

FATEC JESSEN VIDAL
DESENVOLVIMENTO DE SOFTWARE E MULTIPLATAFORMA

GRUPO 1 - THE BUGGER DUCKS

ANTONIO ALEXANDRE (Dev Team)

CAIO VITOR (Dev Team)

GABRIEL CAMARGO (Dev Team)

GIOVANA THAÍS (Dev Team)

JOÃO MARCOS (Product Owner)

MARIA GABRIELA (Scrum Master)

OTÁVIO FERRARONI (Dev Team)

THIAGO HENRIQUE (Dev Team)

PROJETO INTEGRADOR- 3º SEMESTRE

São José dos Campos - São Paulo

2022

SUMÁRIO

1. INTRODUÇÃO	4
1.1. Descrição	4
1.2. Objetivo	4
2. DESENVOLVIMENTO	4
2.1. Tecnologias utilizadas	4
2.1.1. Tecnologias para back-end e front-end:	5
2.1.2. Tecnologias para versionamento e comunicação:	5
2.1.3. Tecnologias para gerenciamento do time:	5
2.2. Levantamento de requisitos	5
2.3. Planejamento das sprints	9
2.4. Arquitetura do front-end	11
2.5. Microsserviços	12
2.5.1 Microsserviço de manipulação de chamados	13
2.5.2. Microsserviço de manipulação de usuários e equipamentos	16
2.5.3. Microsserviço de manipulação de soluções	19
2.5.4. Microsserviço de geração de relatórios	21
2.6. Análise de causa raiz	22
3. CONCLUSÃO	22

Resumo

Este Projeto Integrador foi executado baseando-se nos princípios da metodologia ágil SCRUM, através desta, a equipe é submetida a diversos aprendizados dentre os quais envolvem, principalmente, a proatividade, autonomia, colaboração e entrega de resultados (PACER). A iniciativa foi realizada em parceria com uma empresa que propôs o desafio da criação de um programa cuja premissa fosse obstar o retrabalho já enfrentado pela entidade que se destina o Software. O programa desenvolvido opera com auxílio de microsserviços de modo que o back-end e o front-end conversem dissociadamente através de sua integração.

1. INTRODUÇÃO

1.1. Descrição

Neste semestre a temática escolhida para o projeto integrador refere-se a uma aplicação munida de uma base de dados que visa registrar problemas recorrentes dentro da empresa de maneira ampla e geral. A plataforma deverá conter um cadastro de usuários e equipamentos, o acompanhamento do ciclo do chamado, sistema de login completo com diferentes perfis de usuários (administrador, suporte e cliente) e um método de análise de causa raiz.

1.2. Objetivo

A meta é desenvolver uma aplicação HelpDesk que acumule registros os quais serão analisados, manipulados e apresentados de forma que facilite a vida dos usuários, visando apontar as causas das complicações expostas para que seja evitado o retrabalho.

2. DESENVOLVIMENTO

2.1. Tecnologias utilizadas

2.1.1. Tecnologias para back-end e front-end

- Figma, Typescript, HTML 5, CSS 3, React, Node.JS, Java, MongoDB e VS Code
- Spring Framework, Docker, Insomnia, Heroku, ExpressJS, TypeORM e Swagger
- Axios, Netlify, Python e Flask.

2.1.2. Tecnologias para versionamento e comunicação

- Slack, Discord, Github e VS Code

2.1.3. Tecnologias para gerenciamento do time

- Google Sheets, Google Docs, Figma

2.2. Levantamento de requisitos

Se tratando da metodologia ágil o Product Owner ficou encarregado de levantar os requisitos com a empresa parceira ao longo das Sprints. A comunicação entre as duas partes ocorreu via Slack.

Após o fluxo da recolha de informações o time elaborou as Users Stories e definiu o Backlog do Produto preservando a finalidade de resolver as dores do cliente da maneira mais virtuosa possível.

Nas imagens seguintes (imagem 1 e 2) apresentam-se as Users Stories e o Backlog do Produto:


User Stories

Épico	ID	Ator	Ação	Motivo
2	1	Usuário comum	Faz login	Para acessar seu perfil e funcionalidades
2	2	Suporte	Faz login	Para acessar seu perfil e funcionalidades
2	3	Administrador	Faz login	Para acessar seu perfil e funcionalidades
2	4	Administrador	Cadastra novos usuários	Para que eles tenham acesso à plataforma
1	5	Usuário comum	Cria chamado	Para ter seu problema atendido formalmente
1	6	Usuário comum	Acompanha o andamento do chamado	Para saber o status do chamado
1	7	Suporte	Reserva chamado	Para atendê-lo posteriormente
1	8	Suporte	Atende chamado	Para analisar e adicionar uma solução a um problema
14	9	Usuário comum	Avalia solução proposta do chamado	Para registrar sua satisfação com o a solução
11	10	Administrador	Cadastra novos equipamentos	Para que estejam disponíveis na plataforma
2	11	Usuário comum	Modifica suas informações de cadastro	Para que esteja de acordo com o que o usuário quer
2	12	Administrador	Modifica informações do usuário	Para alterar informações dos usuários
1	13	Usuário comum	Comenta no chamado	Para o usuário interagir sobre a resposta do suporte
1	14	Usuário comum	Visualiza todos os chamados fechados	Para encontrar sua resposta em outros chamados
9	15	Cliente	Quer um sistema responsivo	Para utilizar aplicação em celulares, TV's e tablets
4	16	Usuário comum	Quer acessar o centro de soluções	Para buscar possíveis soluções para seu problema
4	17	Suporte	Quer acessar o centro de soluções	Para buscar possíveis soluções para um problema
1	18	Administrador	Quer visualizar relatórios	Para analisar dados de chamados, usuários e equipamentos

(Imagem 1: Users Stories. Fonte: Autoria própria)

Backlog do Produto

Épicos (Requisitos Funcionais)



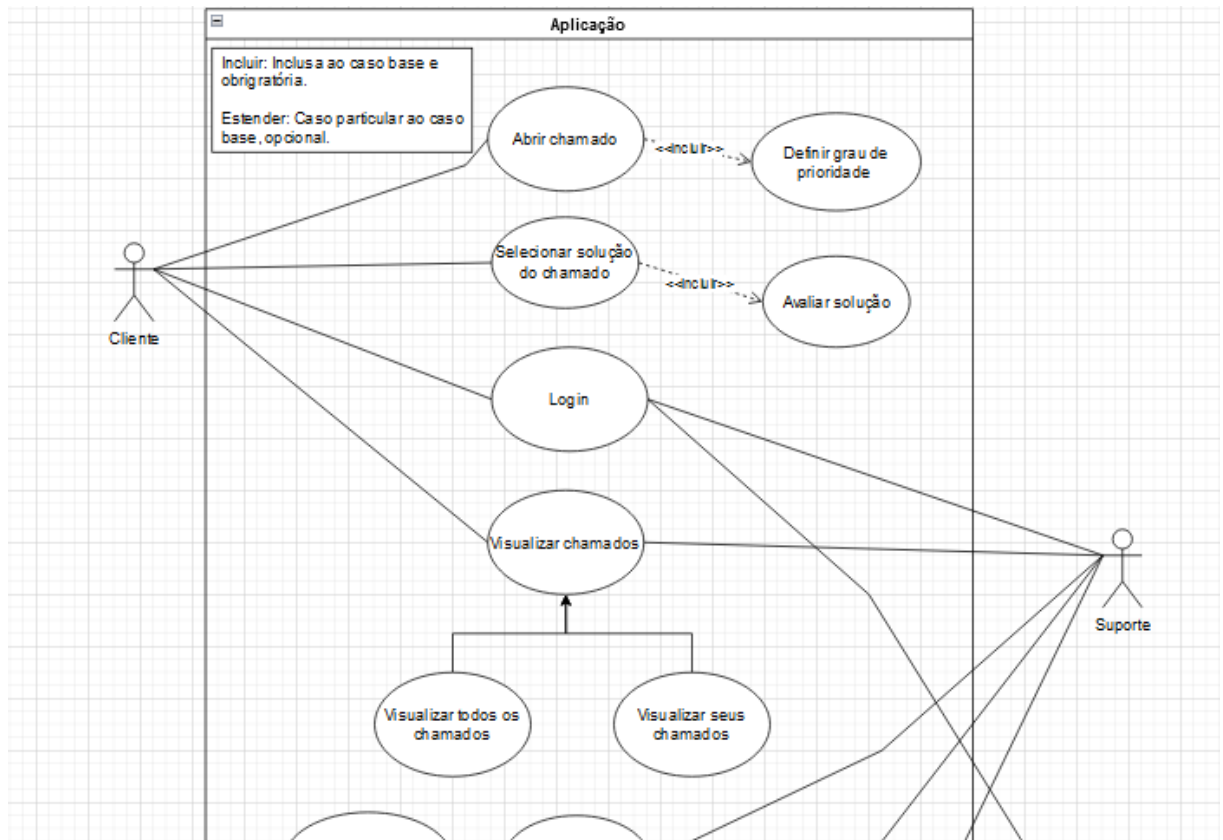
1	Registro e acompanhamento de chamados
2	Gerenciamento e login de usuários
4	Análise de causa raíz
11	Gerenciamento de equipamentos
14	Avaliação de solução do chamado

Requisitos Não Funcionais

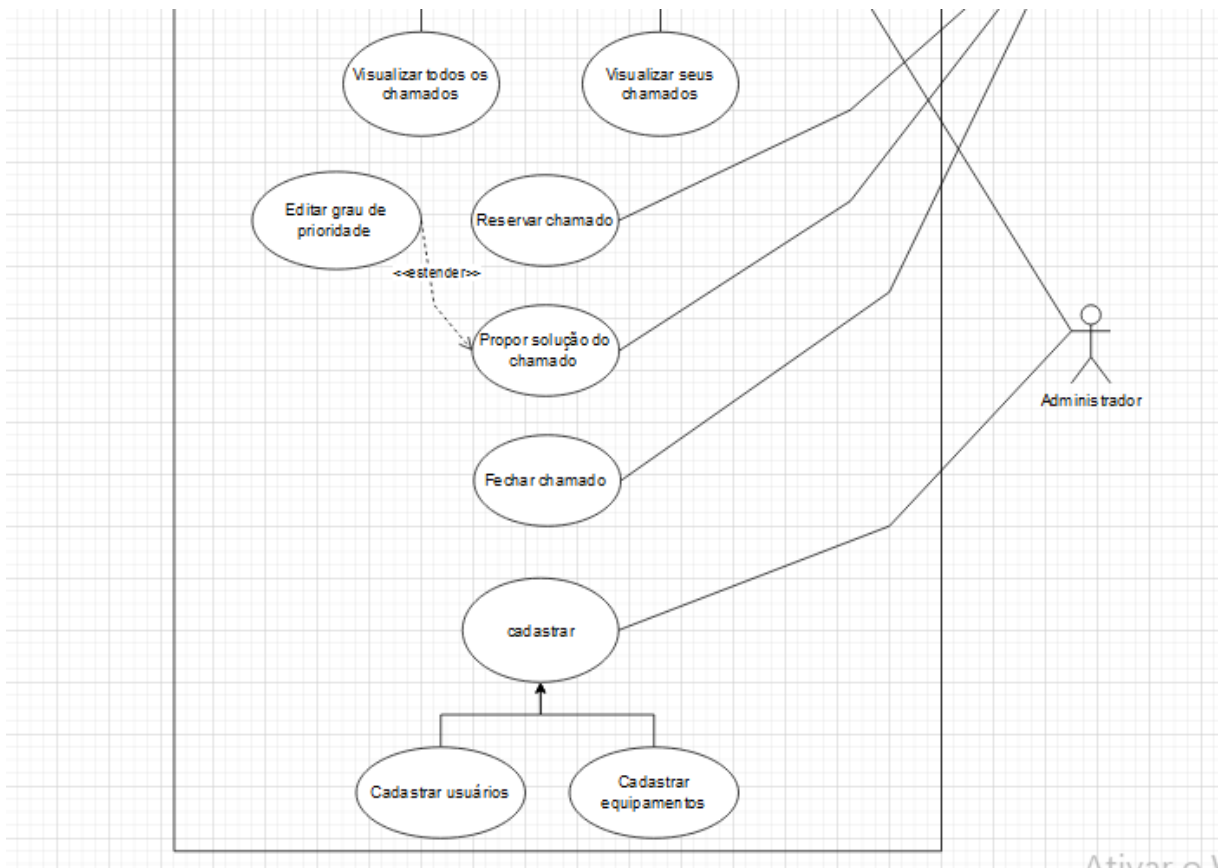
6	Utilizar Java e TypeScript
7	Utilizar banco de dados NoSQL (Sugestão: MongoDB, Redis)
8	Documentação
9	Sistema reponsivo

(Imagem 2: Backlog do Produto. Fonte: Autoria própria)

O Product Owner também desenvolveu um diagrama de casos de uso, que se trata de um esquema descritivo das funções dos usuários do sistema.



(Imagem 3: Diagrama de casos de uso parte 1. Fonte: Autoria própria)



(Imagem 4: Diagrama de Casos de Uso parte 2. Fonte: Autoria própria)

Nesta etapa também, foi desempenhada a validação dos Wireframes e Mockups para construção do protótipo utilizando o React.

2.3. Planejamento das sprints

Tendo em vista o MVP a equipe desjuntou os requisitos para que fosse entregue aqueles de maior valor adequando-os em cada Sprint.

Logo abaixo expõe-se o Backlog das Sprints:

Backlog das Sprints

Sprint 01

1	Registro e acompanhamento de chamados
2	Gerenciamento e login de usuários
6	Utilizar Java e TypeScript
7	Utilizar banco de dados NoSQL (Sugestão: MongoDB, Redis)

Sprint 02

1	Registro e acompanhamento de chamados
2	Gerenciamento e login de usuários
4	Análise de causa raíz
11	Gerenciamento de equipamentos
9	Sistema responsivo

Sprint 03

4	Análise de causa raíz
14	Avaliação de solução do chamado
8	Documentação

(Imagem 5: Backlog das Sprints. Fonte: Autoria própria)

Foi estipulado também, a divisão do time conforme as habilidades dos integrantes em cada frente e as necessidades de cada Sprint.

Inicialmente, Antonio, Caio e Gabriel integraram o Back-end, enquanto Giovana, Otavio e Thiago fizeram parte do Front-end.

Um ponto fundamental durante todo desenvolvimento do projeto foram as numerosas reuniões realizadas, tanto presencialmente quanto distanciadamente. Reuniões estas onde foram determinadas todas as métricas do projeto; a elaboração dos requisitos, a separação dos mesmos por grau de importância com mecanismos

de estimativa, a diluição e quebra das tarefas e por conseguinte, sua distribuição pelo grupo.

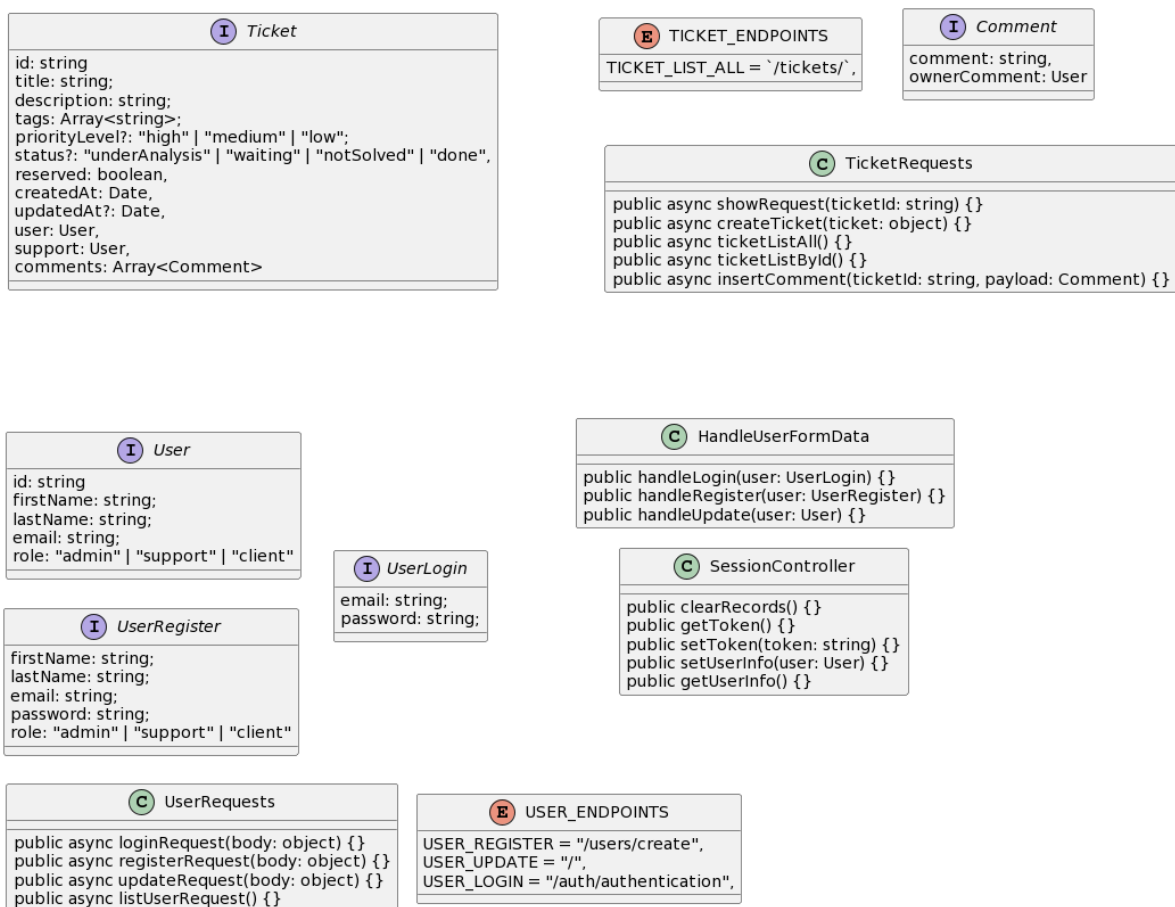
A ferramenta de estimativa escolhida foi o Planning Poker, método que consiste na decisão unânime do grupo (depois de efetuado o alinhamento das expectativas de cada integrante) para etiquetar uma determinada tarefa com um valor de dificuldade. O valor adotado foram os Story Points que conforme estipulado pela equipe, possui uma margem de dificuldade de 0 a 13.

Essa metodologia foi aplicada com o uso de “cartas especiais” que eram atiradas à mesa por todos integrantes ao mesmo tempo após o detalhamento de cada atribuição, e então, com os valores à mostra, o time chegava em um consenso sobre a tarefa e a deixava a cargo de um ou mais indivíduos.

2.4. Arquitetura do front-end

A execução do Front-end se deu criando as interfaces do sistema e consumindo os dados conduzidos pelas APIs do Back-end. Por fim, foi feito o menear dessas informações para contemplação dos usuários.

Ao decorrer das Sprints, o Front-end praticou como fluxo de desenvolvimento o Diagrama de Classes a seguir:



(Imagem 6: Diagrama de Classes do Front-end. Fonte: Autoria própria)

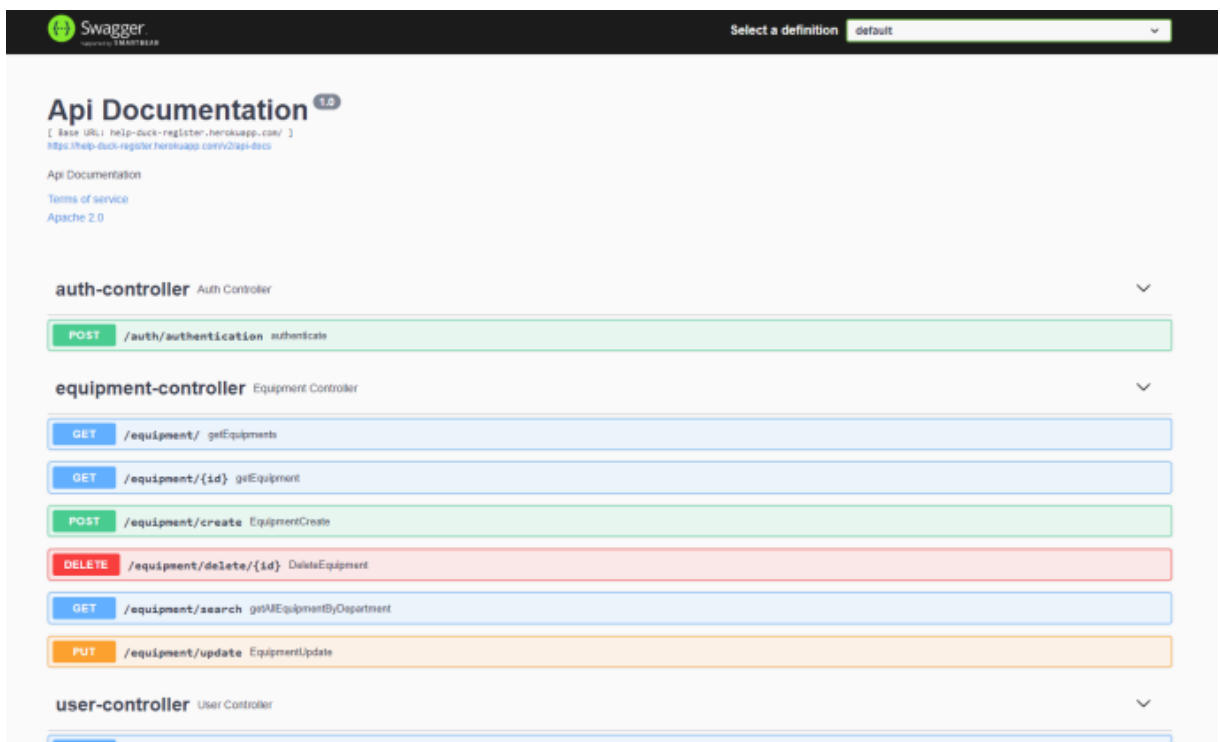
2.5. Microserviços

A arquitetura escolhida foi a de uso de microserviços com o consumo de APIs. Desta forma, será garantido para o software uma maior resiliência e uma escalabilidade mais versátil.

As funcionalidades dos serviços foram sendo incrementadas consoante ao avanço das Sprints.

Para um melhor conforto e eficiência do time, foi feita a implementação do Swagger para alguns microserviços. Sua utilização se deu como uma espécie de documentação de rotas para testes tanto do front-end quanto para o back-end. A partir da segunda Sprint, esses repositórios de dados foram efetivados para os microserviços de: manipulação de chamados, manipulação de usuários e equipamentos e manipulação de soluções.

A imagem a seguir ilustra um pouco da documentação elaborada:



(Imagem 7: Exemplo de documentação no Swagger. Fonte: Autoria própria)

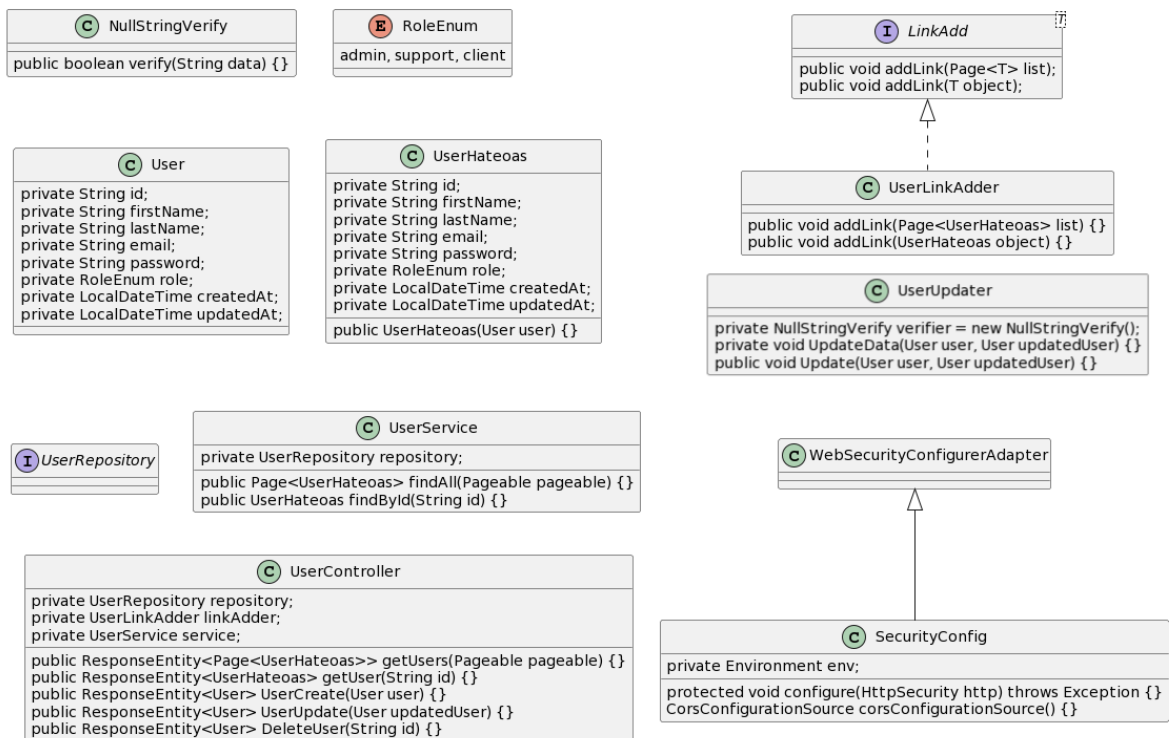
2.5.1 Microserviço de manipulação de chamados

Essa API tem a função de manejar todas as operações que envolvem os chamados. Na Sprint 1 as funcionalidades abordadas foram: a criação, atualização e deleção de chamados, listagem, específica ou não, dos chamados, consulta dos detalhes de determinado chamado, possibilidade de comentar os chamados e a reservação do chamado para atendimento do suporte.

Na Sprint 2 o foco foi: a ordenação dos chamados, a solução do chamado, a identificação do suporte atrelado ao chamado e a permissão de acesso a usuários comuns para o acompanhamento de antigos chamados já fechados de outros usuários.

Na sprint 3 foram feitas as inclusões: dos departamentos, dos problemas com suas soluções e das filtragens com múltiplos elementos ao mesmo tempo, por exemplo, filtro de chamados por usuário, status do chamado e título.

A produção desse serviço baseou-se no seguinte Diagrama de Classes:



(Imagem 8: Diagrama de Classes do serviço de manipulação de chamados. Fonte: Autoria própria)

Ao final das Sprints acordadas o conjunto de rotas disponíveis se apresentou da seguinte maneira:

Tipo	Rota	Ação
GET	/tickets/	Listagem de chamados
GET	/tickets/{ticketId}	Dados de um chamado específico
GET	/tickets/user/{userId}	Chamados de um usuário comum específico
GET	/tickets/support/{userId}	Chamados de um usuário suporte específico
GET	/tickets/status/{statusToFind}	Chamados com um status específico
POST	/tickets/create	Cadastro de chamados
PUT	/tickets/update	Alteração dos dados do chamado
DELETE	/tickets/delete/{ticketId}	Exclusão de chamados
PUT	/helpUser/reserveTicket/{ticketId}	Reserva de chamados
PUT	/helpUser/updateComment/{ticketId}	Adição de comentário no chamado
PUT	/helpUser/closeTicket/{ticketId}	Fechamento de chamado por um usuário suporte

(Imagem 9: Conjunto de rotas do serviço de manipulação de chamados. Fonte: Autoria própria)

Um exemplo da utilização deste microserviço no Front-end pode ser observado na listagem de chamados a seguir:

Help Duck
Olá, user comum

Chamados

Meus chamados

Prioridade	Título	Data de criação	Status
MÉDIA	Máquina não reconhece os periféricos	6/5/2022	Aguardando
BAIXA	Portas de USB não conecta	6/5/2022	Aguardando
ALTA	Pedido de férias	6/5/2022	Aguardando
ALTA	Problema no HD	6/5/2022	Aguardando
MÉDIA	Salário pela metade	6/5/2022	Aguardando
BAIXA	Ativação do Gympass	6/5/2022	Aguardando

Itens por página 20

Página 1 de 1

Acessar centro de soluções

Abrir chamado

Todos os direitos reservados | © The Rubber Ducks

(Imagem 10: Exemplo de listagem de chamados. Fonte: Autoria própria)

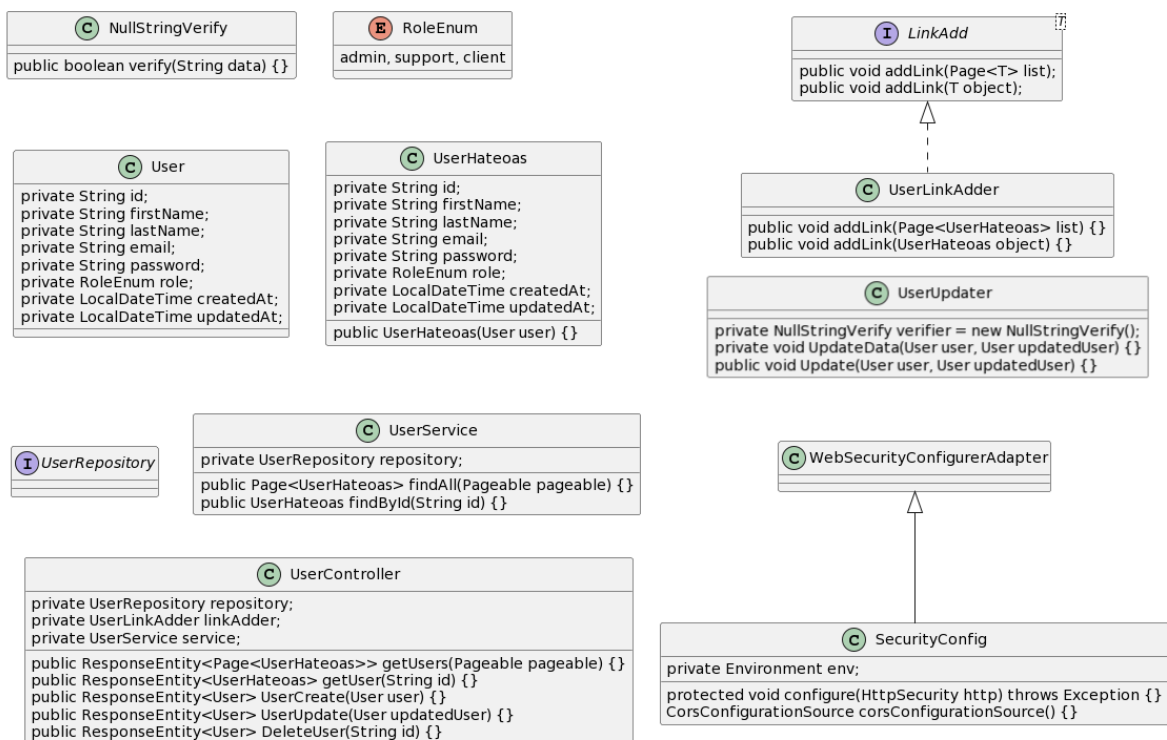
2.5.2. Microserviço de manipulação de usuários e equipamentos

Neste serviço, o objetivo é gerenciar as operações que abrangem os usuários, assim como, abordar as configurações necessárias para que haja a devida autenticação dos usuários, e também, tratar de procedimentos relacionados aos equipamentos. Na Sprint 1 foram implementados: o login e cadastro dos usuários seguindo o padrão de autorização de cada tipo de usuário. Usuários comuns possuem acesso a procedimentos de criação e listagem dos próprios chamados, visualização de detalhes de chamados e adição de comentários. Usuários suporte dispõe de acesso a procedimentos de listagem de qualquer chamado, assim como, a reserva de chamados, adição de comentários e fechamento dos chamados. Usuários administradores possuem admissão para operações de cadastro e listagem de usuários.

Na Sprint 2 foram aderidas: opções de exclusão de perfil e a possibilidade de exclusão de usuários por parte dos administradores, bem como, as configurações de autenticação dos usuários.

Na Sprint 3 foi feita a inserção dos departamentos nos usuários e a filtragem por múltiplos elementos ao mesmo tempo, por exemplo, filtro de usuários por cargo e nome.

O desenvolvimento desse serviço levou em conta o seguinte Diagrama de Classes:



(Imagem 11: Diagrama de Classes do serviço de manipulação de U. e E. Fonte: Autoria própria)

As rotas desta API apresentaram-se da maneira a seguir:

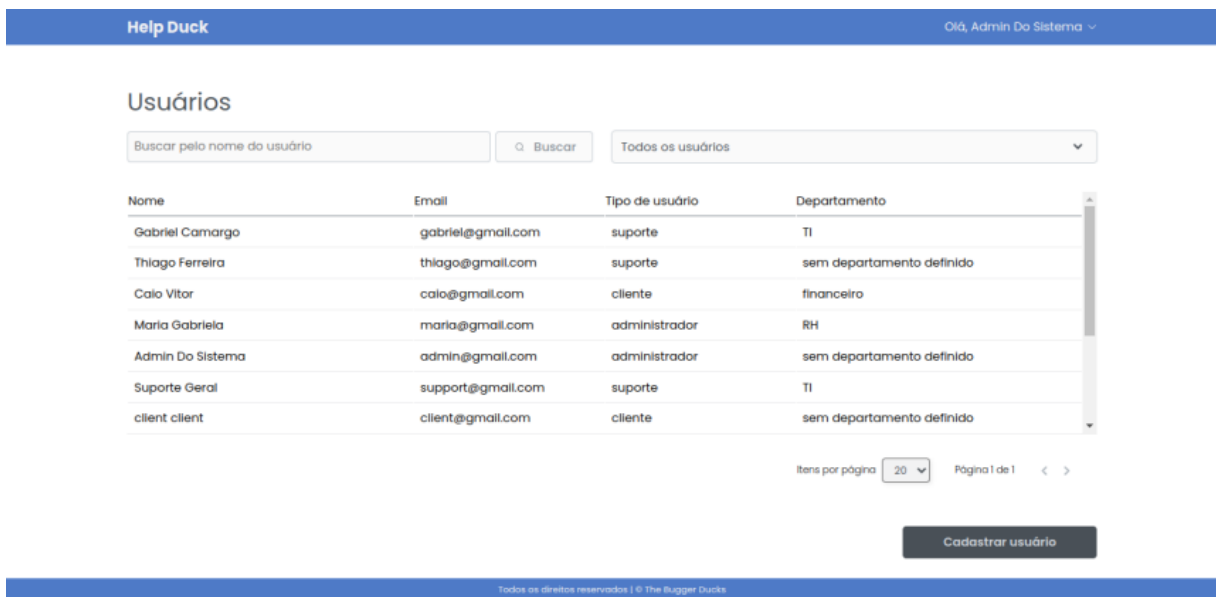
Tipo	Rota	Ação
		Controle de usuários
POST	/auth/authentication	Autenticação de usuários
GET	/users/	Listagem de usuários
GET	/users/{id}	Dados de um usuário específico
POST	/users/create	Cadastro de usuários
PUT	/users/update	Alteração dos dados do usuário
PUT	/users/update-password	Alteração de senha do usuário
DELETE	/users/delete/{id}	Exclusão de usuários
GET	/users/search	Busca de usuários por nome e/ou cargo
		Controle de equipamentos
GET	/equipment/	Listagem de equipamentos

(Imagem 12: Conjunto de rotas do serviço de manipulação de U. e E. parte 1. Fonte: Autoria própria)

		Controle de equipamentos
GET	/equipment/	Listagem de equipamentos
GET	/equipment/{id}	Dados de um equipamento específico
POST	/equipment/create	Cadastro de equipamentos
PUT	/equipment/update	Alteração dos dados do equipamento
DELETE	/equipment/delete/{id}	Exclusão de equipamentos
GET	/equipment/search	Busca de equipamentos por nome e/ou departamento

(Imagem 13: Conjunto de rotas do serviço de manipulação de U. e E. parte 2. Fonte: Autoria própria)

Uma amostra do uso desse serviço no Front-end pode ser analisado através da listagem de usuários a seguir:



(Imagem 14: Exemplo de listagem de usuários. Fonte: Autoria própria)

2.5.3. Microserviço de manipulação de soluções

O microserviço em questão visa gerenciar o escopo das soluções presentes nos chamados.

Teve sua efetivação na Sprint 2 já contando com a funcionalidade de comentários de soluções por parte dos usuários comuns e suporte.

Na Sprint 3 foi estabelecida a solução principal, elaborada apenas pelo suporte, assim como, a avaliação das soluções.

A seguir, o esquema das rotas do serviço de soluções:

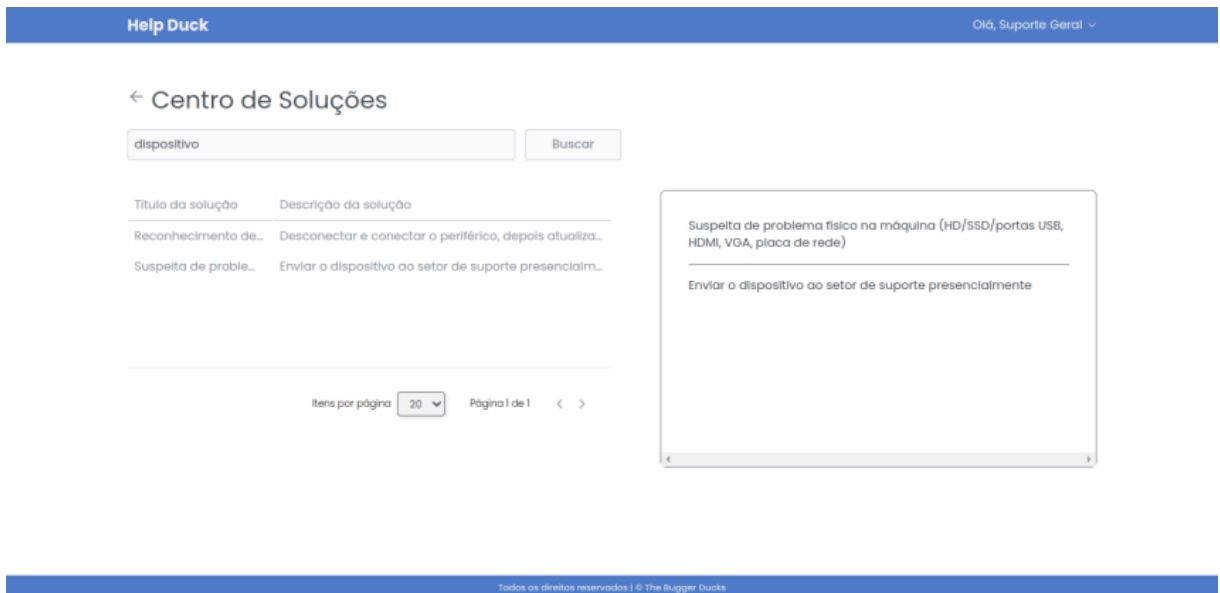
Tipo	Rota	Ação
		Controle de soluções de chamados
GET	/solutions/	Listagem de soluções
GET	/solutions/ticket/{ticketId}	Listagem de soluções por id de um chamado
GET	/solutions/{solutionId}	Dados de uma solução específica
POST	/solutions/create	Cadastro de soluções
PUT	/solutions/update	Alteração dos dados de uma solução
DELETE	/solutions/delete/{solutionId}	Exclusão de soluções
POST	/solutions/vote	Computa os votos de uma solução
		Controle de problemas
GET	/problems/	Listagem de problemas

(Imagem 15: Conjunto de rotas do serviço de manipulação de soluções parte 1. Fonte: Autoria própria)

GET	/problems/	Listagem de problemas
GET	/problems/{problemId}	Dados de um problema específico
POST	/problems/create	Cadastro de problemas
		Controle de soluções de problemas
GET	/solution/	Listagem de soluções
GET	/solution/{solutionId}	Dados de uma solução específica
GET	/solution/search/{searchTitle}	Busca soluções por palavra chave
POST	/solution/create	Cadastro de nova solução

(Imagem 16: Conjunto de rotas do serviço de manipulação de soluções parte 2. Fonte: Autoria própria)

A seguir, um exemplo deste microserviço sendo aplicado no Front-end mediante a interface do Centro de Soluções:



(Imagem 17: Exemplo da interface do Centro de Soluções. Fonte: Autoria própria)

2.5.4. Microserviço de geração de relatórios

Esse serviço tem por finalidade desempenhar a geração de relatórios sobre o sistema, apresentando dados como: usuários por tipo de cargo, total de chamados, total de usuários e etc.

Ele foi implementado na Sprint 3 já com a funcionalidade de visualização de dados gerais do sistema.

Para este microserviço a disposição de rotas ficou da seguinte forma:

Tipo	Rota	Ação
GET	/dashboard/report	Visualização do relatório

(Imagem 18: Conjunto de rotas do serviço de geração de relatórios. Fonte: Autoria própria)

Em seguida, uma amostra do funcionamento do serviço de geração de relatórios no Front-end:

Help Duck

Olá, Admin Do Sistema

← Relatórios

Chamados

Aguardando4

Em Andamento3

Fechados5

Total12

Chamados por problema

Acesso a rede1

Acesso ao email0

Uso de benefícios3

Pagamento2

Mau funcionamento de software2

Documentação insuficiente1

Equipamento danificado3

Outros0

Tempo médio de ações

Reservar chamado0h23min

Fechar chamado0h32min

Usuários cadastrados

Administradores2

Suportes4

Comuns6

Total12

Todos os direitos reservados | © The Rugged Ducks

(Imagem 19: Exemplo da interface de relatórios. Fonte: Autoria própria)

2.6. Análise de causa raiz

Após muitos estudos, a partir da segunda Sprint o grupo iniciou os esforços para aplicar a ACR.

O método confeccionado para tal análise, sustenta-se pela ideia: “Possíveis soluções para certos tipos de problema”. Esse conceito consiste na categorização de chamados por tipos de problemas, de modo que posterior ao encerramento de um chamado, haveria o cadastro de sua solução. Portanto, os chamados com o mesmo tipo de problema gozarão dessas soluções já cadastradas de outros chamados recorrentes, amenizando assim, a carga de retrabalho do suporte

Ainda neste conceito, pensou-se numa maneira de aliviar também o retrabalho por parte do usuário comum. Assim sendo, foi desenvolvido o chamado: “Centro de Soluções”. Este, por sua vez, se refere a uma interface que concede a usuários comuns e suporte a perspectiva de pesquisar por palavras-chave ou expressões com o intuito de obterem todos os resultados referentes a essa busca, englobando componentes como: título do problema, título da solução ou descrição da mesma.

Para visualização do Centro de soluções vide imagem 17 “Exemplo da interface do Centro de Soluções”

3. CONCLUSÃO

Ao final da entrega conclui-se que o grupo exerceu e atingiu com proficiência o desafio proposto pela empresa, pois no desfecho, teve-se um site com todas as operações de login e de chamados, como também, com mecanismos para identificação de falhas e problemas de maneira a driblar o retrabalho. Logo ao fim da primeira Sprint o time já havia entregado todos os principais requisitos, e desta forma, as Sprints subsequentes trataram de implementar melhorias e sugestões apontadas pelo cliente.

A análise de causa raiz, fator predominante do sistema, foi o artefato causador da resolução da principal dor do cliente, o retrabalho.