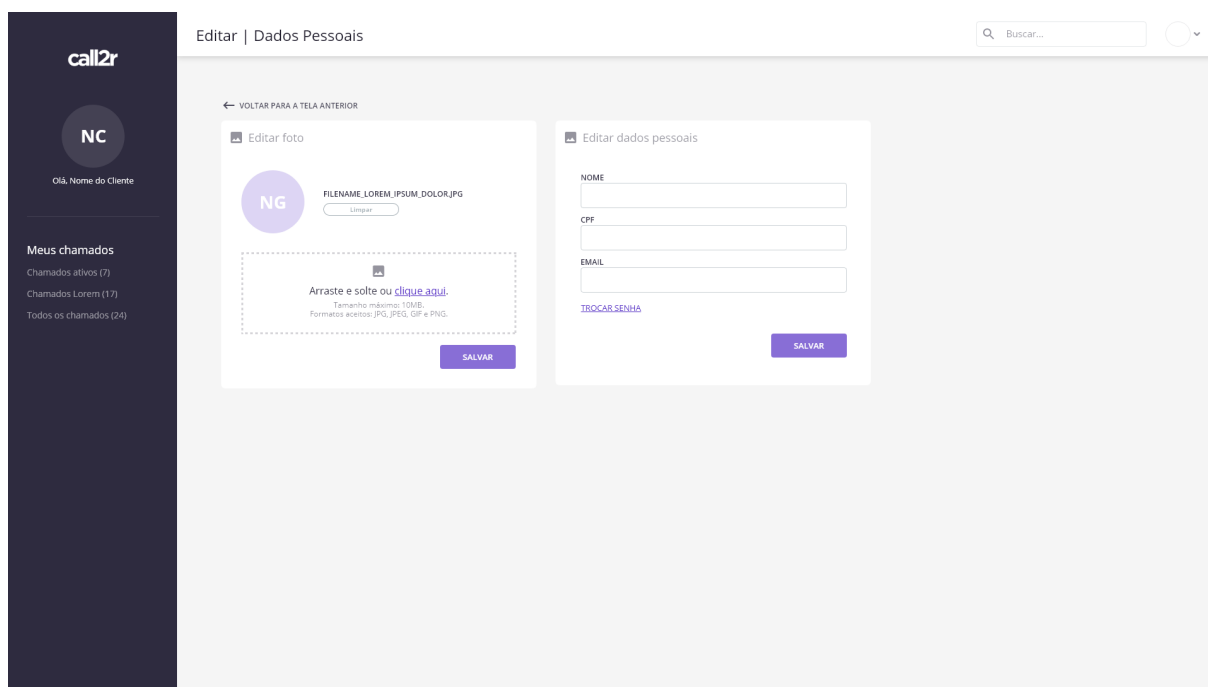
1. O sistema retorna uma mensagem de erro informando que o usuário cliente já existe;
2. O caso de uso é encerrado;

E2: Cliente vazio ou campos vazios:

1. O sistema retorna uma mensagem de erro informando que existem campos vazios;
2. O caso de uso é encerrado;

UC028 - Alterar Perfil Próprio

Figura B.35: TELA ALTERAR PERFIL



FONTE: Os autores (2019)

Descrição: O ator pode visualizar e alterar seu perfil.

Atores: Suporte, cliente, gerente suporte e gerente cliente.

Pré-Condição: O ator deve estar logado no sistema.

Pós-Condição: O ator altera seu perfil.

Fluxo de Eventos Principal:

- P1.** O ator seleciona a opção para visualizar o seu perfil;
- P2.** O sistema carrega os campos que ele pode alterar;
- P3.** O ator preenche esses campos com novos dados;
- P4.** O ator clica em Salvar (E1);
- P5.** O sistema altera os dados do usuário;
- P6.** O caso de uso é encerrado;

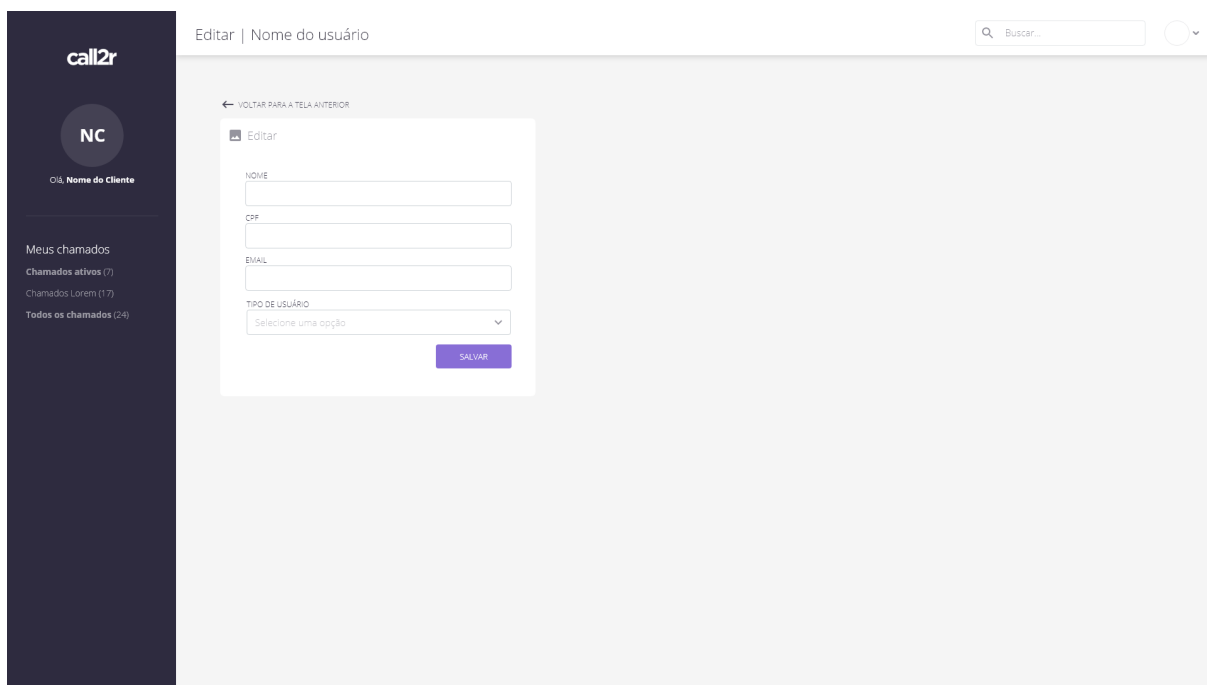
Fluxos de Exceção:

E1: E-mail existente:

- 1. O sistema retorna uma mensagem de erro informando o e-mail já existe;
- 2. O caso de uso é encerrado;

UC029 - Alterar Perfil

Figura B.36: TELA DE ALTERAÇÃO DE PERFIL



FONTE: OS AUTORES (2019)

Descrição: O ator pode alterar o perfil do usuário de gerente cliente para cliente ou de gerente suporte para suporte.

Atores: Admin.

Pré-Condição: O ator deve estar logado como Admin e deve existir um usuário gerente cliente ou gerente suporte válido.

Pós-Condição: O usuário selecionado deixará de ser gerente.

Fluxo de Eventos Principal:

- P1.** O ator clica em um gerente cliente ou gerente suporte na lista de gerentes cliente ou de gerentes suporte;
- P2.** O sistema apresenta uma tela com os dados do usuário;
- P3.** O ator clica em Alterar Perfil;
- P4.** O sistema apresenta uma tela de confirmação;
- P5.** O ator confirma;
- P6.** O sistema altera o perfil do usuário selecionado para cliente ou suporte;
- P7.** O caso de uso é encerrado;

Fluxos Alternativos: Não se aplica.

Fluxos de Exceção: Não se aplica.

UC030 - Alterar Perfil para Gerente

Figura B.37: TELA DE ALTERAÇÃO DE PERFIL

A imagem mostra a interface de usuário para a alteração de perfil de um usuário no sistema call2r. O layout é dividido em uma barra lateral esquerda e uma área principal de conteúdo.

Barra Lateral Esquerda:

- Logo "call2r" no topo.
- Avatar circular com as letras "NC".
- Saudação: "Olá, Nome do Cliente".
- Seção "Meus chamados" com sub-itens: "Chamados ativos (7)", "Chamados Lorem (1/7)" e "Todos os chamados (2/4)".

Área Principal de Conteúdo:

- Botão "Editar" no topo esquerdo.
- Formulário de edição com campos para: "NOME", "CPF", "EMAIL" e "TIPO DE USUÁRIO" (menu suspenso com a opção "Selecione uma opção").
- Botão "SALVAR" no canto inferior direito do formulário.
- Botão "VOLTAR PARA A TELA ANTERIOR" no topo esquerdo da área principal.
- Barra de busca "Buscar..." no topo direito.

FONTE: OS AUTORES (2019)

Descrição: O ator pode alterar o perfil de um cliente ou suporte para gerente cliente ou gerente suporte, respectivamente.

Atores: Gerente cliente e gerente suporte.

Pré-Condição: O ator deve estar logado como gerente cliente e deve existir um usuário cliente válido ou o ator deve estar logado como gerente suporte e deve existir um usuário suporte válido.

Pós-Condição: O usuário selecionado passará a ser gerente.

Fluxo de Eventos Principal:

- P1.** O ator clica em um cliente ou suporte na lista de clientes ou de suportes;
- P2.** O sistema apresenta uma tela com os dados do usuário;
- P3.** O ator clica em Alterar Perfil;
- P4.** O sistema apresenta uma tela de confirmação;
- P5.** O ator confirma;
- P6.** O sistema altera o perfil do usuário selecionado para gerente;
- P7.** O caso de uso é encerrado;

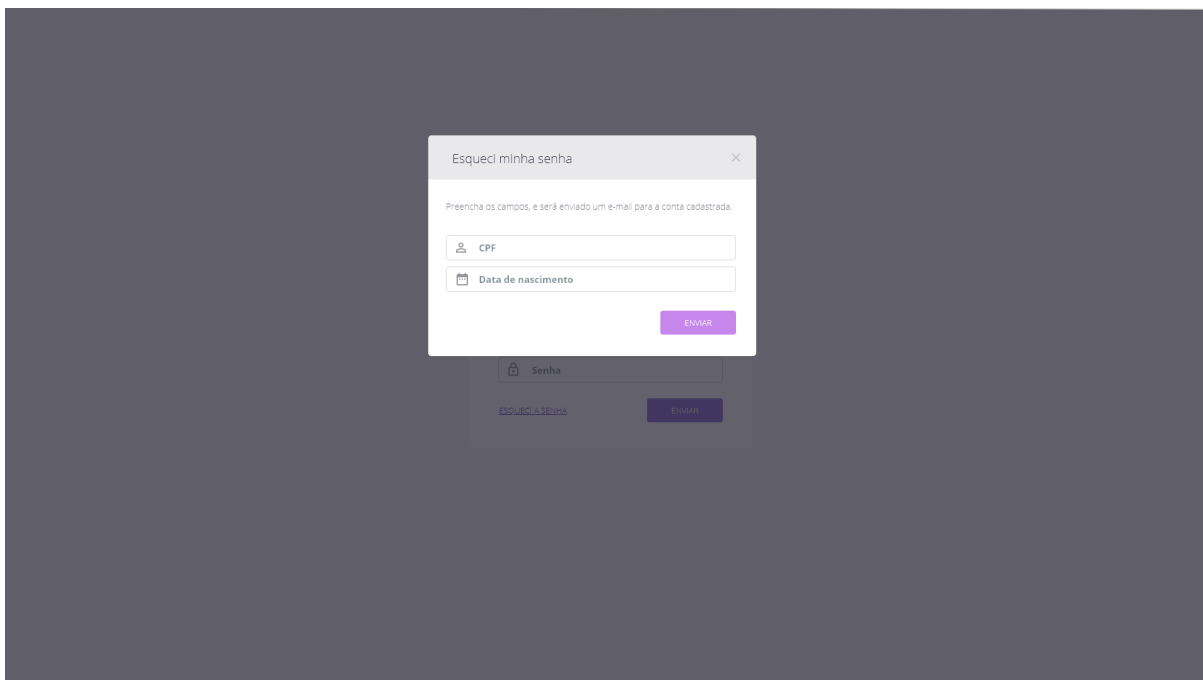
Fluxos Alternativos: Não se aplica.

Fluxos de Exceção: Não se aplica.

Regras de Negócio: Gerentes cliente só podem alterar o perfil de usuários cliente e gerentes suporte só podem alterar o perfil de usuários suporte.

UC031 - Recuperar Senha

Figura B.38: TELA DE ESQUECIMENTO DA SENHA



FONTE: OS AUTORES (2019)

Descrição: O ator pode recuperar sua senha de acesso.

Atores: Gerente Cliente, cliente, gerente suporte, suporte e admin.

Pré-Condição: O ator deve estar cadastrado no sistema.

Pós-Condição: A senha do ator é alterada.

Fluxo de Eventos Principal:

- P1.** O ator seleciona a opção de recuperar senha;
- P2.** O sistema apresenta uma tela com os campos CPF e data de nascimento;
- P3.** O ator preenche os campos;
- P4.** O ator clica em Recuperar (E1);
- P5.** O sistema apresenta uma tela com os campos nova senha e confirmação;
- P6.** O ator preenche os campos;
- P7.** O ator clica em Atualizar (E2, E3);
- P8.** O sistema atualiza a senha do ator;
- P9.** O caso de uso é encerrado;

Fluxos Alternativos: Não se aplica.

Fluxos de Exceção:

E1: CPF e data de nascimento errados:

1. O sistema retorna uma mensagem de erro informando que o CPF e/ou data de nascimento estão errados;
2. O caso de uso é encerrado;

E2: Confirmação da senha e senha nova diferentes:

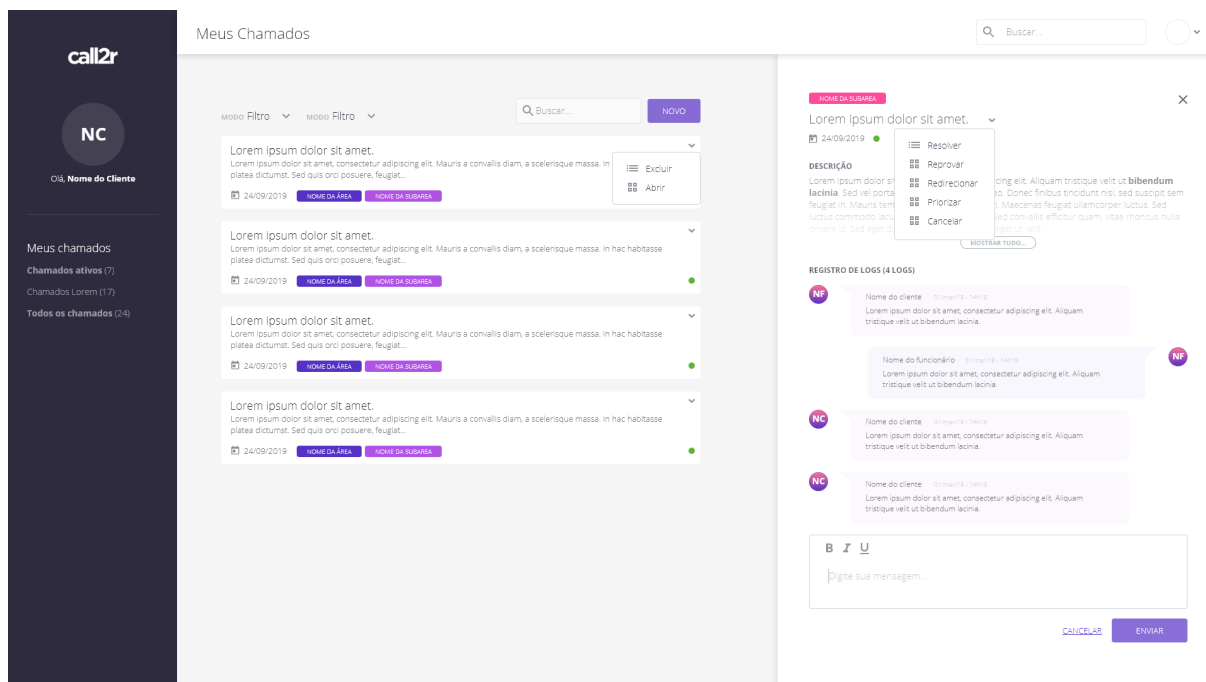
1. O sistema retorna uma mensagem de erro informando que a nova senha e a confirmação estão diferentes;
2. O caso de uso é encerrado;

E3: Nova senha igual a anterior:

1. O sistema retorna uma mensagem de erro informando que a nova senha e a anterior são iguais;
2. O caso de uso é encerrado;

UC032 - Adicionar log

Figura B.39: TELA DE ADIÇÃO DE LOG



FONTE: Os autores (2019)

Descrição: O ator pode adicionar logs aos chamados, acrescentando novas informações.

Atores: Gerente Cliente, cliente, gerente suporte e suporte.

Pré-Condição: O ator deve estar cadastrado no sistema e deve existir um chamado válido.

Pós-Condição: O texto é adicionado aos logs.

Fluxo de Eventos Principal:

- P1.** O ator visualiza um chamado;
- P2.** O ator preenche o campo de texto do log;
- P3.** O ator clica em uma opção (A1, A2, A3) (E1);
- P4.** O sistema adiciona o texto aos logs;
- P5.** O caso de uso é encerrado;

Fluxos Alternativos:

A1: Adicionar log:

1. O ator clica em Adicionar;
2. O sistema adiciona o texto aos logs;

3. O caso de uso é encerrado;

A2: Encaminhar para Cliente:

1. O ator clica em Encaminhar para Cliente;
2. O sistema adiciona o texto aos logs;
3. O sistema altera o estado do chamado para Aguardando resposta;
4. O sistema altera o responsável para o cliente solicitante;
5. O caso de uso é encerrado;

A3: Responder Suporte:

1. O ator clica em Responder Suporte;
2. O sistema adiciona o texto aos logs;
3. O sistema altera o estado do chamado para Em atendimento;
4. O sistema altera o responsável para o suporte responsável anterior;
5. O caso de uso é encerrado;

Fluxos de Exceção:

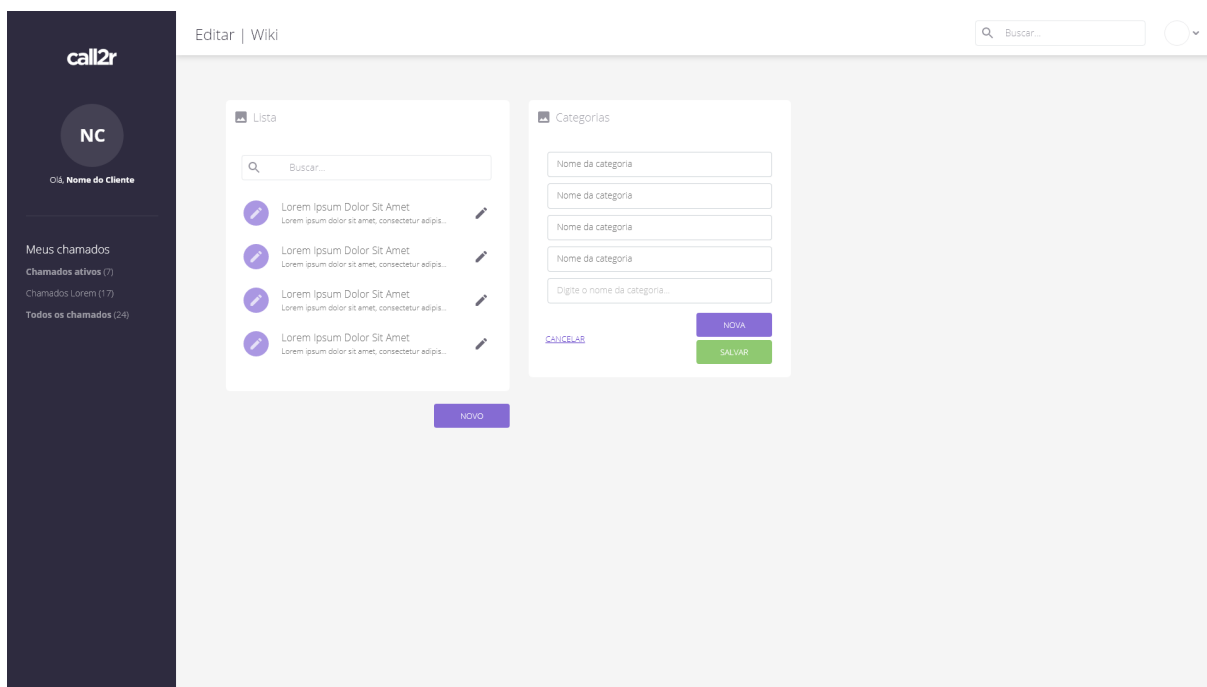
E1: Campo texto vazio:

1. O sistema retorna uma mensagem de erro informando que não é possível adicionar log vazio;
2. O caso de uso é encerrado;

Regras de Negócio: O fluxo alternativo A2 só é possível ser realizado pelos atores Suporte e Gerente Suporte com o chamado no estado Em atendimento. O fluxo alternativo A3 só é possível ser realizado pelos atores Cliente e Gerente Cliente com o chamado no estado Aguardando Resposta.

UC033 - Manter Categoria

Figura B.40: TELA DE EDIÇÃO DE CATEGORIAS



FONTE: Os autores (2019)

Descrição: O ator pode criar, visualizar, alterar ou excluir categorias da wiki.

Atores: Gerente suporte e suporte.

Pré-Condição: O ator deve estar logado como gerente suporte ou suporte.

Pós-Condição: O ator poderá ter criado, alterado, excluído ou visualizado categorias da wiki.

Fluxo de Eventos Principal:

- P1.** O ator acessa a tela de wiki;
- P2.** O sistema carrega uma lista de categorias;
- P3.** O ator pode visualizar as categorias (A1, A2, A3, A4, A5);
- P4.** O caso de uso é encerrado;

Fluxos Alternativos:

A1: Criar categoria:

- 1. O ator clica em Novo;
- 2. O sistema apresenta uma tela de cadastro de categoria;
- 3. O ator preenche os dados da categoria;
- 4. O ator clica em Salvar (E1, E2);

5. O sistema cadastra a categoria;
6. O caso de uso é encerrado;

A2: Visualizar categoria:

1. O ator clica em uma categoria na lista de subáreas;
2. O sistema apresenta uma tela com os dados da categoria selecionado;
3. O ator pode visualizar os dados;
4. O caso de uso é encerrado;

A3: Alterar categoria:

1. O ator clica em Editar em uma categoria existente;
2. O sistema apresenta uma tela de alteração dos dados da categoria;
3. O ator pode alterar os dados da categoria;
4. O ator clica em Salvar (E1, E2);
5. O sistema apresenta uma tela de confirmação das alterações;
6. O ator confirma;
7. O sistema salva as alterações;
8. O caso de uso é encerrado;

A4: Excluir categoria:

1. O ator clica em Excluir em uma categoria existente;
2. O sistema apresenta uma tela de confirmação da exclusão;
3. O ator confirma a exclusão;
4. O sistema exclui a categoria;
5. O caso de uso é encerrado;

Fluxos de Exceção:**E1:** Categoria já existe:

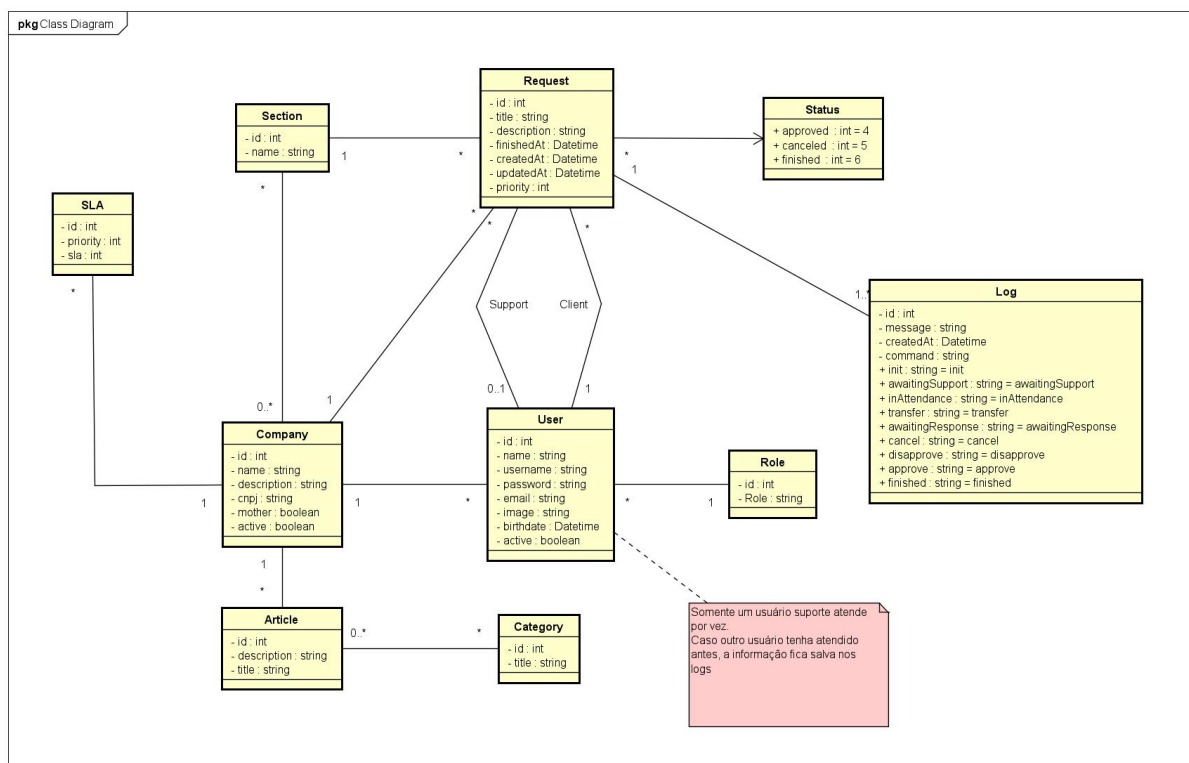
1. O sistema retorna uma mensagem de erro informando que a categoria já existe;
2. O caso de uso é encerrado;

E2: Categoria ou campos vazios:

1. O sistema retorna uma mensagem de erro informando que existem campos vazios;
2. O caso de uso é encerrado;

APÊNDICE C – DIAGRAMA DE CLASSES

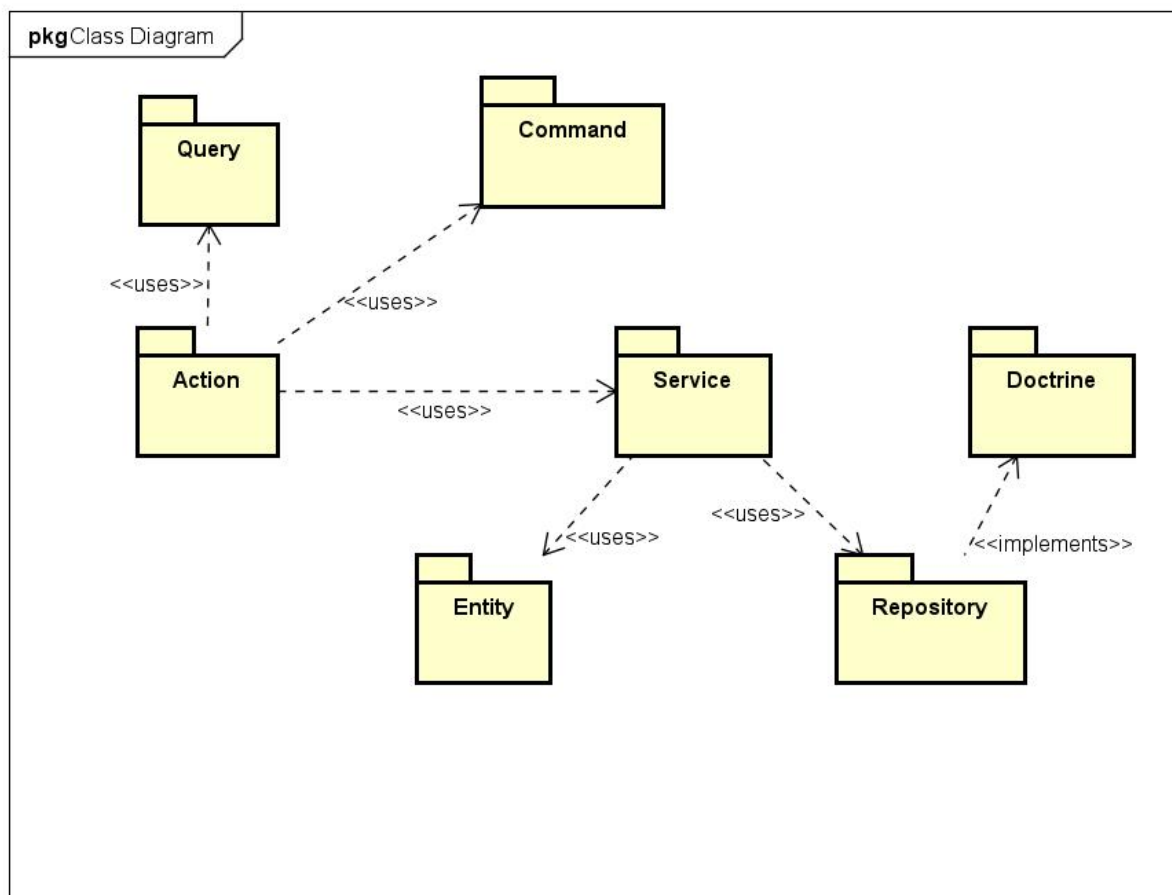
Figura C.1: DIAGRAMA DE CLASSES



FONTE: Os autores (2019)

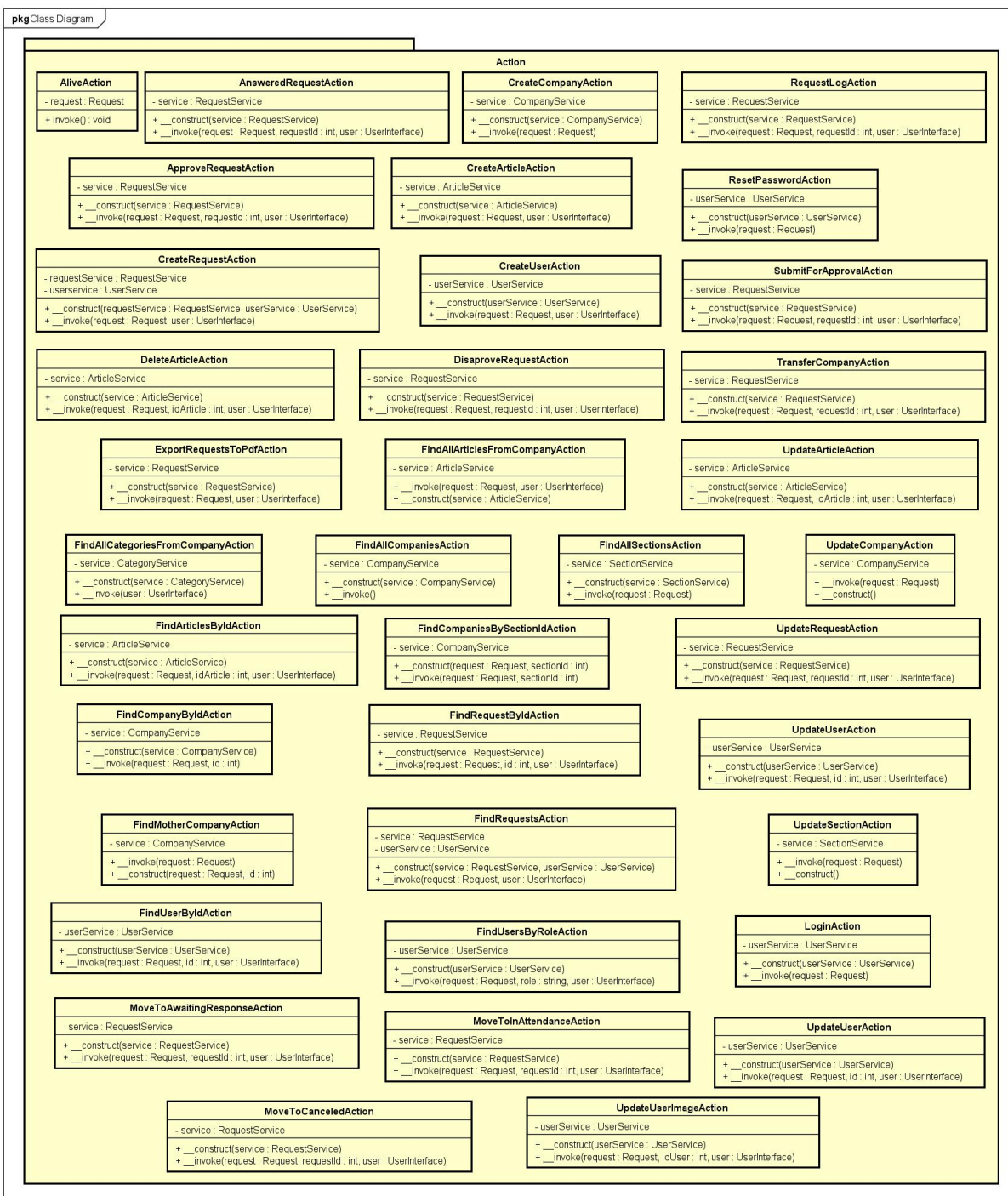
APÊNDICE D – DIAGRAMAS DE CLASSES DE IMPLEMENTAÇÃO

Figura D.1: CLASSES DE IMPLEMENTAÇÃO



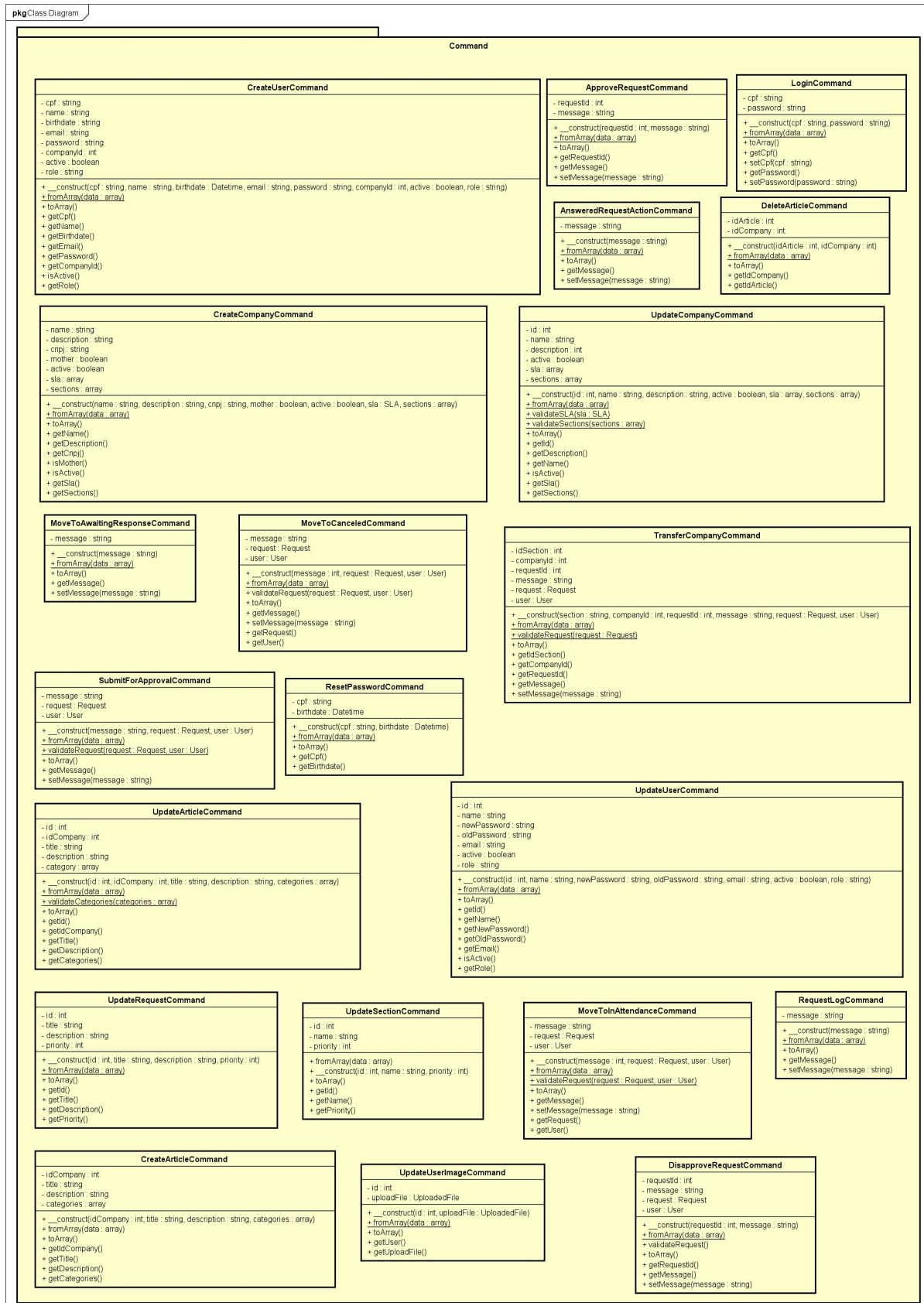
FONTE: Os autores (2020)

Figura D.2: CLASSES DE IMPLEMENTAÇÃO DE AÇÕES



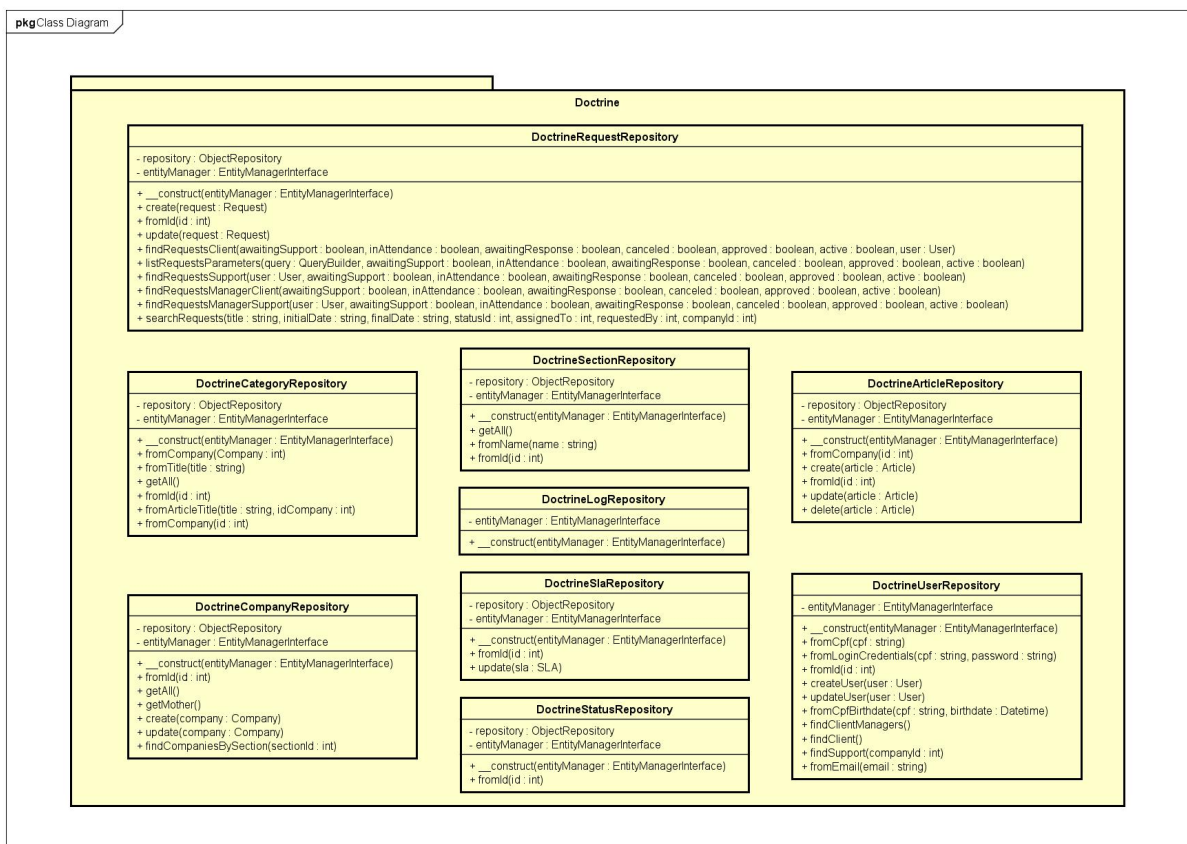
FONTE: Os autores (2020)

Figura D.3: CLASSES DE IMPLEMENTAÇÃO DE COMANDOS



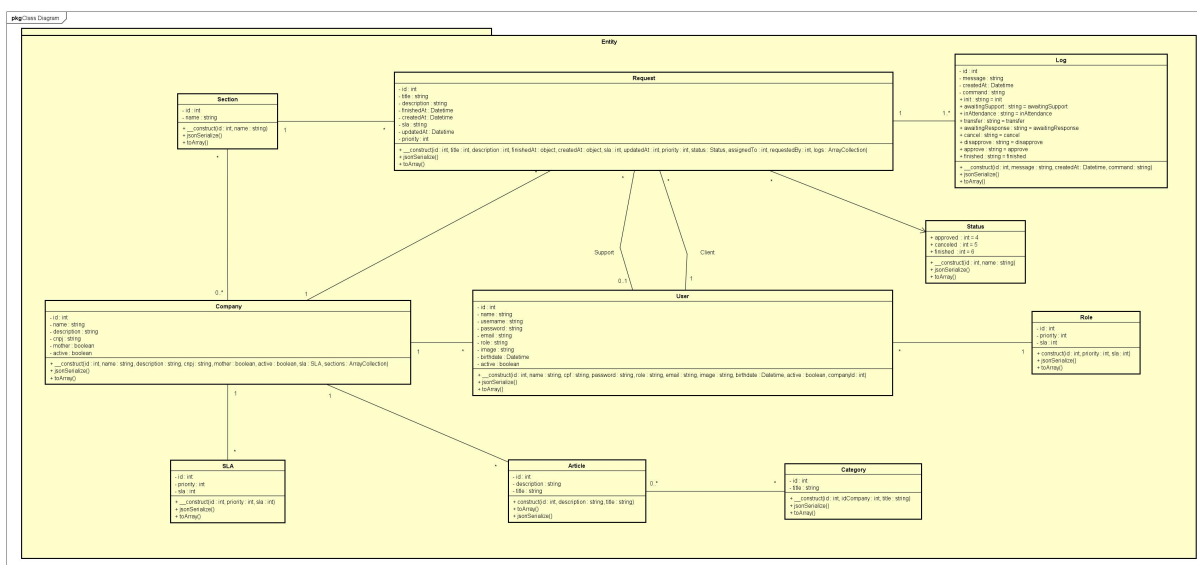
FONTE: Os autores (2020)

Figura D.4: CLASSES DE IMPLEMENTAÇÃO DOCTRINE



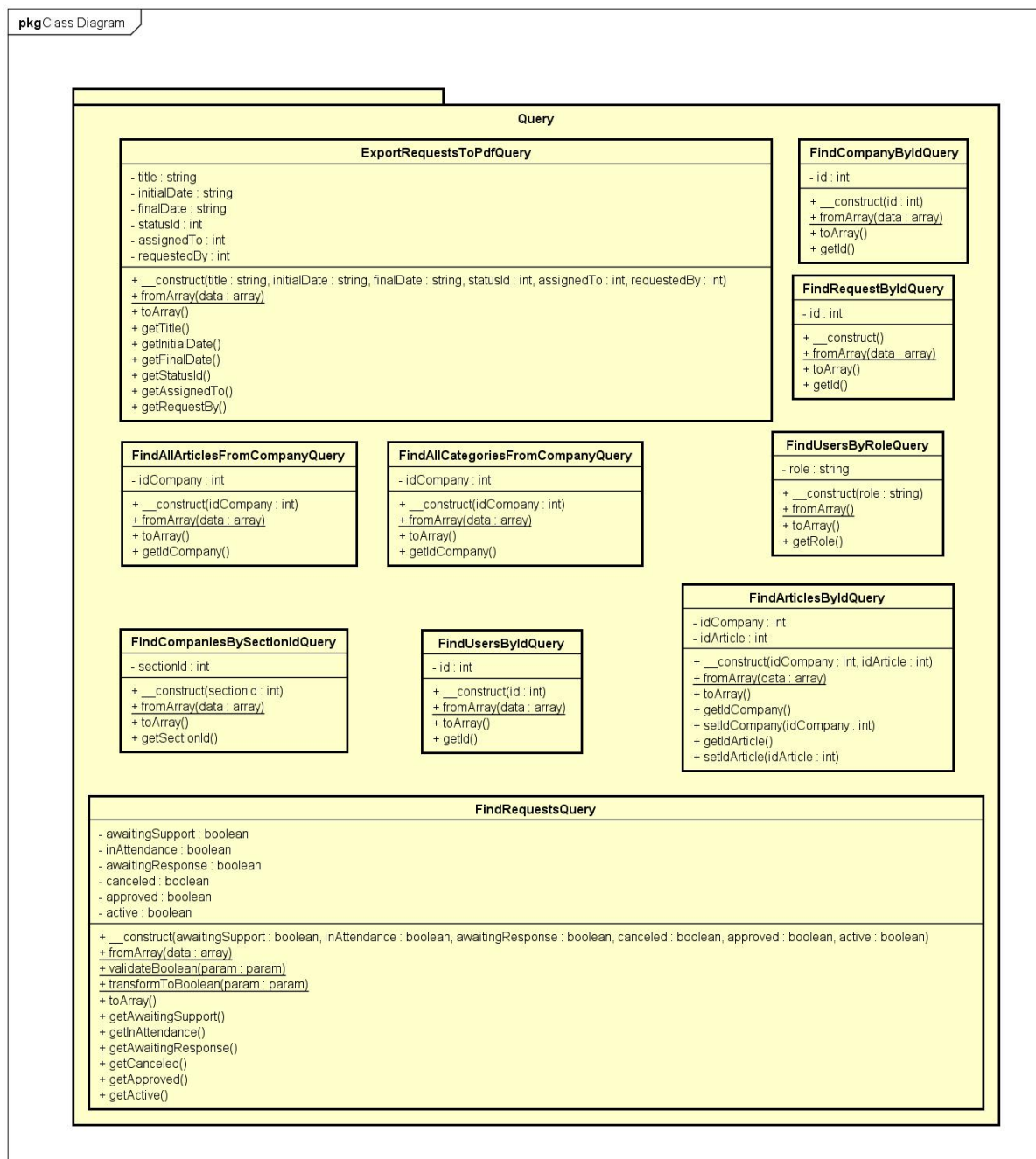
FONTE: Os autores (2020)

Figura D.5: CLASSES DE IMPLEMENTAÇÃO DE ENTIDADES



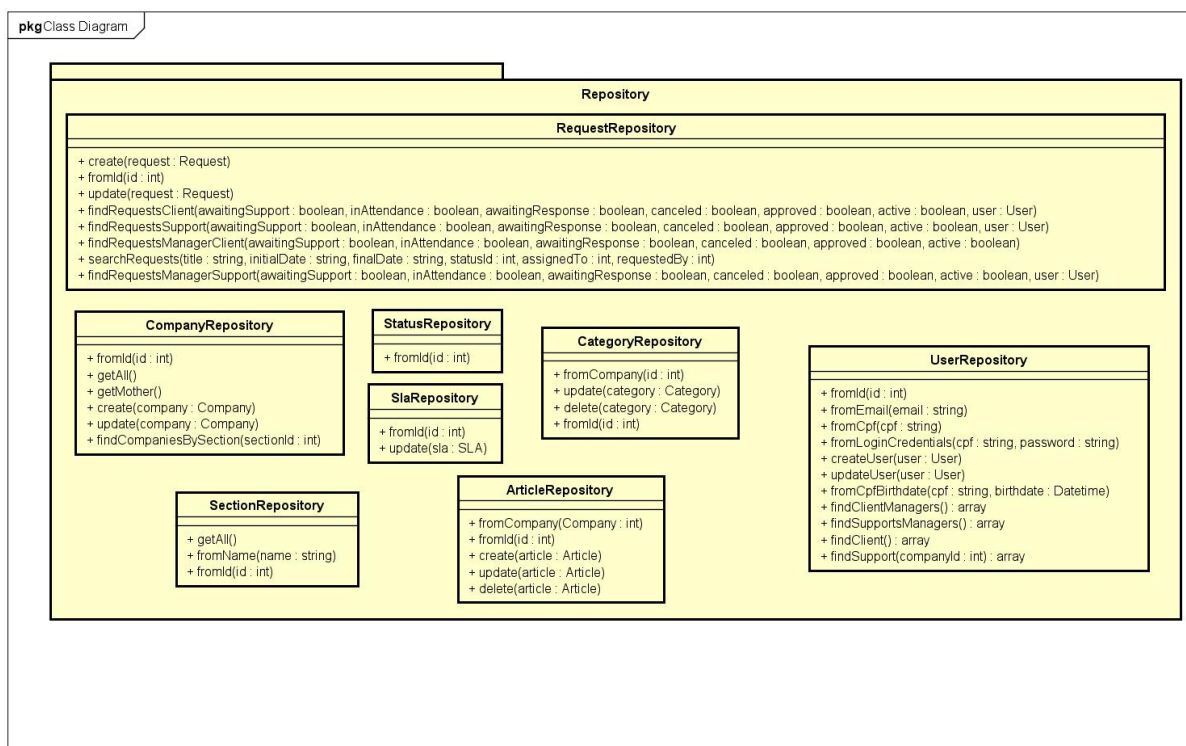
FONTE: Os autores (2020)

Figura D.6: CLASSES DE IMPLEMENTAÇÃO DE PESQUISAS



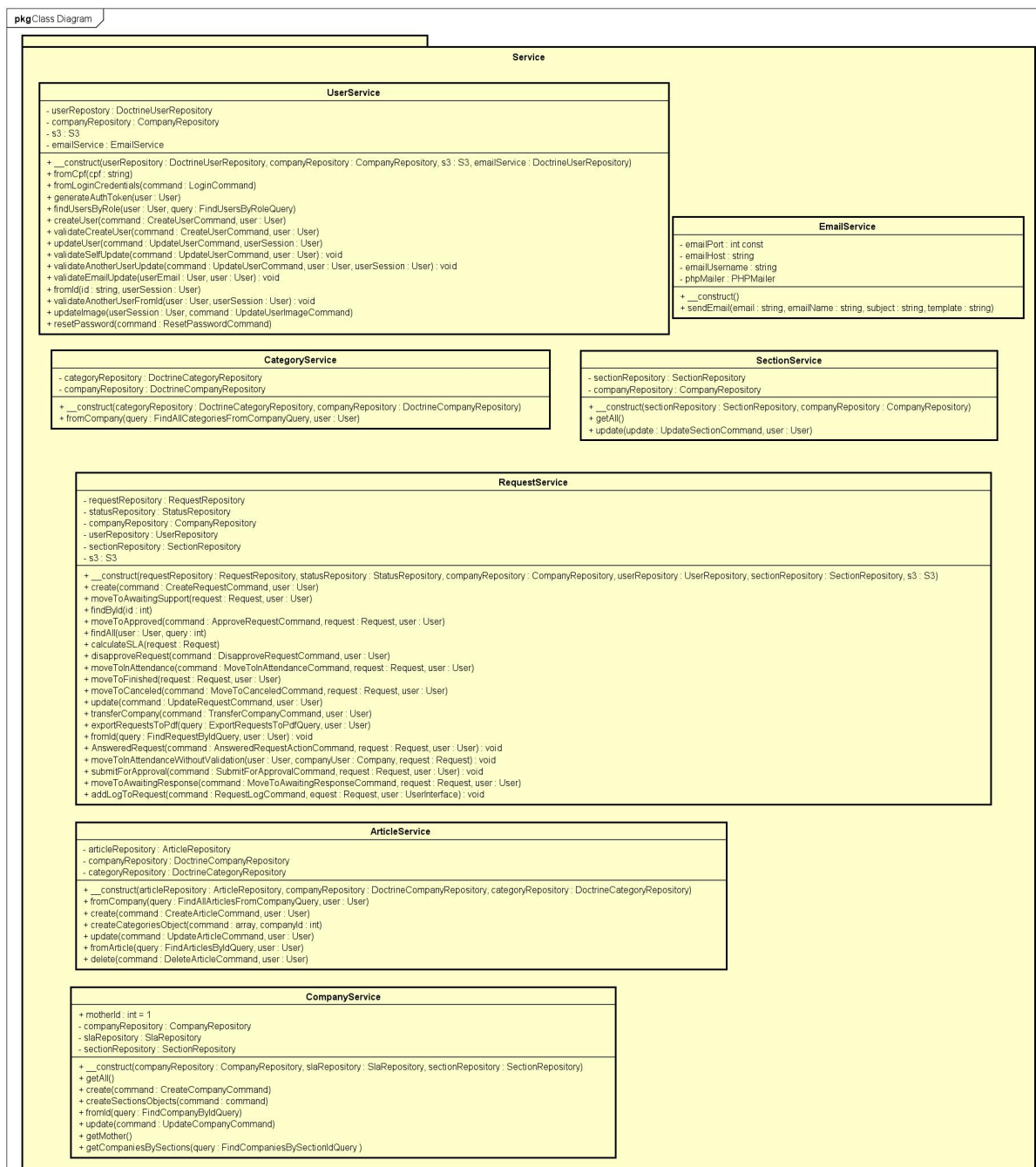
FONTE: Os autores (2020)

Figura D.7: CLASSES DE IMPLEMENTAÇÃO DE REPOSITÓRIOS



FONTE: Os autores (2020)

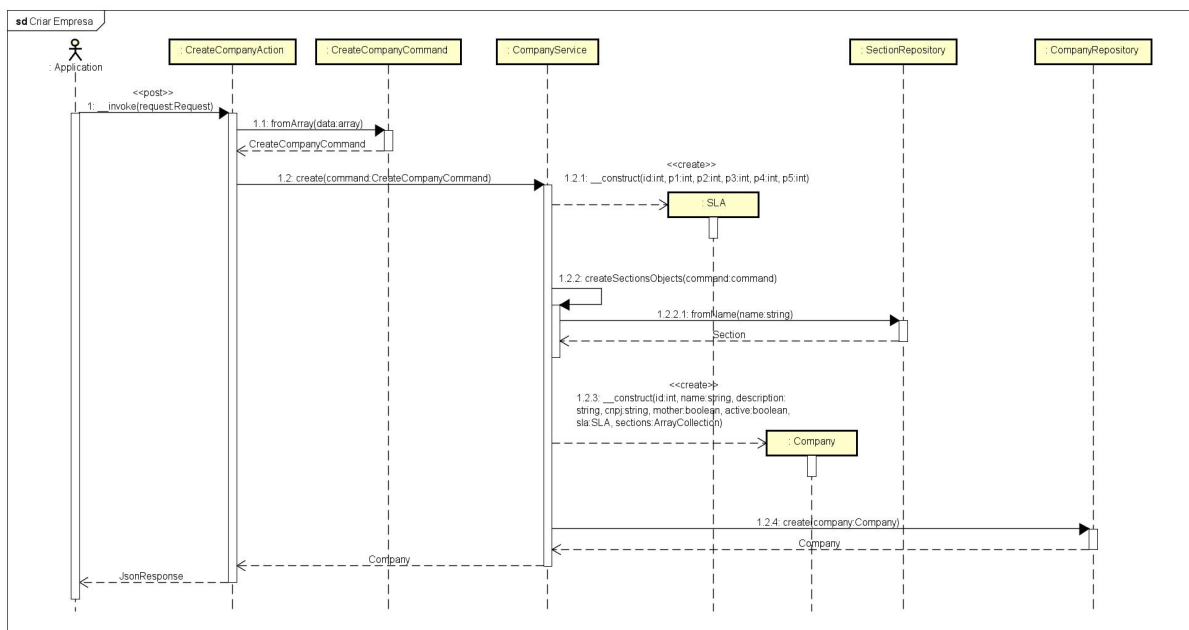
Figura D.8: CLASSES DE IMPLEMENTAÇÃO DE SERVIÇOS



FONTE: Os autores (2020)

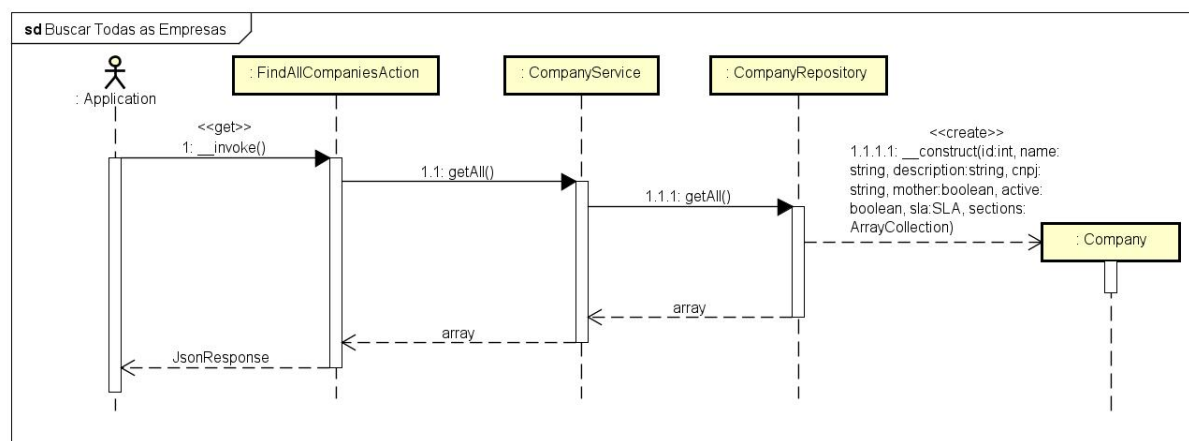
APÊNDICE E – DIAGRAMAS DE SEQUÊNCIA

Figura E.1: CRIAR EMPRESA



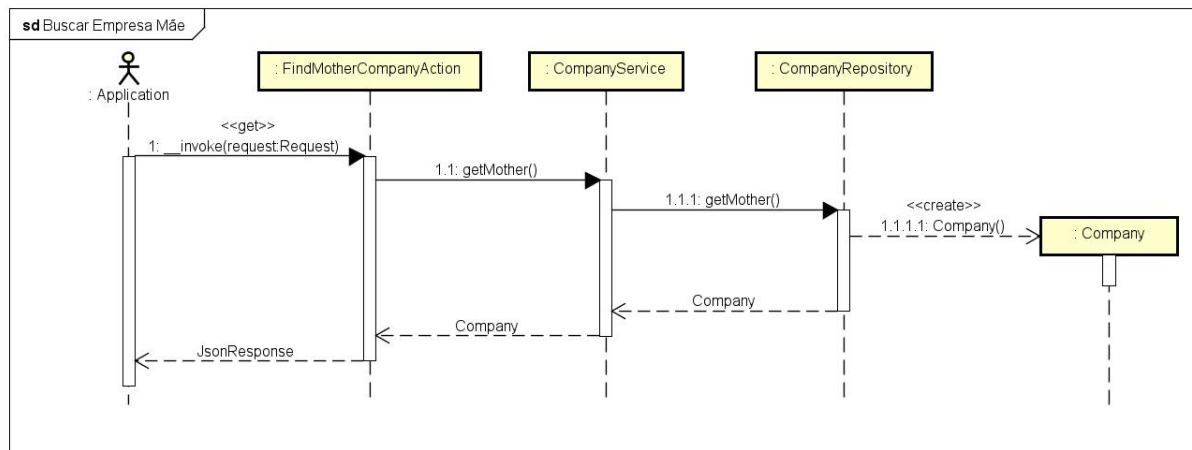
FONTE: Os autores (2020)

Figura E.2: BUSCAR TODAS AS EMPRESAS



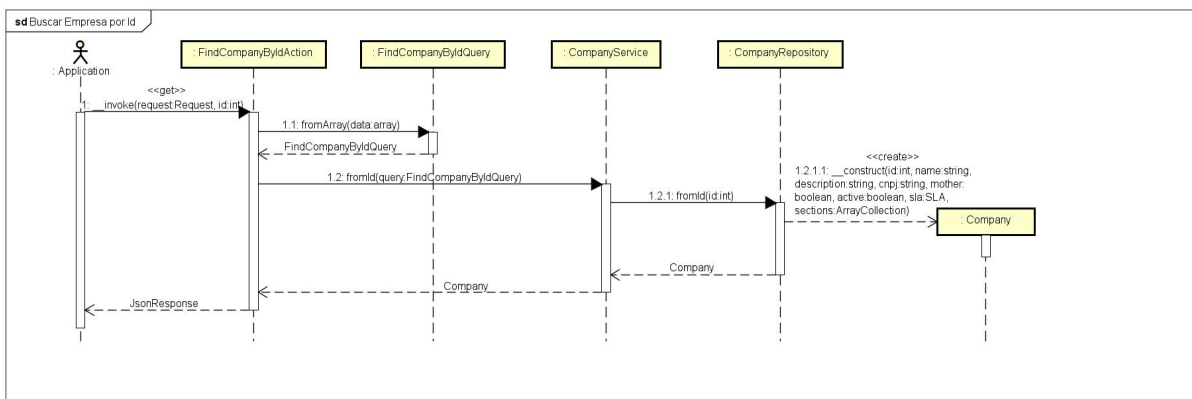
FONTE: Os autores (2020)

Figura E.3: BUSCAR EMPRESA MÃE



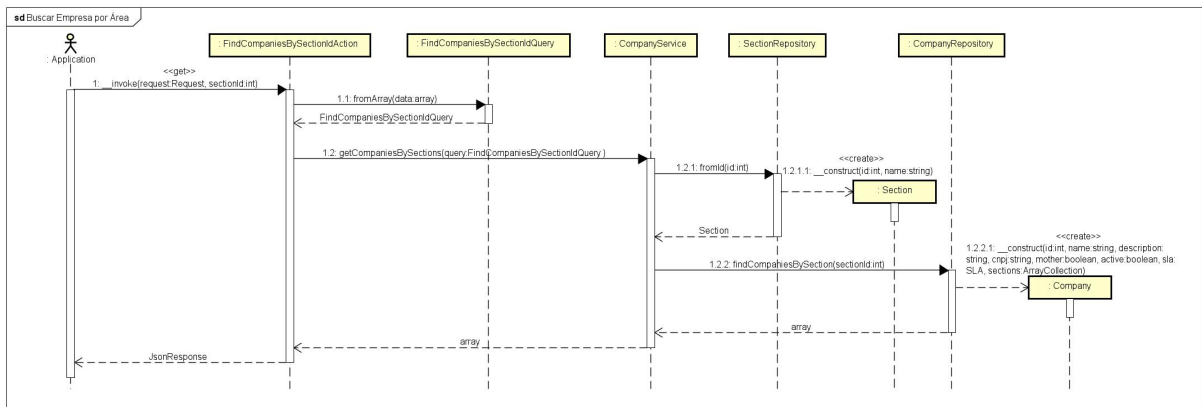
FONTE: Os autores (2020)

Figura E.4: BUSCAR EMPRESA POR ID



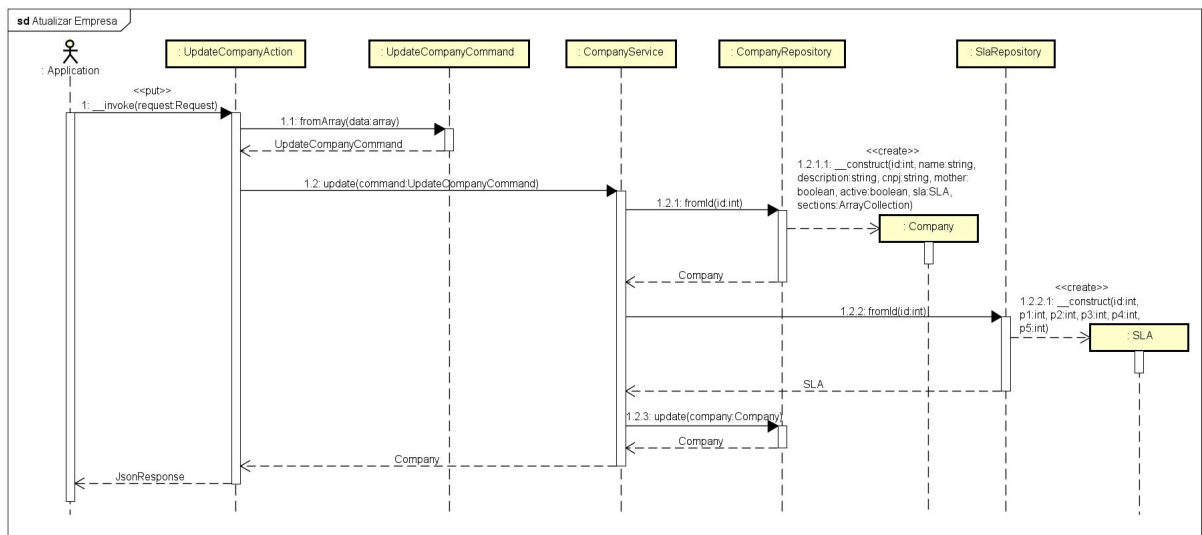
FONTE: Os autores (2020)

Figura E.5: BUSCAR EMPRESA POR ÁREA



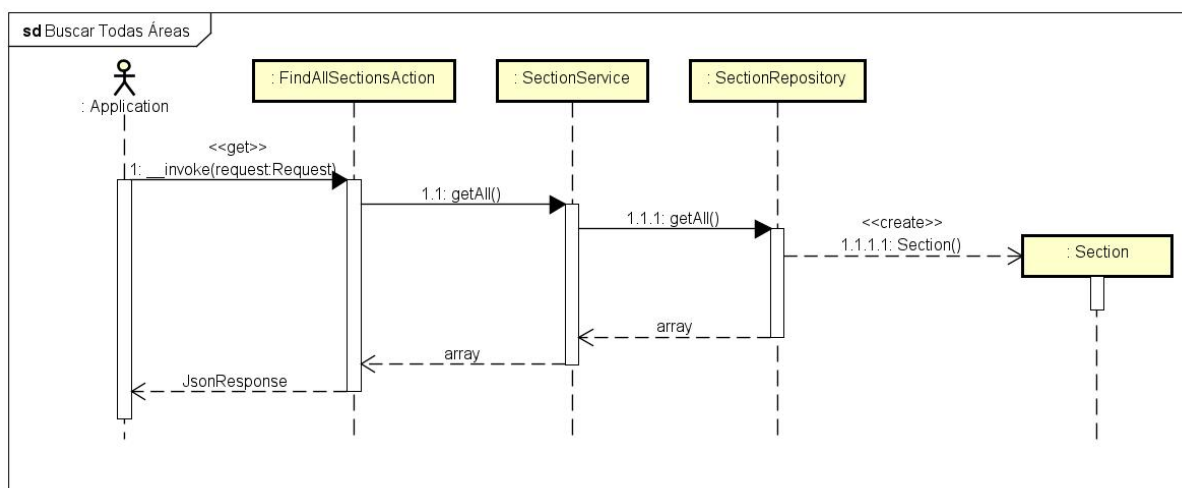
FONTE: Os autores (2020)

Figura E.6: ATUALIZAR EMPRESA



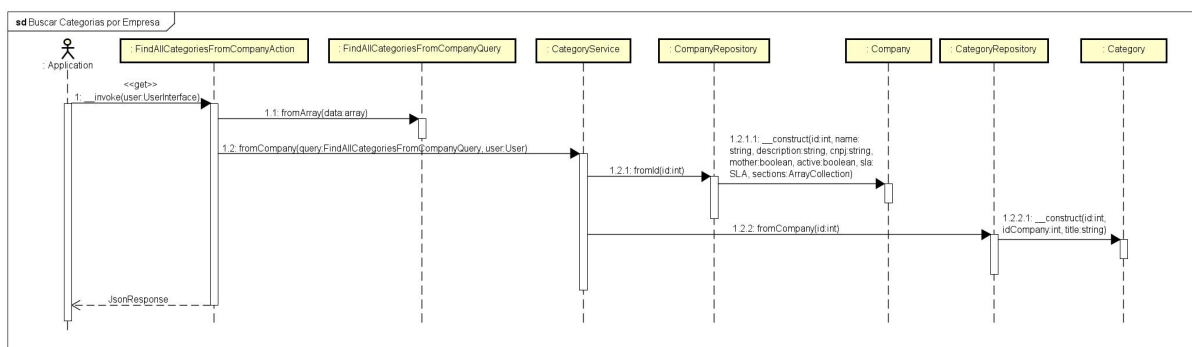
FONTE: Os autores (2020)

Figura E.7: BUSCAR TODAS AS ÁREAS



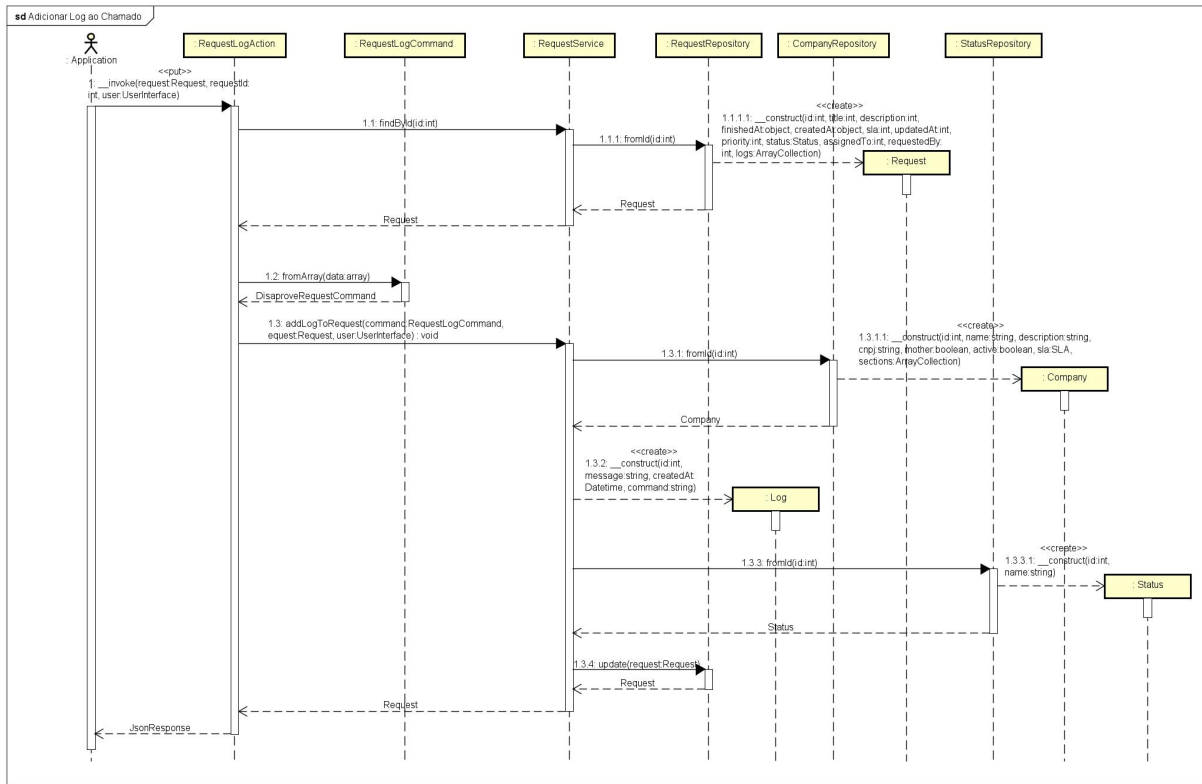
FONTE: Os autores (2020)

Figura E.8: BUSCAR CATEGORIAS POR EMPRESA



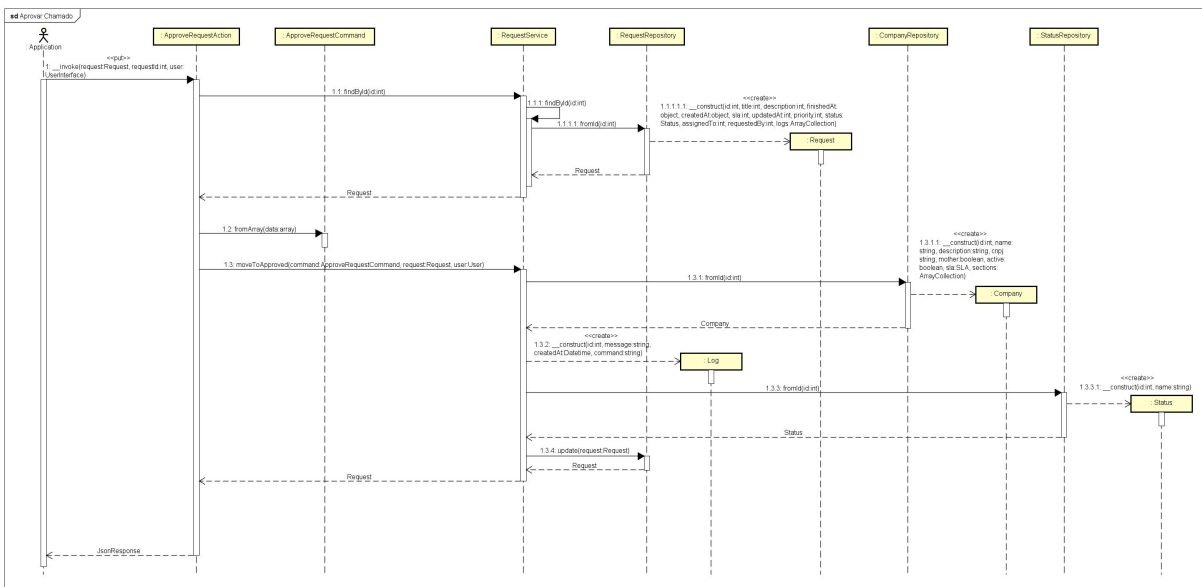
FONTE: Os autores (2020)

Figura E.9: ADICIONAR LOG AO CHAMADO



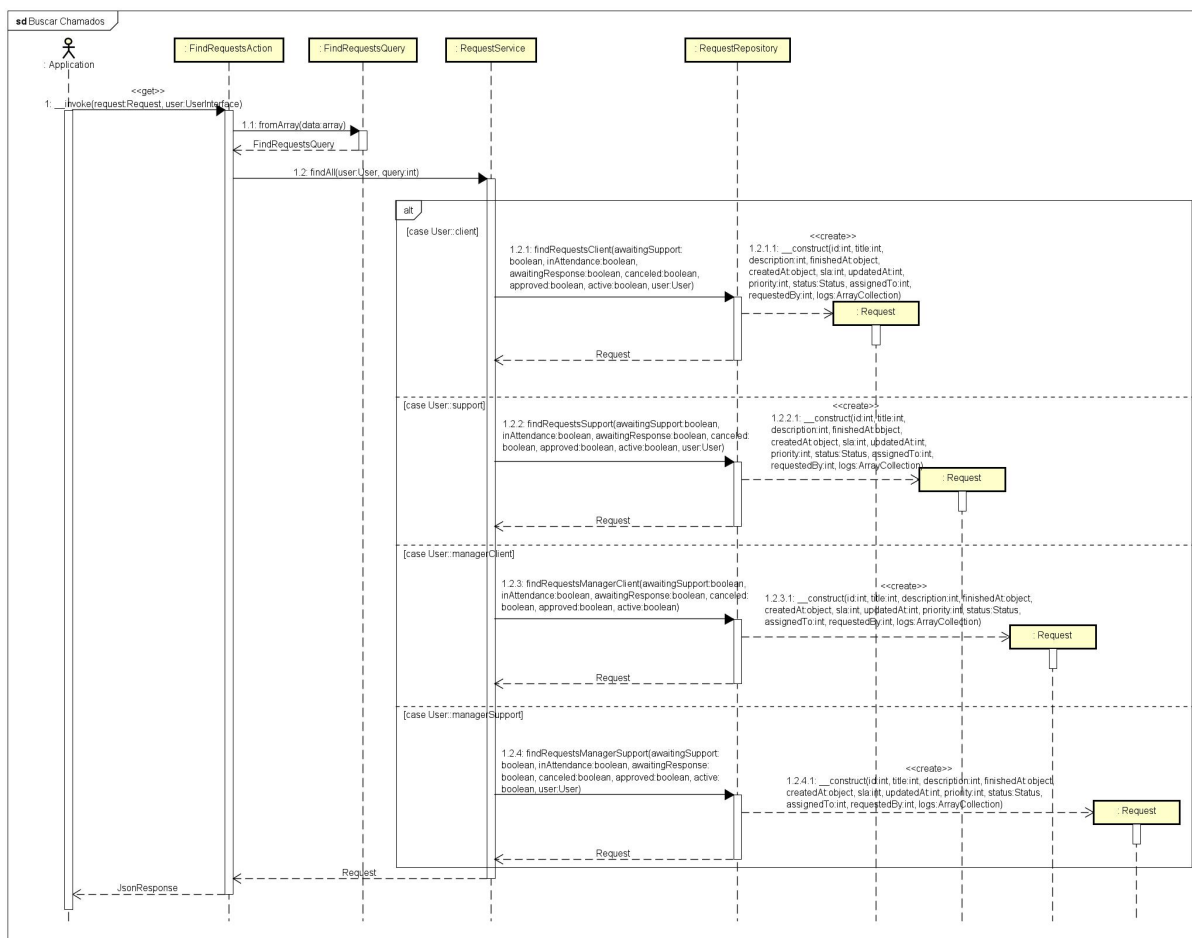
FONTE: Os autores (2020)

Figura E.10: APROVAR CHAMADO



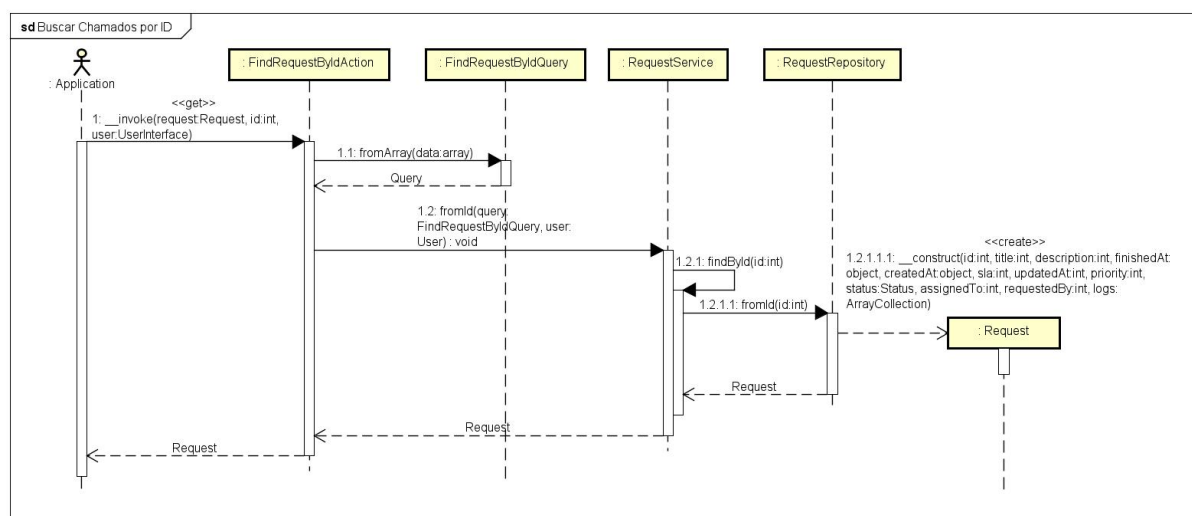
FONTE: Os autores (2020)

Figura E.11: BUSCAR CHAMADOS



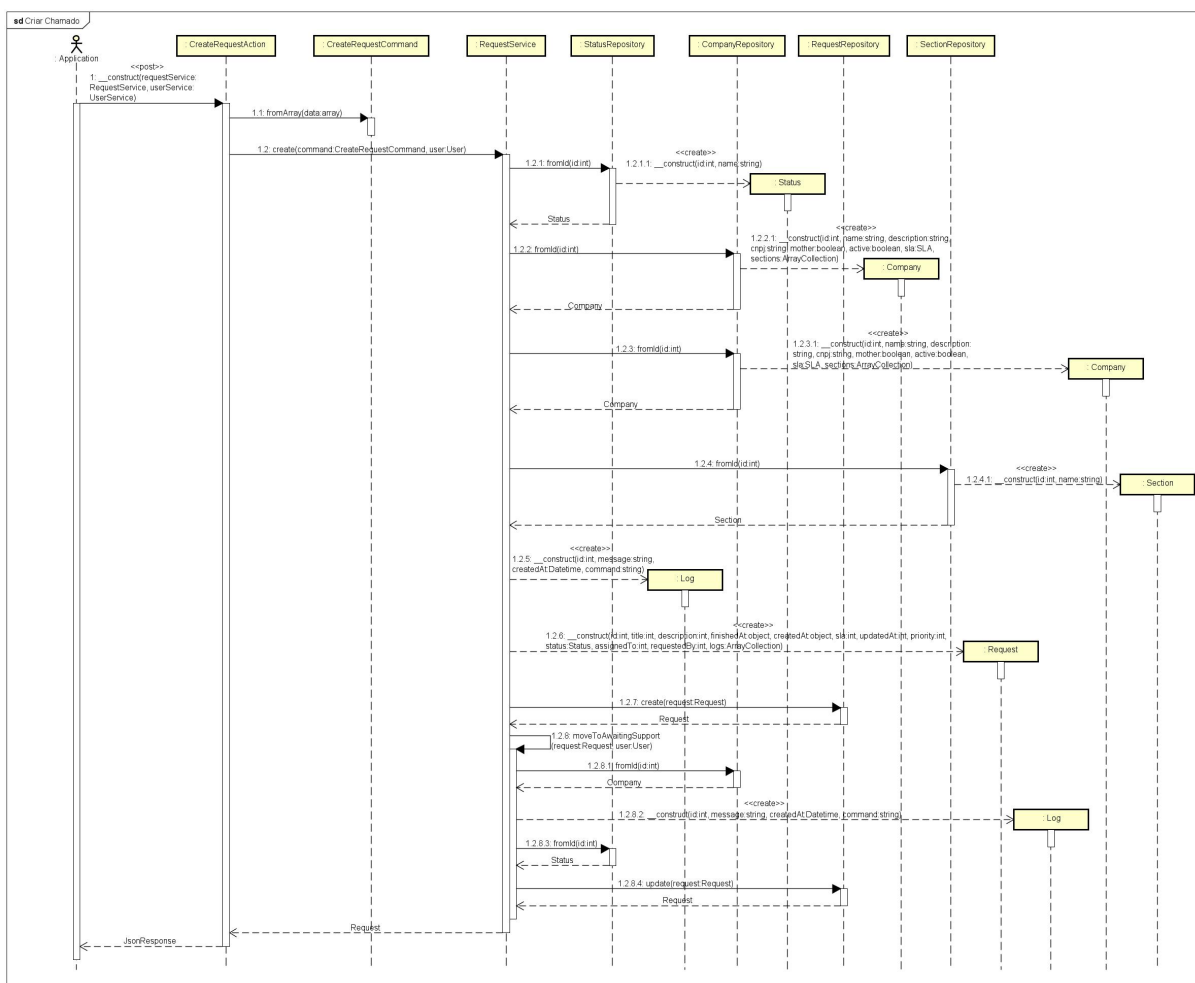
FONTE: Os autores (2020)

Figura E.12: BUSCAR CHAMADOS POR ID



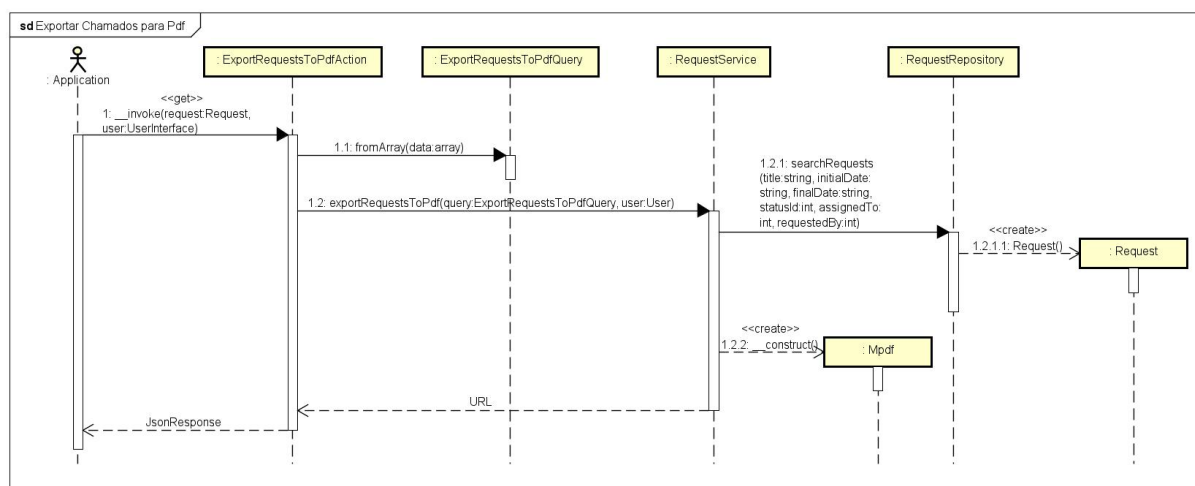
FONTE: Os autores (2020)

Figura E.13: CRIAR CHAMADO



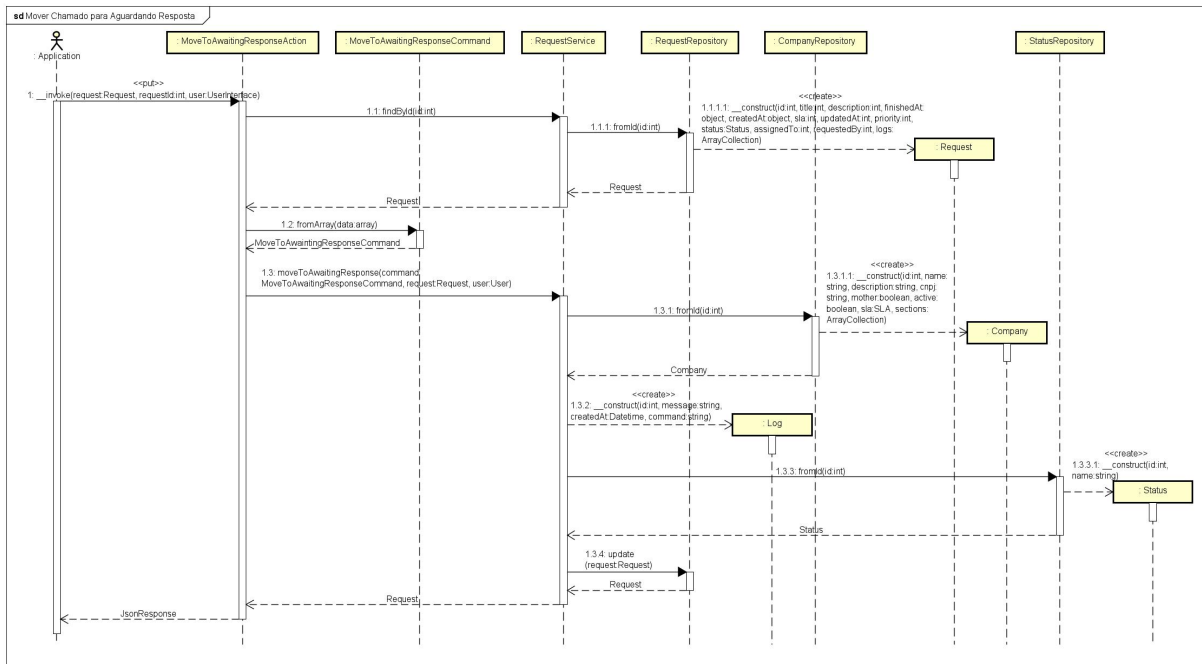
FONTE: Os autores (2020)

Figura E.14: EXPORTAR CHAMADOS PARA PDF



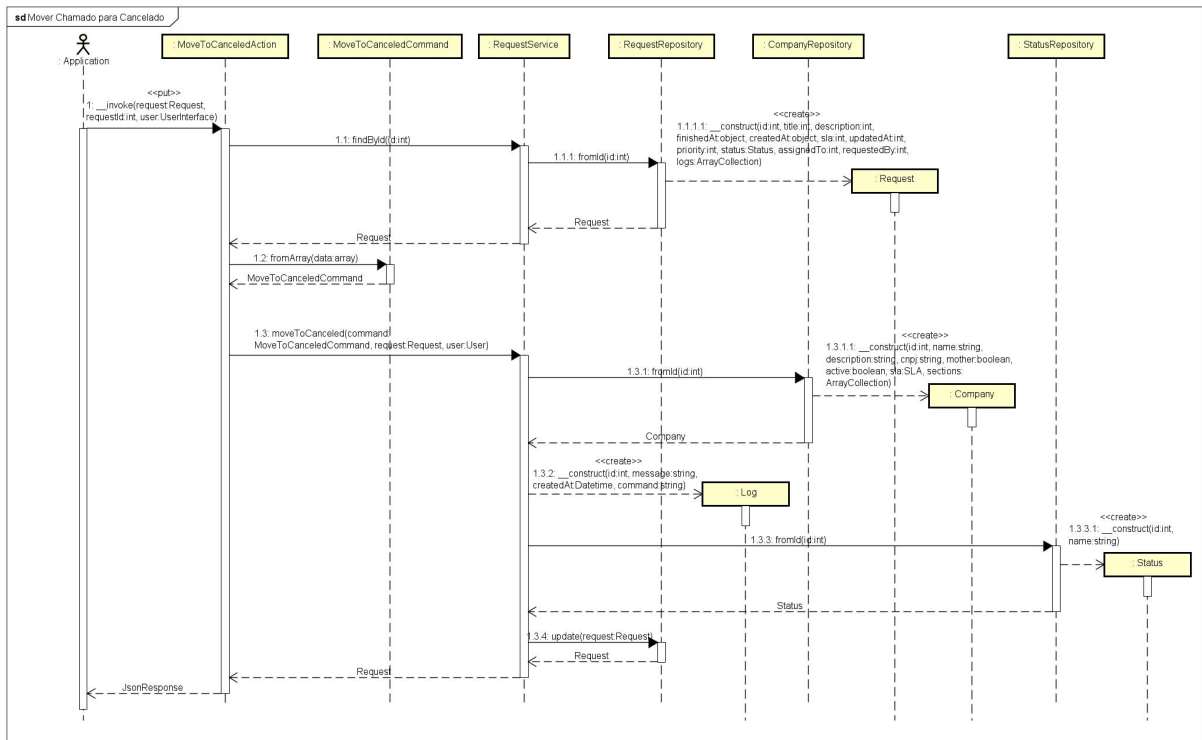
FONTE: Os autores (2020)

Figura E.15: MOVER CHAMADO PARA AGUARDANDO RESPOSTA



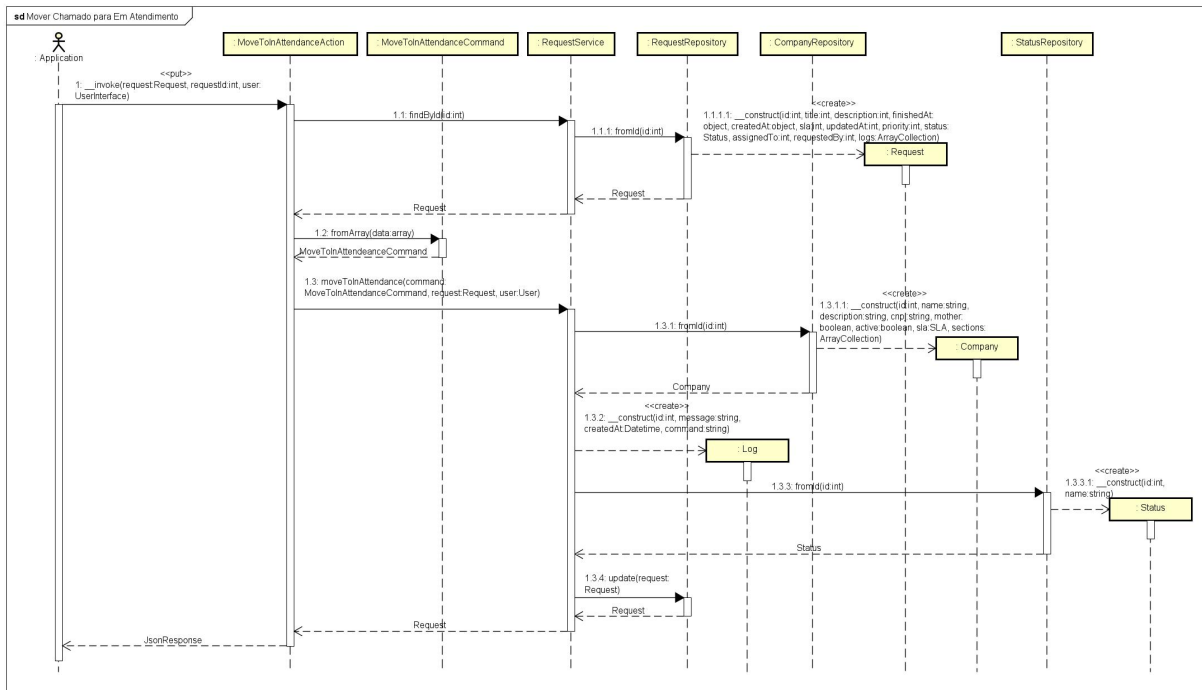
FONTE: Os autores (2020)

Figura E.16: MOVER CHAMADO PARA CANCELADO



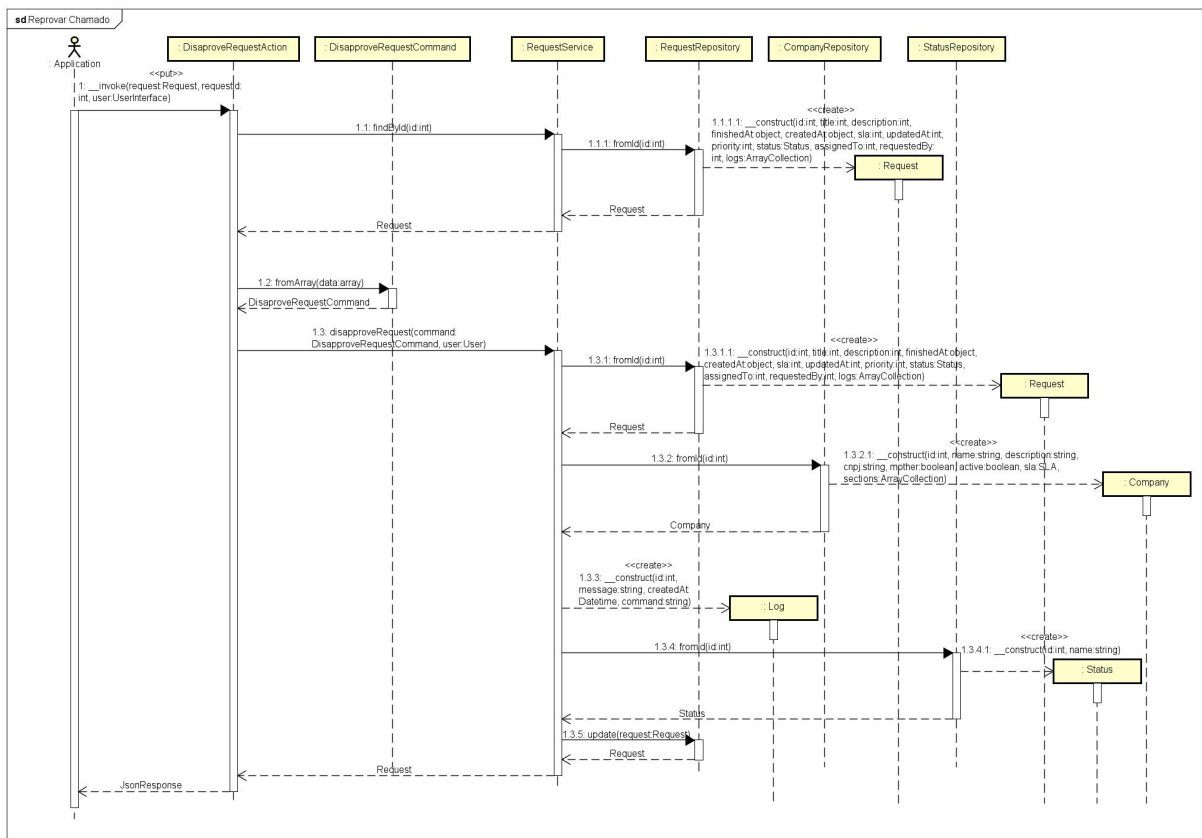
FONTE: Os autores (2020)

Figura E.17: MOVER CHAMADO PARA EM ATENDIMENTO



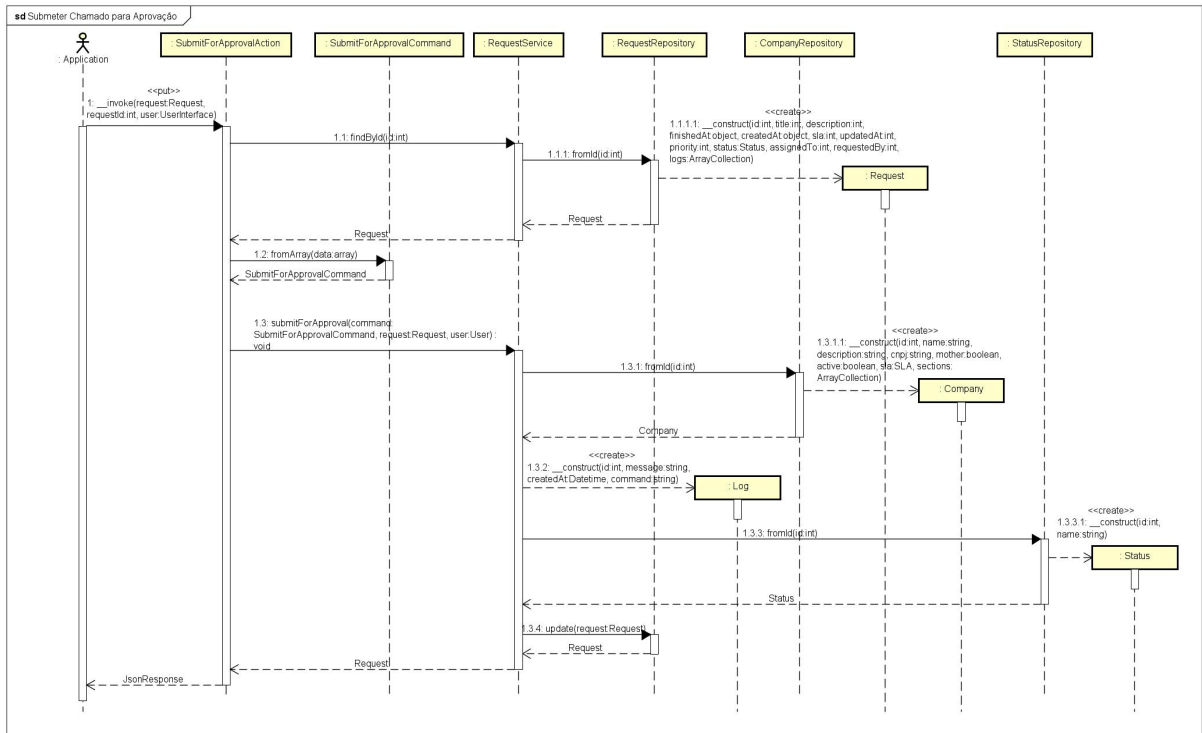
FONTE: Os autores (2020)

Figura E.18: REPROVAR CHAMADO



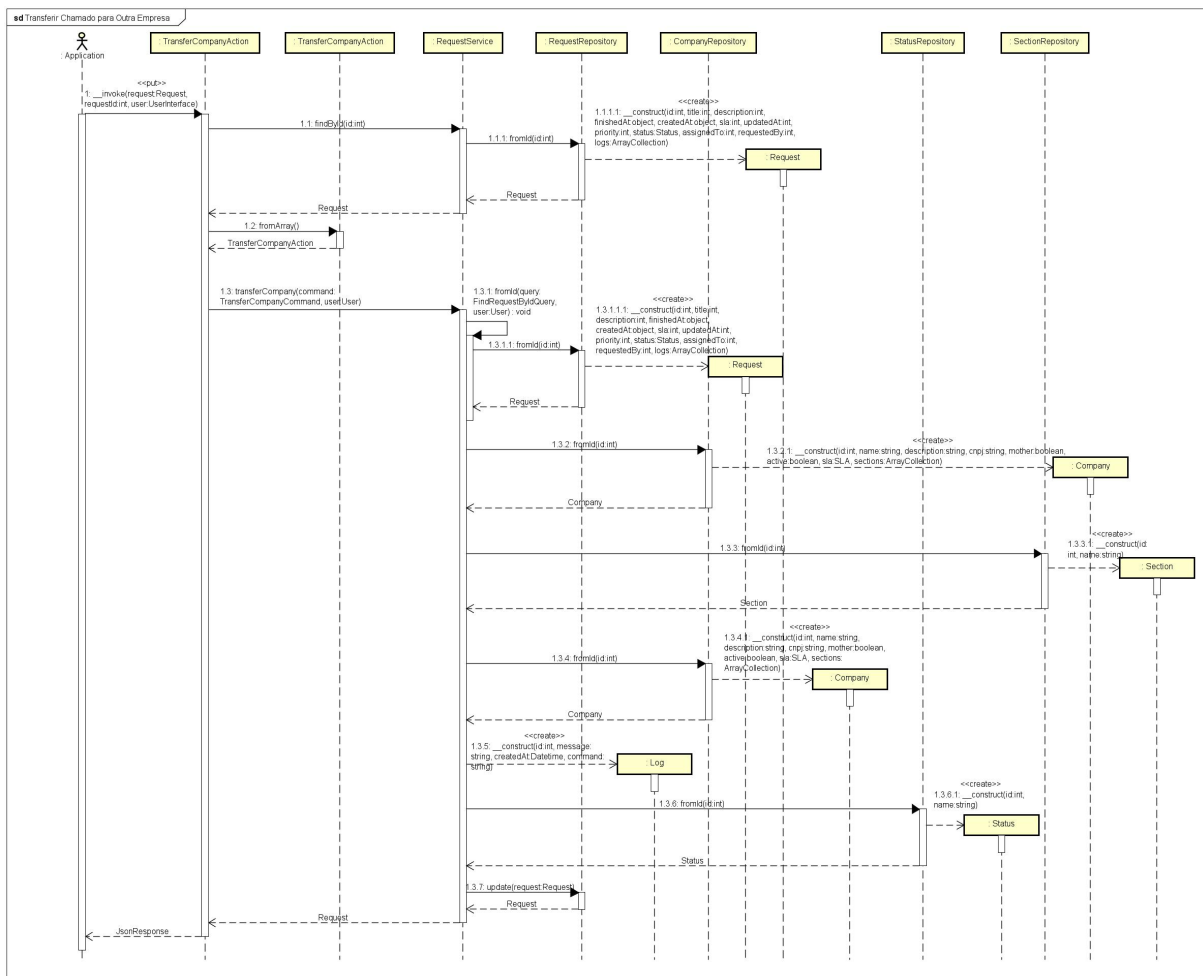
FONTE: Os autores (2020)

Figura E.19: SUBMETER CHAMADO PARA APROVAÇÃO



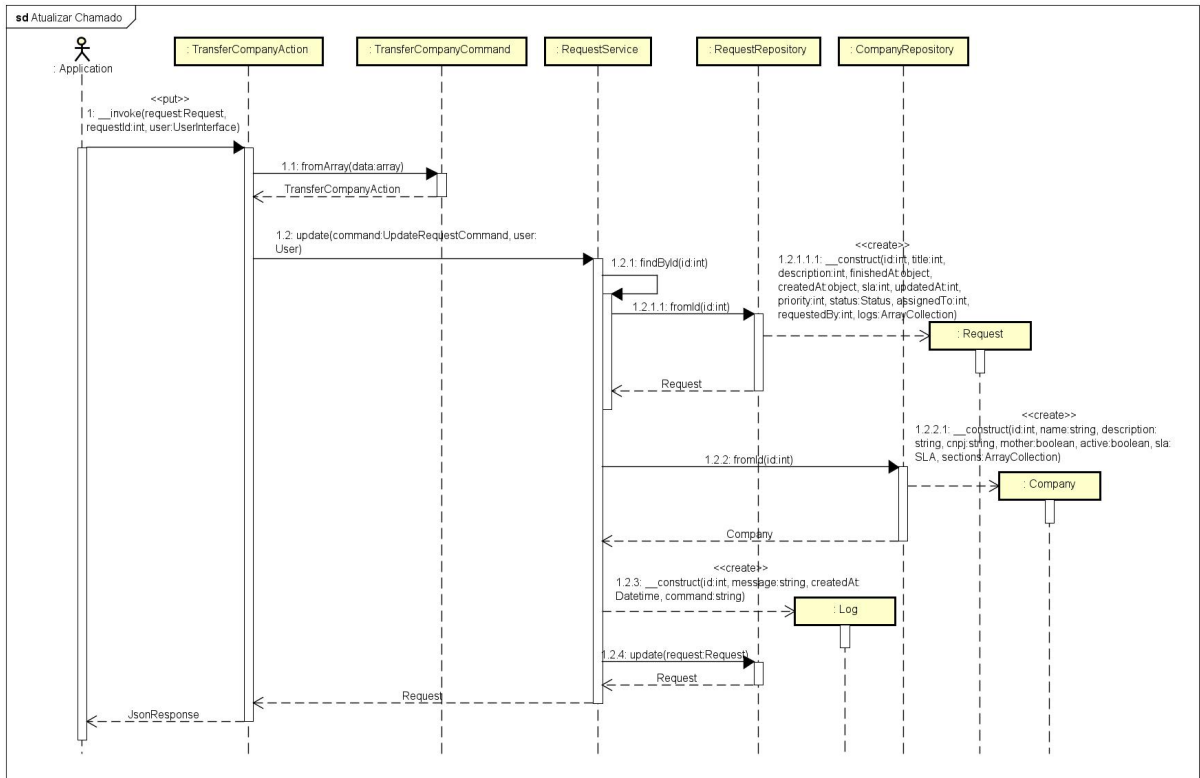
FONTE: Os autores (2020)

Figura E.20: TRANSFERIR CHAMADO PARA OUTRA EMPRESA



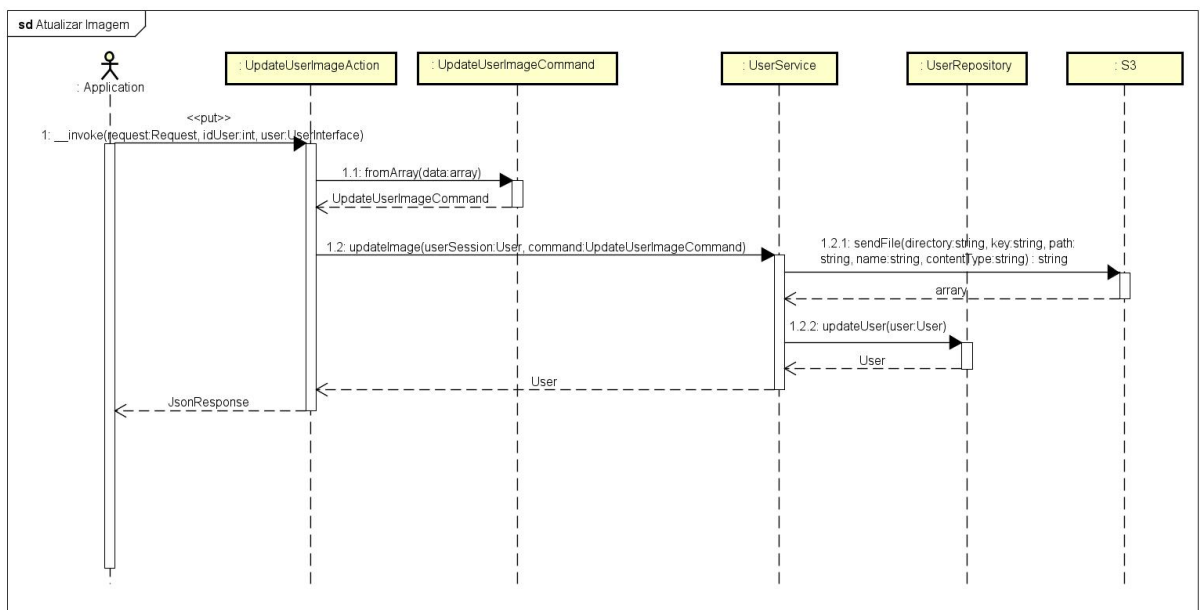
FONTE: Os autores (2020)

Figura E.21: ATUALIZAR CHAMADO



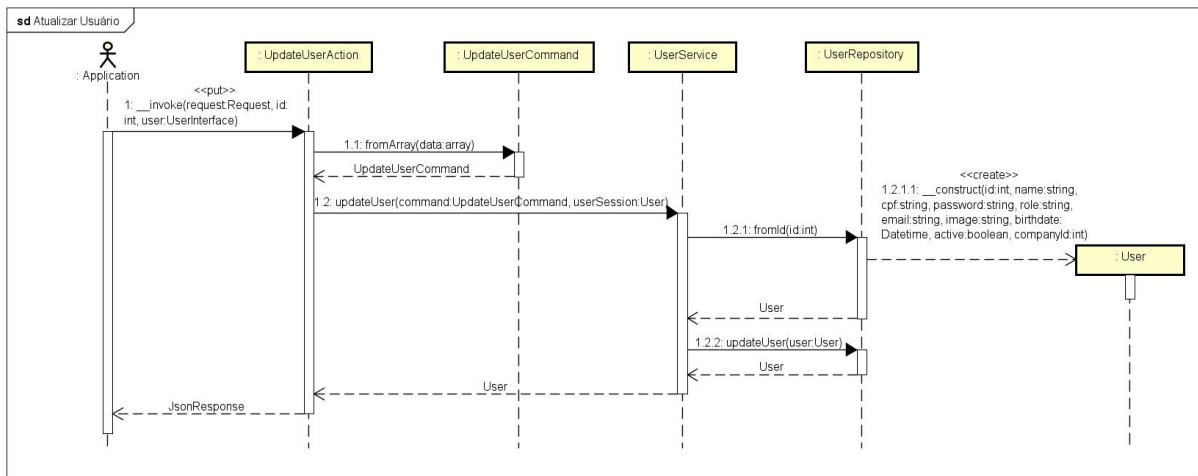
FONTE: Os autores (2020)

Figura E.22: ATUALIZAR IMAGEM



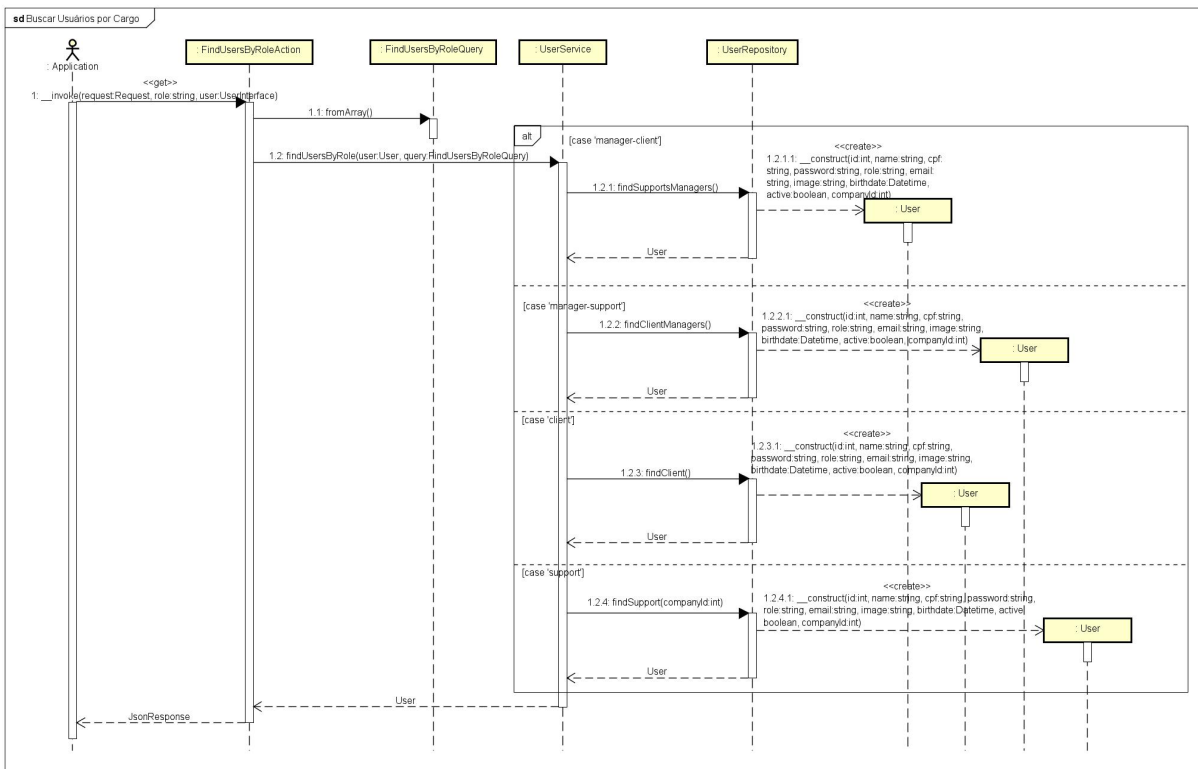
FONTE: Os autores (2020)

Figura E.23: ATUALIZAR USUÁRIO



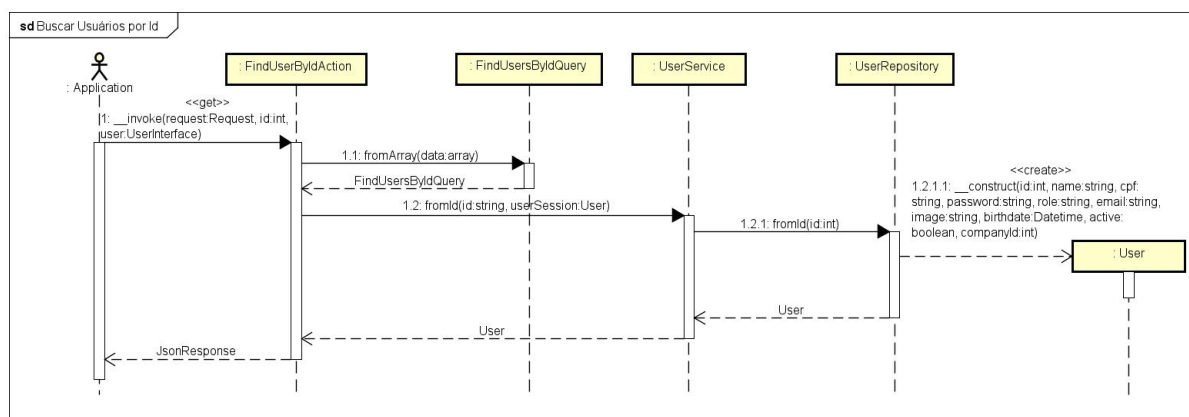
FONTE: Os autores (2020)

Figura E.24: BUSCAR USUÁRIOS POR CARGO



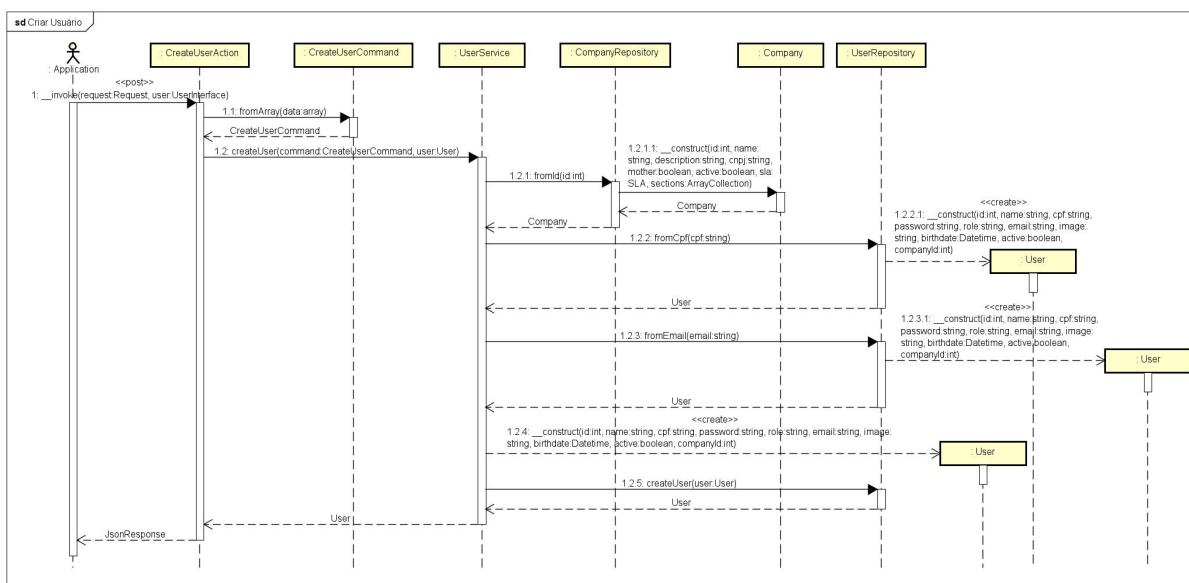
FONTE: Os autores (2020)

Figura E.25: BUSCAR USUÁRIOS POR ID



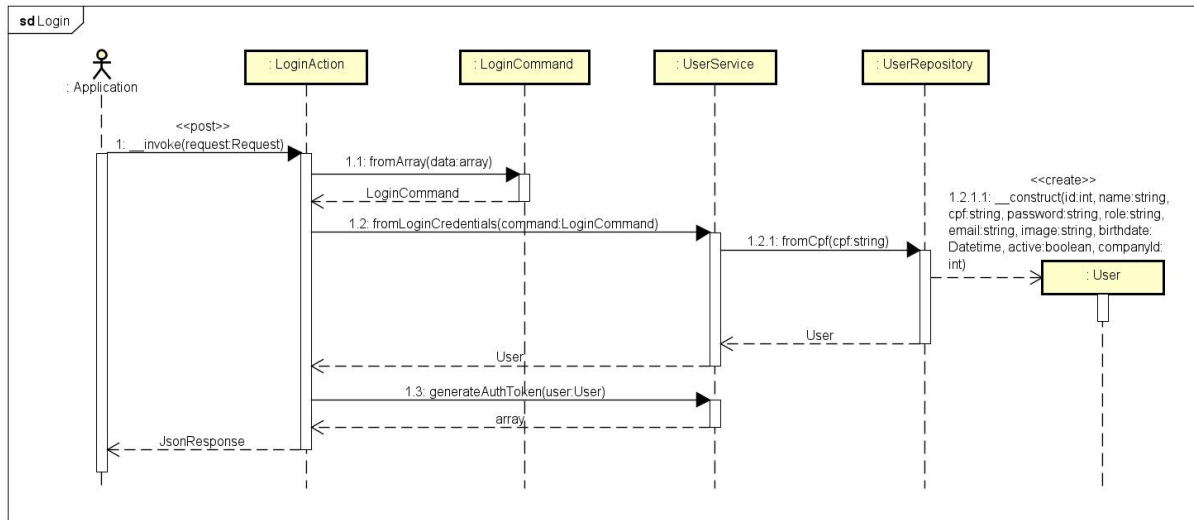
FONTE: Os autores (2020)

Figura E.26: CRIAR USUÁRIO



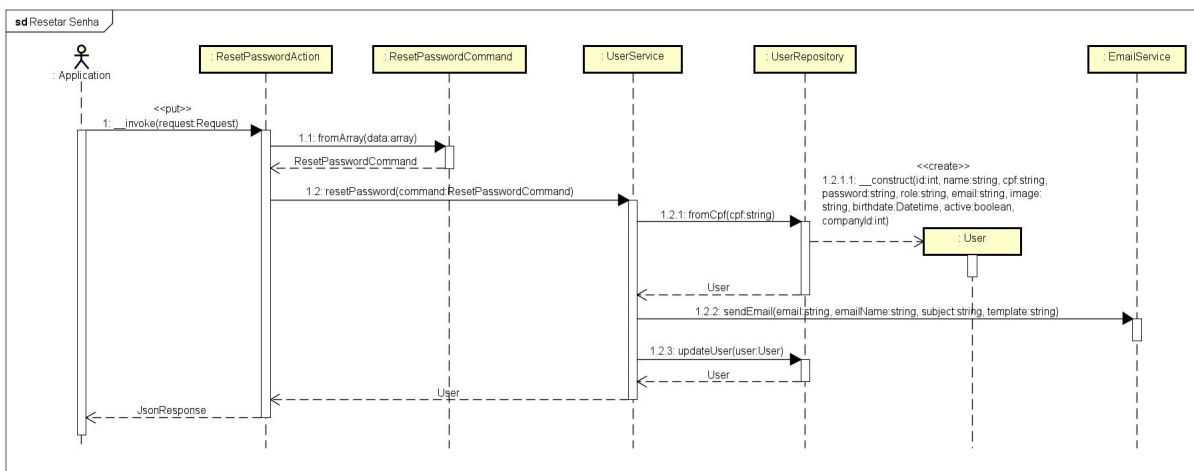
FONTE: Os autores (2020)

Figura E.27: LOGIN



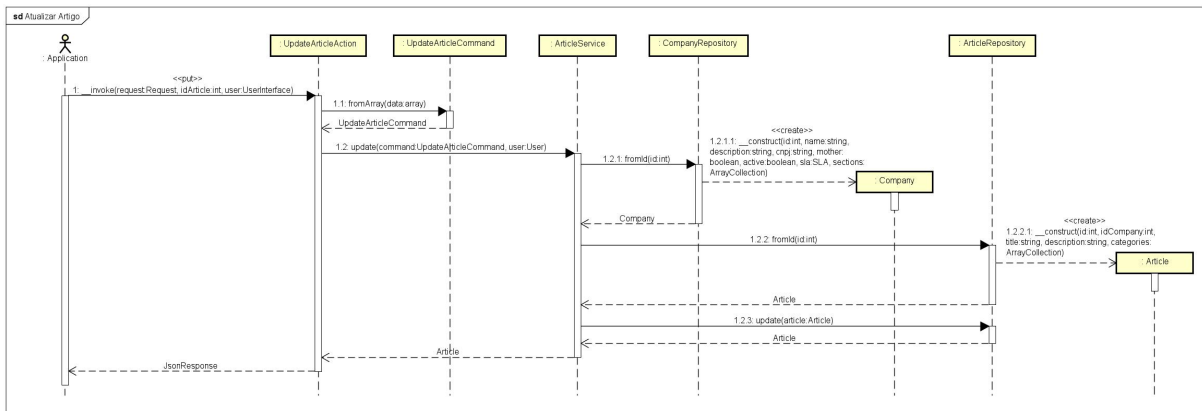
FONTE: Os autores (2020)

Figura E.28: RESETAR SENHA



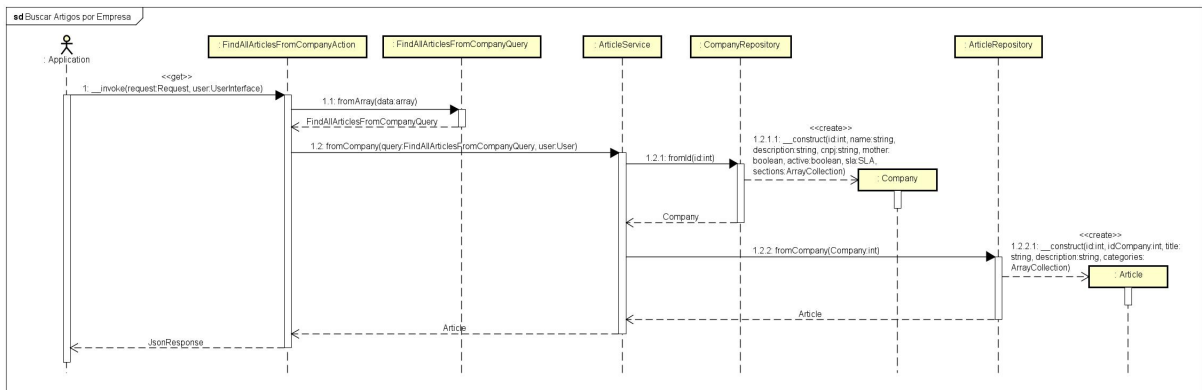
FONTE: Os autores (2020)

Figura E.29: ATUALIZAR ARTIGO



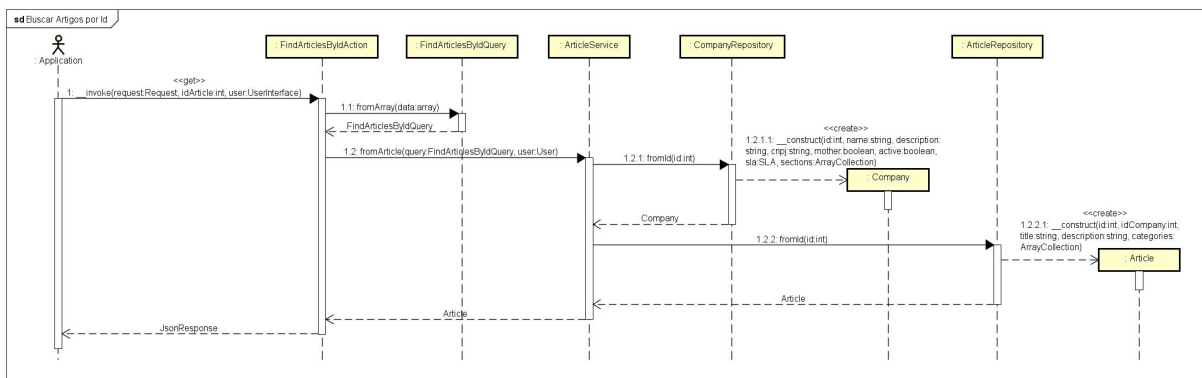
FONTE: Os autores (2020)

Figura E.30: BUSCAR ARTIGOS POR EMPRESA



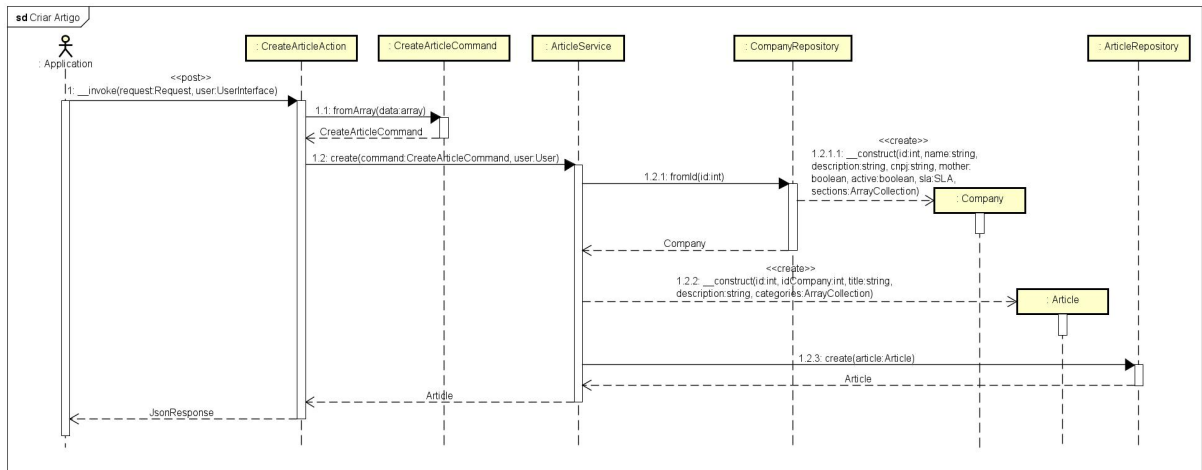
FONTE: Os autores (2020)

Figura E.31: BUSCAR ARTIGOS POR ID



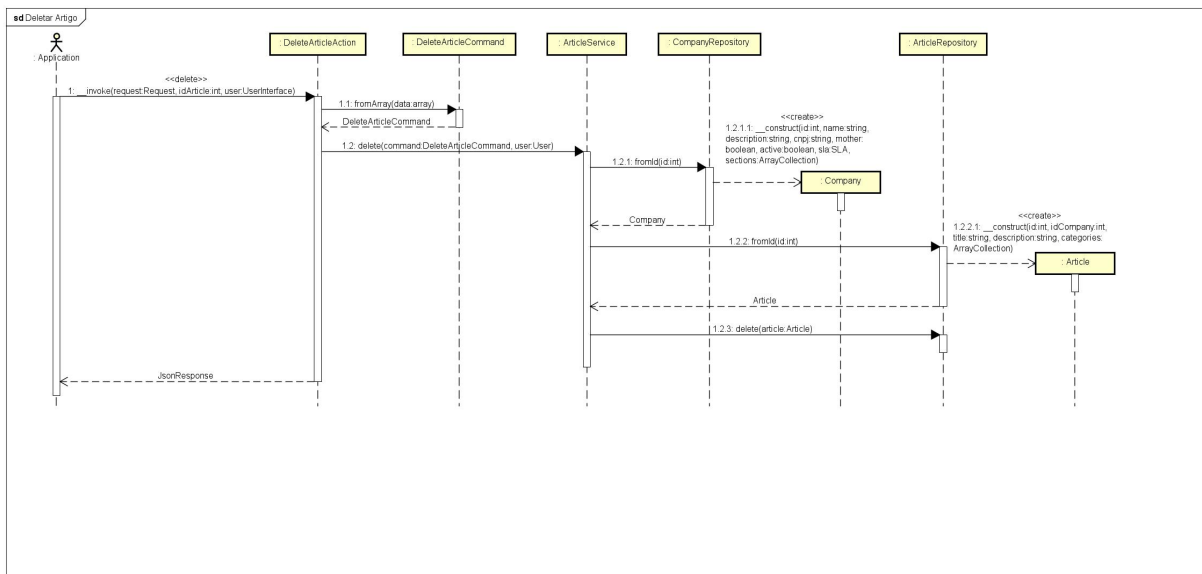
FONTE: Os autores (2020)

Figura E.32: CRIAR ARTIGO



FONTE: Os autores (2020)

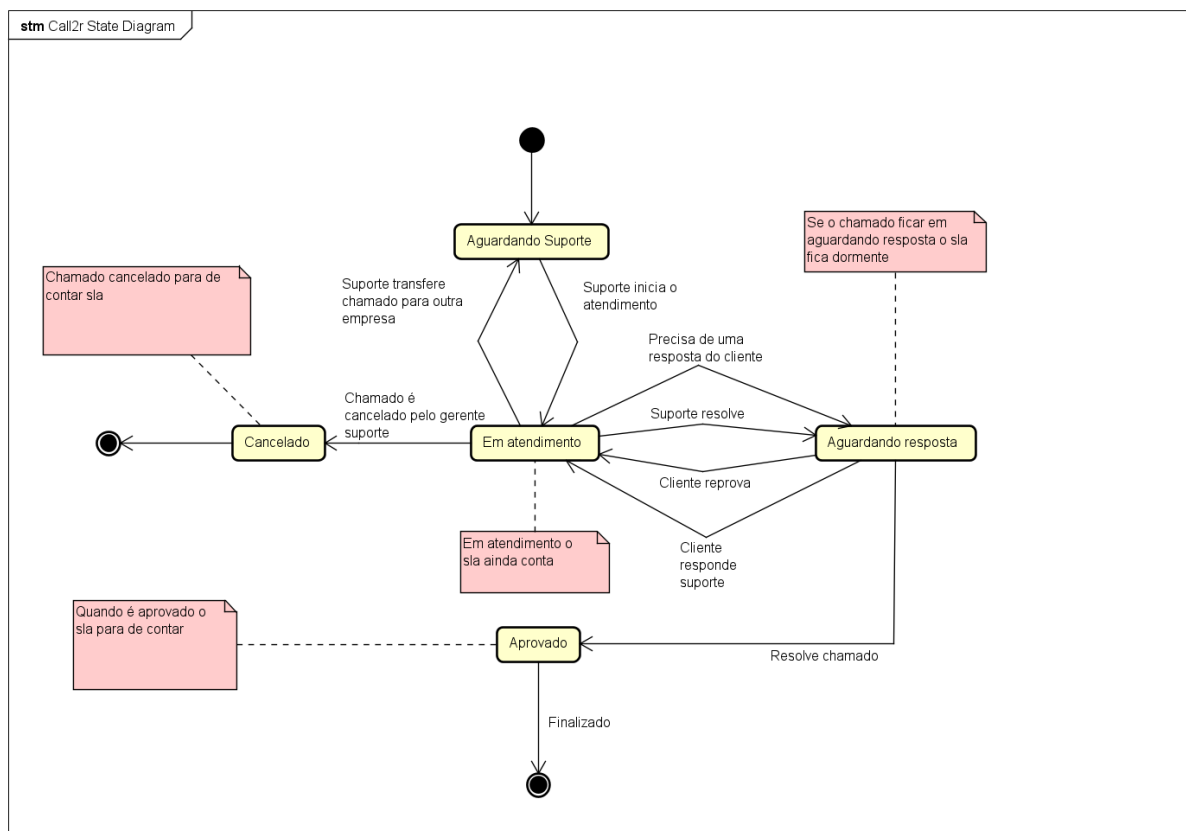
Figura E.33: DELETAR ARTIGO



FONTE: Os autores (2020)

APÊNDICE F – DIAGRAMA DE ESTADOS

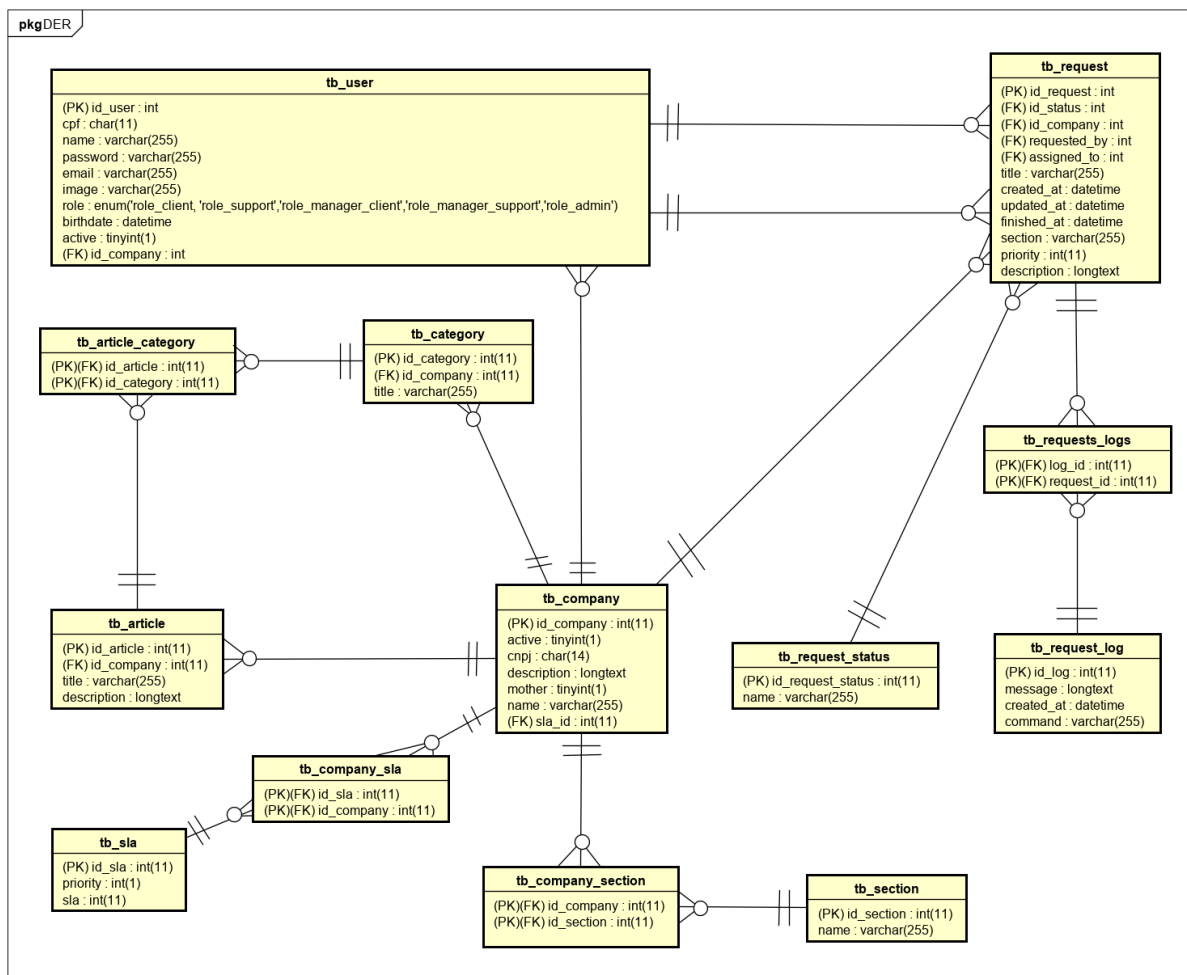
Figura F.1: DIAGRAMA DE ESTADOS



FONTE: Os autores (2019)

APÊNDICE G – DIAGRAMA DE ENTIDADE RELACIONAMENTO

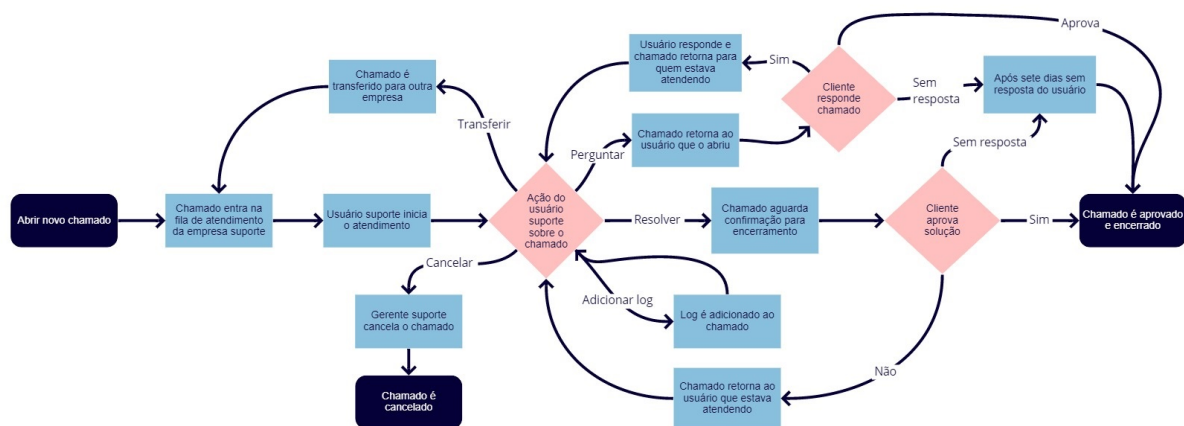
Figura G.1: DIAGRAMA DE ENTIDADE RELACIONAMENTO



FONTE: Os autores (2019)

APÊNDICE H – FLUXOGRAMA

Figura H.1: FLUXOGRAMA DO CHAMADO



FONTE: Os autores (2020)