UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

DEPARTAMENTO ACADÊMICO DE INFORMÁTICA

TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

BRUNO CEZAR VOLPATO LERIA

ESPECIFICAÇÃO DE REQUISITOS: SISTEMA ACIDENTES RODOVIÁRIOS

TRABALHO DA DISCIPLINA DE PROJETO CLIENTE SERVIDOR

PONTA GROSSA

2023

BRUNO CEZAR VOLPATO LERIA

ESPECIFICAÇÃO DE REQUISITOS: SISTEMA FEIRA ONLINE

Trabalho apresentado como requisito parcial à obtenção da nota da disciplina de Projeto Cliente Servidor, do Curso de Tecnologia em Análise e Desenvolvimento de Sistemas da Universidade Tecnológica Federal do Paraná – Campus Ponta Grossa.
Prof. Dr. Richard Duarte Ribeiro

PONTA GROSSA

2023

SUMÁRIO

1 DESCRIÇÃO GERAL DO SISTEMA....3
2 DESCRIÇÃO DOS REQUISITOS....4
2.1 REQUISITOS FUNCIONAIS....4
2.2 REQUISITOS NÃO FUNCIONAIS....6
3 PLANEJAMENTO....7
3.1 DIAGRAMA DE GANTT....7
3.1.1 Descrição das Atividades....7
3.1.2 Diagrama de Gantt....8
4 MODELAGEM....9
4.1 DIAGRAMA DE CASOS DE USO....9
4.2 DESCRIÇÃO DE CASOS DE USO....10
5 DEFINIÇÃO DO BANCO DE DADOS....14
6 PROTOCOLO DE TROCA DE MENSAGENS....15

1 DESCRIÇÃO GERAL DO SISTEMA

Devido às frequentes viagens do professor Richard entre Ponta Grossa e Curitiba, ele identificou a necessidade de desenvolver um sistema para alertar os usuários da rodovia sobre acidentes que ocorrem nela. Como resultado, o professor solicitou aos alunos a criação de um sistema que utilize uma API REST para permitir a comunicação entre o servidor e o cliente.

O sistema terá dois tipos de usuários: aqueles que possuem cadastro e aqueles que não possuem. Os usuários com cadastro terão a capacidade de relatar novos incidentes e filtrar as ocorrências com base no código identificador dos usuários. Além disso, os usuários com cadastro poderão gerenciar suas ocorrências, editando ou excluindo-as. A qualquer momento, um usuário com cadastro poderá atualizar seu perfil, modificando seu nome e senha, ou excluir sua conta.

Este projeto pode proporcionar uma maneira eficiente de compartilhar informações sobre acidentes na rodovia, permitindo que os usuários relatem e acessem informações relevantes. Isso pode contribuir para a segurança no trânsito, reduzir congestionamentos e melhorar a experiência dos motoristas que utilizam a rodovia regularmente.

2 DESCRIÇÃO DOS REQUISITOS

Neste capítulo serão descritos os requisitos do sistema. A seção 2.1 apresenta os requisitos funcionais e a seção 2.2 aborda os requisitos não funcionais.

2.1 REQUISITOS FUNCIONAIS

- **Cadastro de usuário**

  Descrição: O sistema deve permitir que os usuários se cadastrem, fornecendo as informações necessárias, como nome, e-mail e senha. Esse requisito visa criar uma conta para o usuário no sistema, permitindo que ele acesse mais funcionalidades disponíveis, além da listagem de incidentes.

- **Login**

  Descrição: Após o cadastro, os usuários devem ser capazes de fazer login no sistema utilizando suas credenciais de acesso. Esse requisito é importante para autenticar o usuário e permitir que ele acesse as funcionalidades restritas aos usuários cadastrados.

- **Listagem de incidentes**

  Descrição: O sistema deve fornecer uma lista de todos os incidentes ocorridos na rodovia. Essa funcionalidade permite que os usuários visualizem os incidentes relatados por outros usuários, oferecendo informações atualizadas sobre a situação da rodovia.

- **Listagem de incidentes reportados pelo usuário**

  Descrição: Os usuários cadastrados devem ter a capacidade de visualizar uma lista dos incidentes que eles próprios reportaram. Isso permite que eles acompanhem e tenham controle sobre as ocorrências que relataram.

- **Atualizar cadastro no sistema**

  Descrição: Os usuários cadastrados devem ter a possibilidade de atualizar suas informações de cadastro no sistema, como nome, e-mail, senha, entre outros. Essa funcionalidade oferece flexibilidade aos usuários para manterem seus dados atualizados.

- **Reportar incidentes na rodovia**

Descrição: Os usuários cadastrados devem poder relatar novos incidentes ocorridos na rodovia. Esse requisito permite que os usuários contribuam com informações sobre situações perigosas, tais como acidentes, para alertar outros usuários.

- **Atualizar incidente reportado**

  Descrição: Caso um usuário tenha relatado um incidente incorretamente ou queira fazer uma atualização nas informações, o sistema deve permitir que ele atualize os detalhes do incidente reportado anteriormente.

- **Deletar um incidente reportado**

  Descrição: Os usuários cadastrados devem ter a opção de excluir um incidente reportado por eles, caso seja necessário. Essa funcionalidade garante que os usuários tenham controle sobre as ocorrências que reportaram.

- **Deletar um cadastro de usuário**

  Descrição: Os usuários cadastrados devem ter a capacidade de excluir permanentemente sua conta do sistema, caso desejem. Isso permite que os usuários tenham controle sobre seus dados e privacidade.

2.2 REQUISITOS NÃO FUNCIONAIS

- **Segurança**

  Descrição: Acesso à todas as funções da aplicação serão feitas somente com autenticação de usuário através do login e a verificação por parte do Servidor do token gerado e enviado para o usuário.

- **Facilidade de acesso ao sistema**

  Descrição: O usuário poderá acessar a aplicação por meio de um navegador da sua escolha em qualquer computador desktop, laptop ou celular sem a necessidade de instalação de aplicativo em qualquer dispositivo.

- **RESTful**

  Descrição: A aplicação será desenvolvida com tecnologia RESTful.

- **Manutenibilidade**

  Descrição: A aplicação permitirá uma fácil manutenção com a inclusão de novas funcionalidades que o cliente (usuários) possa desejar no futuro.

- **Desenvolvimento em Linguagem TypeScript**

  Descrição: A aplicação foi desenvolvida em linguagem TypeScript, utilizando no front-end Vue.JS 3 e no back-end Nest.JS.

- **Banco de Dados**

  Descrição: A aplicação irá utilizar o NoSQL, MongoDB. Ele é orientado a documentos e não tem relacionamentos entre eles.

3 PLANEJAMENTO

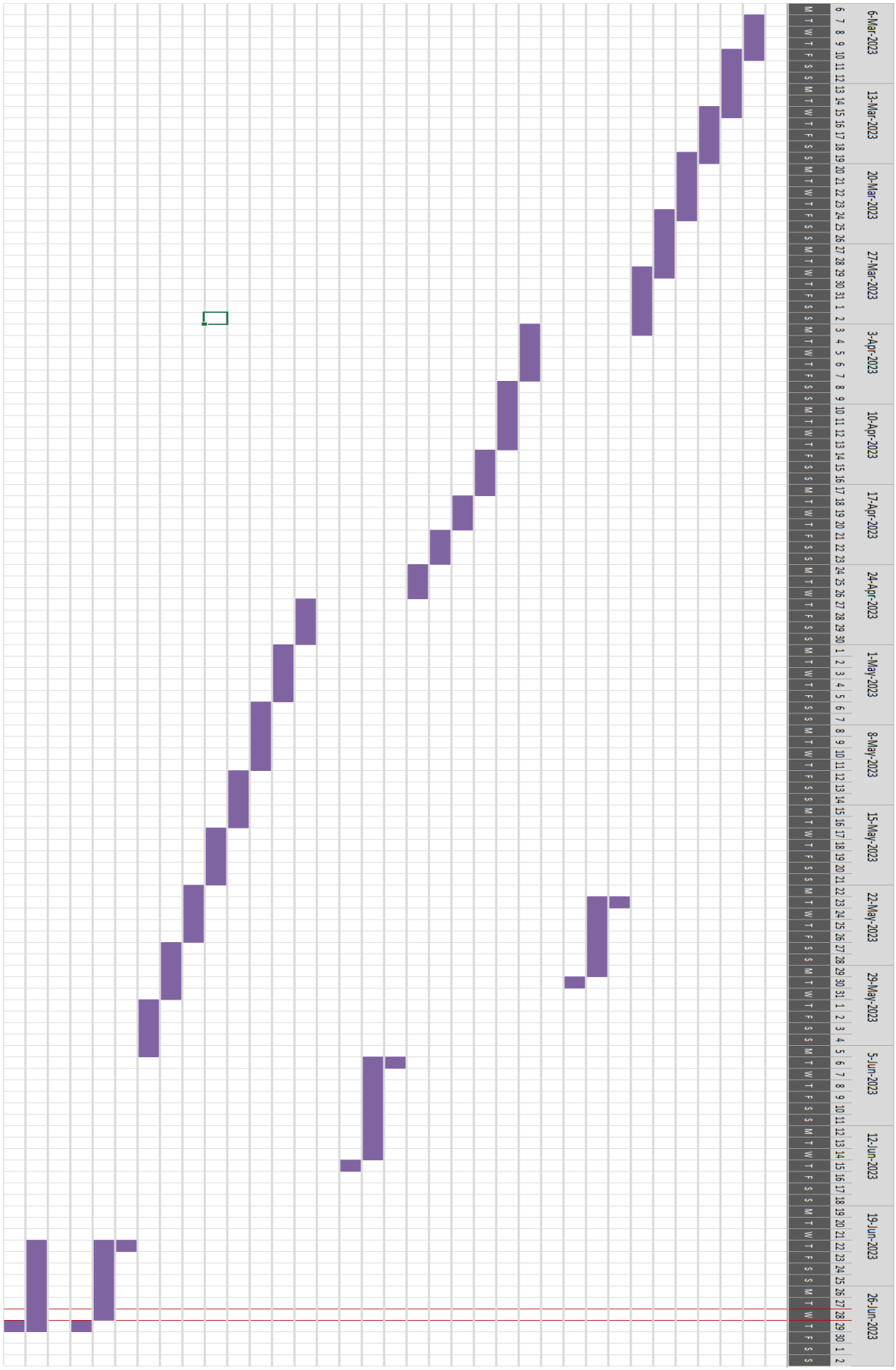
Neste capítulo será descrito o planejamento do projeto. A seção 3.1 exibe o Diagrama de Gantt.

3.1 DIAGRAMA DE GANTT

3.1.1 Descrição das Atividades

TAREFA	SISTEMA	INÍCIO	TÉRMINO
Entrega Parcial 1			
Enviar dados de login para o servidor	CLIENTE	7/3/23	10/3/23
Enviar dados de cadastro de usuário para o servidor	CLIENTE	10/3/23	15/3/23
Enviar dados de logout para o servidor	CLIENTE	15/3/23	19/3/23
Receber a tratar pedido de login do cliente	SERVIDOR	19/3/23	24/3/23
Receber e realizar cadastro de usuário	SERVIDOR	24/3/23	29/3/23
Receber a tratar pedido de logout do cliente	SERVIDOR	29/3/23	3/4/23
Teste da Entrega Parcial 1	CLIENTE/SERVIDOR	23/5/23	23/5/23
Ajustes para a Entrega Parcial 1	CLIENTE/SERVIDOR	23/5/23	29/5/23
Entrega Parcial 1	CLIENTE/SERVIDOR	30/5/23	30/5/23
Entrega Parcial 2			
Envia ao servidor pedido de listagem de incidentes	CLIENTE	3/4/23	7/4/23
Manda atualização de cadastro do usuário no Sistema	CLIENTE	8/4/23	13/4/23
Envia dados de incidente na rodovia ao servidor	CLIENTE	14/4/23	17/4/23
Recebe pedido de listagem de incidentes e a envia para o cliente	SERVIDOR	18/4/23	20/4/23
Recebe atualização de cadastro do usuário e se tudo correto atualiza BD	SERVIDOR	21/4/23	23/4/23
Recebe dados de incidente na rodovia e se tudo correto atualiza BD	SERVIDOR	24/4/23	26/4/23
Teste da Entrega Parcial 2	CLIENTE/SERVIDOR	6/6/23	6/6/23
Ajustes para a Entrega Parcial 2	CLIENTE/SERVIDOR	6/6/23	14/6/23
Entrega Parcial 2	CLIENTE/SERVIDOR	15/6/23	15/6/23
Entrega Parcial 3			
Pedir a listagem de incidentes reportados pelo usuário	CLIENTE	27/4/23	30/4/23
Atualizar um incidente reportado pelo usuário	CLIENTE	1/5/23	5/5/23
Pedir para remover um incidente reportador pelo usuário	CLIENTE	6/5/23	11/5/23
Pedir para remover o cadastro do usuário	CLIENTE	12/5/23	16/5/23
Pedir a listagem de incidentes reportados pelo usuário	SERVIDOR	17/5/23	21/5/23
Atualizar um incidente reportado pelo usuário	SERVIDOR	22/5/23	26/5/23
Pedir para remover um incidente reportador pelo usuário	SERVIDOR	27/5/23	31/5/23
Pedir para remover o cadastro do usuário	SERVIDOR	1/6/23	5/6/23
Teste da Entrega Parcial 3	CLIENTE/SERVIDOR	22/6/23	22/6/23
Ajustes para a Entrega Parcial 3	CLIENTE/SERVIDOR	22/6/23	28/6/23
Entrega Parcial 3	CLIENTE/SERVIDOR	29/6/23	29/6/23
Documentação do Projeto Final			
Criação da Documentação do Projeto Final		22/6/23	29/6/23
Entrega da Documentação do Projeto Final		29/6/23	29/6/23

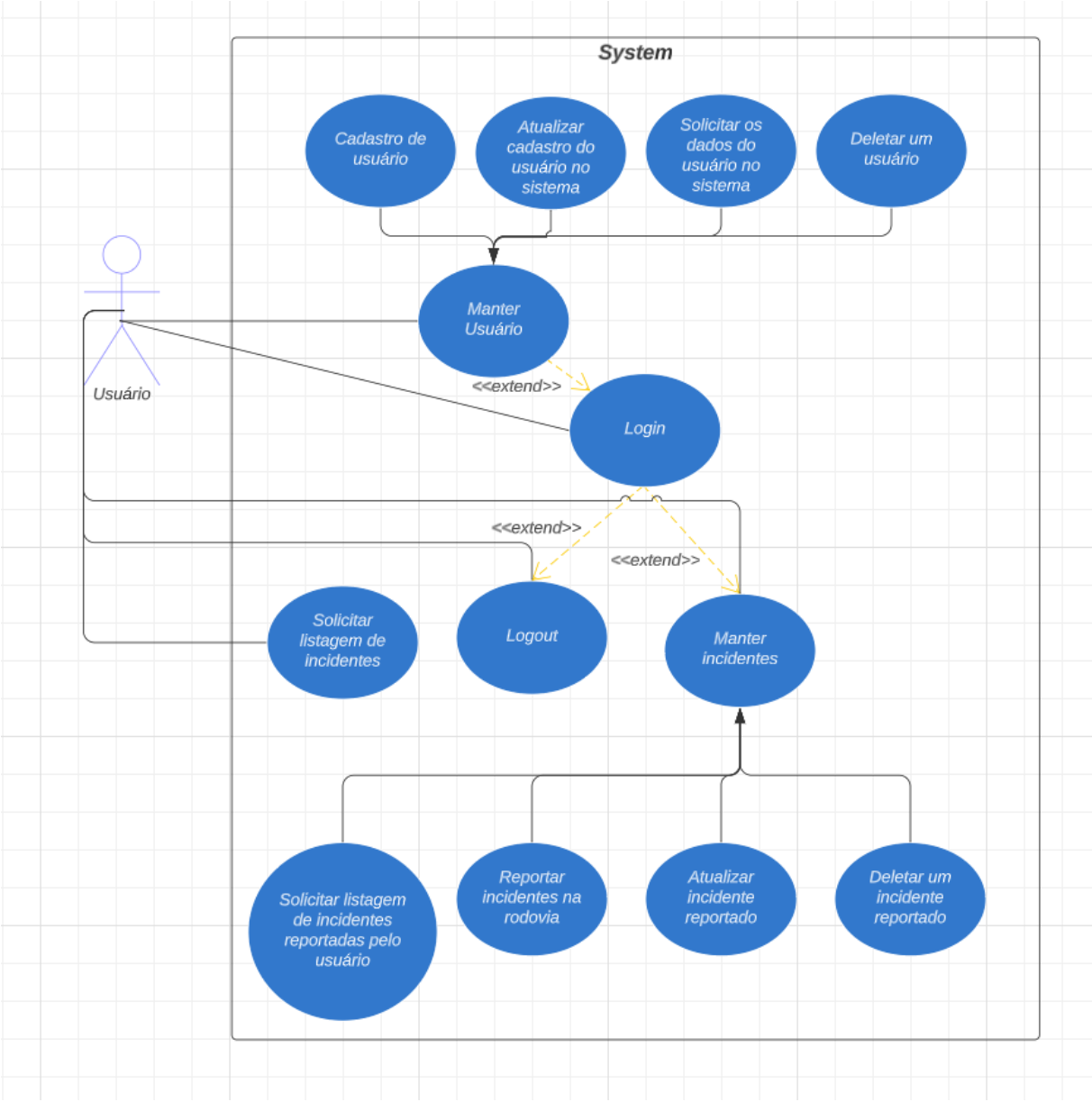
3.1.2 Diagrama de Gantt



4 MODELAGEM

Neste capítulo será abordada a modelagem do sistema. A seção 4.1 apresenta o Diagrama de Casos de Uso, onde demonstra-se a interação do usuário (ator) com a aplicação cliente e a seção 4.2 exibe as Descrições de Casos de Uso, onde apresenta-se a interação da aplicação entre usuário e a aplicação cliente e com o servidor referente ao caso de uso de cada função.

4.1 DIAGRAMA DE CASOS DE USO



4.2 DESCRIÇÃO DE CASOS DE USO

Nome do Caso de Uso	Login
Caso de Uso Geral	
Ator	Usuário
Resumo	Este caso de uso descreve as atividades percorridas pelo usuário para efetuar o login
Pré-Condições	
Pós-Condições	
Fluxo Principal	
Ações do Ator	**Ações do Sistema**
1. Acessa a aplicação pela URL:endereço/	
	2. Aplicação Cliente apresenta Opção de Login
3. Informa o e-mail do usuário e senha	
	4. Aplicação Cliente envia requisição ao servidor com os dados do login
	4.1. Servidor verifica as informações na base de dados e efetua o login do usuário. Gera um token e envia para aplicação Cliente
	4.2. Aplicação Cliente apresenta mensagem: Login Efetuado com sucesso e armazena o token para futuras operações, e apresenta as opções de Profile e Logout
Fluxo Alternativo I – Informar dados incorretos	
Ações do Ator	**Ações do Sistema**
	Servidor retorna mensagem de dados incorretos. Aplicação Cliente irá mostrar uma mensagem informando que os dados estão incorretos
Fluxo Alternativo II – Usuário não está cadastrado	
Ações do Ator	**Ações do Sistema**
	Servidor retorna emite a mensagem de que as credenciais são inválidas. Aplicação cliente irá mostrar a mensagem para o usuário de que as credenciais são inválidas.

Nome do Caso de Uso	Solicitar listagem de incidentes
Caso de Uso Geral	
Ator	Usuário
Resumo	Este caso de uso descreve as atividades percorridas pelo usuário para consultar a lista de incidentes
Pré-Condições	
Pós-Condições	
Fluxo Principal	
Ações do Ator	**Ações do Sistema**
1. Acessa a aplicação pela URL:endereço/	

	2. Aplicação Cliente apresenta a lista de incidentes
Fluxo Alternativo I – Aplicação Cliente não conseguiu conectar com o Servidor	
Ações do Ator	**Ações do Sistema**
	Aplicação Cliente irá mostrar os incidentes disponíveis em cache. Caso não haja incidentes, irá apresentar a tela de consulta vazia.
Fluxo Alternativo II – Não tenha nenhum incidente cadastrado.	
Ações do Ator	**Ações do Sistema**
	Aplicação cliente irá apresentar a tela de consulta vazia.

Nome do Caso de Uso	Logout
Caso de Uso Geral	
Ator	Usuário
Resumo	Este caso de uso descreve as atividades percorridas pelo usuário para efetuar o logout
Pré-Condições	Login
Pós-Condições	
Fluxo Principal	
Ações do Ator	**Ações do Sistema**
1. Acessa a aplicação pela URL:endereço/	
	2. Aplicação Cliente apresenta Opção de Logout
	3. Aplicação Cliente envia a solicitação de logout para o Servidor.
	4. Servidor retorna a mensagem de sucesso.
	5. Aplicação Cliente retorna a página principal.
Fluxo Alternativo I – Usuário solicitante não está autenticado	
Ações do Ator	**Ações do Sistema**
	Servidor retorna emite a mensagem de que retorna o erro informando que o usuário não está autenticado.

Nome do Caso de Uso	Manter usuário
Caso de Uso Geral	
Ator	Usuário
Resumo	Este caso de uso descreve as atividades percorridas pelo usuário para realizar as ações relacionadas a manter um usuário
Pré-Condições	
Pós-Condições	
Fluxo Principal	
Ações do Ator	**Ações do Sistema**

1. Acessa a aplicação pela URL:endereço/	
	2. Aplicação Cliente apresenta Opção de Register
3. Informa o nome, e-mail do usuário e senha	
	4. Aplicação Cliente envia requisição ao servidor com os dados do registro
	4.1. Servidor verifica as informações na base de dados e efetua o registro do novo usuário .
	4.2. Aplicação Cliente retorna a página inicial com o Login realizado, utilizando os mesmos dados que foram informados no registro.
	5. Aplicação Cliente apresenta Opção de Profile
	6. Aplicação Cliente apresenta a página de perfil do usuário após solicitar os dados do usuário no sistema para o servidor. E disponibiliza a edição e exclusão do seu perfil.
Fluxo Alternativo I – Informar dados incorretos	
Ações do Ator	**Ações do Sistema**
	Servidor retorna mensagem de dados incorretos. Aplicação Cliente irá mostrar uma mensagem informando que os dados estão incorretos
Fluxo Alternativo II – Usuário está cadastrando um perfil com um e-mail já existente no banco de dados	
Ações do Ator	**Ações do Sistema**
	Servidor retorna emite a mensagem de que o e-mail já está sendo utilizado. Aplicação cliente irá mostrar a mensagem para o usuário de que o e-mail já está sendo utilizado.

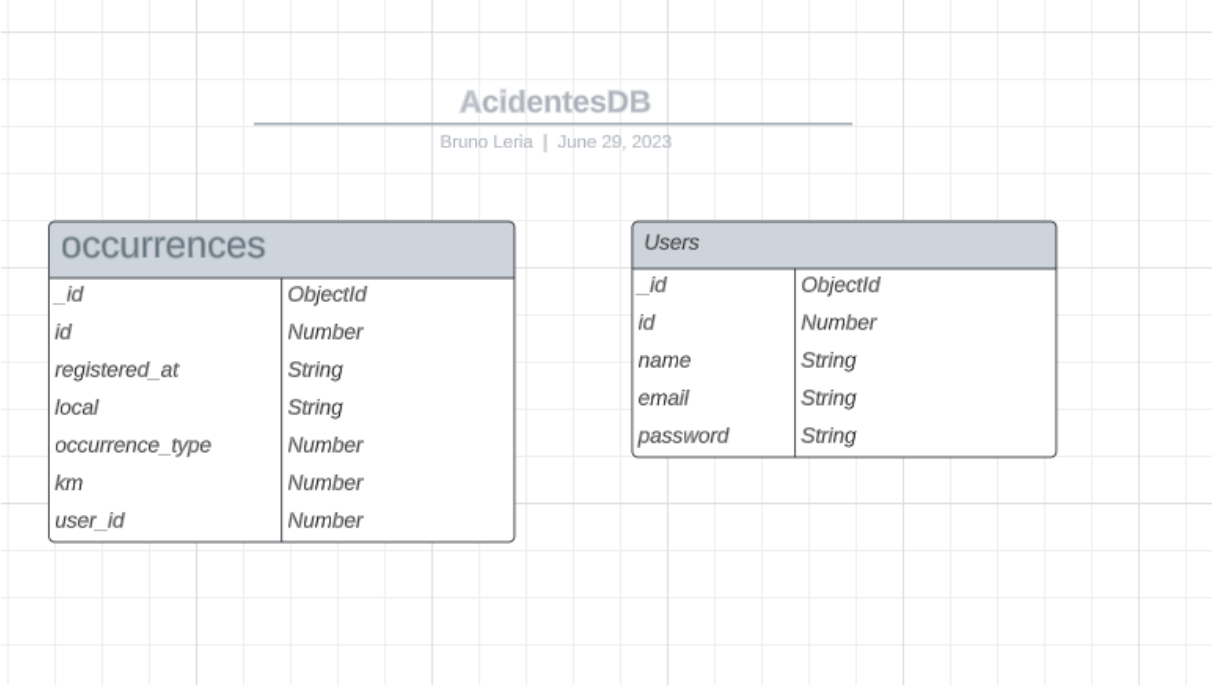
Nome do Caso de Uso	Manter incidentes
Caso de Uso Geral	
Ator	Usuário
Resumo	Este caso de uso descreve as atividades percorridas pelo usuário para realizar as ações relacionadas a manter um incidente
Pré-Condições	Login
Pós-Condições	
Fluxo Principal	
Ações do Ator	**Ações do Sistema**
1. Acessa a aplicação pela URL:endereço/	
	2. Aplicação Cliente apresenta Opção de Adicionar um incidente
3. Informa o local, a data e hora, o tipo do incidente, o quilometro que aconteceu.	

	4. Aplicação Cliente envia requisição ao servidor com os dados do registro
	5. Servidor verifica as informações na base de dados e efetua o registro do novo incidente, e retorna a confirmação.
	6. Aplicação Cliente apresenta o novo incidente no painel e apresenta as Opções de edição e deleção junto ao incidente
	7. Aplicação Cliente apresenta também a opção de solicitar a listagem de incidentes reportadas pelo usuário.
Fluxo Alternativo I – Informar dados incorretos	
Ações do Ator	**Ações do Sistema**
	Servidor retorna mensagem de dados incorretos. Aplicação Cliente irá mostrar uma mensagem informando que os dados estão incorretos
Fluxo Alternativo II – Usuário solicitante não está autenticado	
Ações do Ator	**Ações do Sistema**
	Servidor retorna emite a mensagem de que retorna o erro informando que o usuário não está autenticado. Aplicação cliente irá mostrar a mensagem para o usuário de erro e redirecionar para página de login.

5 DEFINIÇÃO DO BANCO DE DADOS

Neste capítulo é abordado a definição conceitual do banco de dados utilizado na aplicação. O banco de dados é composto por duas tabelas, sendo a tabela usuários que irá conter os dados informados no cadastro de usuários e suas alterações. A tabela incidente irá conter os dados informados no cadastro de incidentes e suas alterações.

A Figura 1 apresenta o esquema conceitual do banco de dados AcidentesDB.



6 PROTOCOLO DE TROCA DE MENSAGENS

Neste capítulo apresenta-se o protocolo de troca de mensagens entre a aplicação cliente e aplicação servidores. Todas as trocas de mensagens foram feitas em arquivo em formato JSON com campo Accept preenchido como application/json. A seção 6.1 aborda o protocolo para o cadastro, alteração, exclusão, consulta, login e logout do usuário e consulta por id do usuário que registrou o incidente, cadastro, alteração, exclusão, consulta dos incidentes.

1	Cadastro de usuário					
CAMPO	**TIPO**	**MIN**	**MAX**	**EXTRAS**	**OBRIGATÓRIO?**	**ENVIADO POR**
name	string	2	125		sim	RequestBody
email	string	10	125	obrigatório o @	sim	RequestBody
password	string	2	125	O hash utilizado será MD5 e o tamanho máx da HASH é 32 dígitos hexadecimais	sim	RequestBody
SITUAÇÃO (201 - SUCESSO)						
Validar todos os campos (tipo, mínimo, máximo, obrigatoriedade, extras), campos são válidos? Erro 400						
SIM:	Continua para a próxima validação.					
NÃO:	Para a execução e retorna o erro informando que os campos são inválidos.					
Já existe usuário com este e-mail? Erro 422						

SIM:	Não permite o cadastro e retorna que o usuário já está cadastrado.
NÃO:	Cadastra o usuário, informa que o cadastro foi realizado com sucesso e devolver os dados do usuário

2	Login

CAMPO	**TIPO**	**MIN**	**MAX**	**EXTRAS**	**OBRIGATÓRIO?**	**ENVIADO POR**
email	string	10	125	obrigatório o @	sim	RequestBody
password	string	2	125	O hash utilizado será MD5 e o tamanho máx da HASH é 32 dígitos hexadecimais	sim	RequestBody

SITUAÇÃO (200 - SUCESSO)	
- Validar todos os campos (tipo, mínimo, máximo, obrigatoriedade, extras), campos são válidos? Erro 400	
SIM:	Continua para a próxima validação.
NÃO:	Para a execução e retorna o erro informando que os campos são inválidos.
- Existe usuário com este e-mail? Erro 401	
SIM:	Continua para a próxima validação.
NÃO:	Emite a mensagem de que as credenciais são inválidas.
- O hash recebido é igual ao hash do banco? Erro 401	

SIM:		Cria o token e envia o token e os dados do usuário para o cliente.				
NÃO:		Emite a mensagem de que as credenciais são inválidas.				
3	Solicitar listagem dos incidentes					
SITUAÇÃO (200 - SUCESSO)						
Existe incidentes?						
SIM:		Retorna o json com o array de incidentes. 200				
NÃO:		Retorna o json vazio 200				
4	Solicita a listagem dos incidentes reportados pelo usuário					
CAMPO	**TIPO**	**MIN**	**MAX**	**EXTRAS**	**OBRIGATÓRIO?**	**ENVIADO POR**
userId	inteiro	1	125		sim	URL
token	JWT Bearer	--	--		sim	Header - Bearer
SITUAÇÃO (200 - SUCESSO)						

O usuário solicitante está autenticado? Erro 401						
SIM:	Continua para a próxima validação.					
NÃO:	Para a execução e retorna o erro informando que o usuário não foi autenticado.					
O usuário informado existe? Erro 401						
SIM:	Continua para a próxima validação.					
NÃO:	Para a execução e retorna o erro informando que o usuário informado não existe.					
O ID informado na URL corresponde ao ID do usuário do token? Erro 401						
SIM:	Retorna um json com as ocorrências cadastradas pelo usuário.					
NÃO:	Para a execução e retorna o erro informando que não é possível realizar a solicitação.					
Existe incidentes?						
SIM:	Retorna o array de incidentes. [{}, {}, {}, {}] 200					
NÃO:	Retorna o array vazio [] 200					
5	Solicitar os dados do usuário no sistema					
CAMPO	**TIPO**	**MIN**	**MAX**	**EXTRAS**	**OBRIGATÓRIO?**	**ENVIADO POR**

id	inteiro	1	125		sim	URL
token	JWT Bearer	--	--		sim	Header - Bearer

SITUAÇÃO (200 - SUCESSO)	
O usuário solicitante está autenticado? Erro 401	
SIM:	Continua para a próxima validação.
NÃO:	Para a execução e retorna o erro informando que o usuário não foi autenticado.
O usuário informado existe? (URL) Erro 401	
SIM:	Continua para a próxima validação.
NÃO:	Para a execução e retorna o erro informando que o usuário informado não existe.
O ID informado na URL corresponde ao ID do usuário solicitante? Erro 401	
SIM:	Retorna um json com as ocorrências cadastradas pelo usuário.
NÃO:	Para a execução e retorna o erro informando que não é possível realizar a solicitação.
Validar o tipo do campo id, campo inválido? Erro 400	
SIM:	Continua para a próxima validação.

NÃO:		Para a execução e retorna o erro informando que os campos são inválidos.				
6	Atualizar cadastro do usuário no sistema					
CAMPO	**TIPO**	**MIN**	**MAX**	**EXTRAS**	**OBRIGATÓRIO?**	**ENVIADO POR**
name	string	2	125		sim	RequestBody
email	string	10	125	obrigatório o @	sim	RequestBody
password	string	2	125	É obrigatório o envio do campo, mas no caso do usuário não atualizar a senha é enviado null como valor do campo O hash utilizado será MD5 e o tamanho máx da HASH é 32 dígitos hexadecimais	sim	RequestBody
id	inteiro	1	125		sim	URL
token	JWT Bearer	--	--		sim	Header - Bearer
SITUAÇÃO (200 - SUCESSO)						
O usuário solicitante está autenticado? Erro 401						
SIM:		Continua para a próxima validação.				
NÃO:		Para a execução e retorna o erro informando que o usuário não foi autenticado.				

O usuário informado existe? (URL) Erro 401						
SIM:	Continua para a próxima validação.					
NÃO:	Para a execução e retorna o erro informando que o usuário informado não existe.					
O ID informado na URL corresponde ao ID do usuário solicitante? Erro 401						
SIM:	Retorna um json com as ocorrências cadastradas pelo usuário.					
NÃO:	Para a execução e retorna o erro informando que não é possível realizar a solicitação.					
Validar todos os campos (tipo, mínimo, máximo, obrigatoriedade, extras), campos são válidos? Erro 400						
SIM:	Continua para a próxima validação.					
NÃO:	Para a execução e retorna o erro informando que os campos são inválidos.					
Existe outro usuário com este e-mail? Erro 422						
SIM:	Não permite a atualização dos dados e retorna que o e-mail já pertence a outro usuário já está cadastrado.					
NÃO:	Atualiza os dados e informa que foi atualizado com sucesso e devolve os dados atualizados para o cliente.					
7	Reportar incidentes na rodovia					
CAMPO	**TIPO**	**MIN**	**MAX**	**EXTRAS**	**OBRIGATÓRIO?**	**ENVIADO POR**

registered_at	string	--	--	Formato ISO (2019-01-25T02:00:00.000Z)	sim	RequestBody
local	string	1	125		sim	RequestBody
occurrence_type	integer	1	10	Listagem fixa informada na ABA tipos	sim	RequestBody
km	integer	1	9999		sim	RequestBody
user_id	integer	1	--		sim	RequestBody
token	JWT Bearer	--	--		sim	Header

SITUAÇÃO (201 - SUCESSO)	
O usuário solicitante está autenticado? Erro 401	
SIM:	Continua para a próxima validação.
NÃO:	Para a execução e retorna o erro informando que o usuário não foi autenticado.
O usuário informado existe? Erro 401	
SIM:	Continua para a próxima validação.
NÃO:	Para a execução e retorna o erro informando que o usuário informado não existe.
O user_id informado na requisição corresponde ao ID do usuário solicitante (ID do token)? Erro 401	

SIM:		Continua para a próxima validação.				
NÃO:		Para a execução e retorna o erro informando que não é possível realizar a solicitação.				
- Validar todos os campos (tipo, mínimo, máximo, obrigatoriedade, extras), campos são válidos? Erro 400						
SIM:		Cadastra o incidente e retorna os dados do incidente cadastrado com código de sucesso.				
NÃO:		Para a execução e retorna o erro informando que os campos são inválidos.				
8	Atualizar incidente reportado					
CAMPO	**TIPO**	**MIN**	**MAX**	**EXTRAS**	**OBRIGATÓRIO?**	**ENVIADO POR**
registered_at	string	--	--	Formato ISO (2019-01-25T02:00:00.000Z)	sim	RequestBody
local	string	1	125		sim	RequestBody
occurrence_type	integer	1	10	Listagem fixa informada na ABA tipos	sim	RequestBody
km	integer	1	9999		sim	RequestBody
user_id	integer	1	--		sim	RequestBody
token	JWT Bearer	--	--		sim	Header - Bearer

SITUAÇÃO (200 - SUCESSO)	
O usuário solicitante está autenticado? Erro 401	
SIM:	Continua para a próxima validação.
NÃO:	Para a execução e retorna o erro informando que o usuário não foi autenticado.
O usuário informado existe? Erro 401	
SIM:	Continua para a próxima validação.
NÃO:	Para a execução e retorna o erro informando que o usuário informado não existe.
O ID informado na ocorrencia corresponde ao ID do usuário solicitante? Erro 401	
SIM:	Continua para a próxima validação.
NÃO:	Para a execução e retorna o erro informando que não é possível realizar a solicitação.
- Validar todos os campos (tipo, mínimo, máximo, obrigatoriedade, extras), campos são válidos? Erro 400	
SIM:	Atualiza o incidente e retorna os dados do incidente atualizado com código de sucesso.
NÃO:	Para a execução e retorna o erro informando que os campos são inválidos.
A ocorrência informada existe? Erro 400	
SIM:	Realiza a deleção e retorna com a mensagem de sucesso.

NÃO:		Para a execução e retorna o erro informando que a ocorrência não existe.				
9	Deletar um incidente reportado					
CAMPO	**TIPO**	**MIN**	**MAX**	**EXTRAS**	**OBRIGATÓRIO?**	**ENVIADO POR**
token	JWT	--	--		sim	Header - Bearer
SITUAÇÃO (200 - SUCESSO)						
O usuário solicitante está autenticado? Erro 401						
SIM:	Continua para a próxima validação.					
NÃO:	Para a execução e retorna o erro informando que o usuário não foi autenticado.					
O usuário informado no JWT existe? Erro 401						
SIM:	Continua para a próxima validação.					
NÃO:	Para a execução e retorna o erro informando que o usuário informado não existe.					
A ocorrência informada existe? Erro 400						
SIM:	Realiza a deleção e retorna com a mensagem de sucesso.					

NÃO:		Para a execução e retorna o erro informando que a ocorrência não existe.				
O ID informado no JWT corresponde ao user_id do incidente? Erro 401						
SIM:		Continua para a próxima validação.				
NÃO:		Para a execução e retorna o erro informando que não é possível realizar a solicitação.				
10	Deletar um usuário					
CAMPO	**TIPO**	**MIN**	**MAX**	**EXTRAS**	**OBRIGATÓRIO?**	**ENVIADO POR**
id	integer	1	--	ID DO USUÁRIO	sim	URL
token	JWT Bearer	--	--		sim	Header - Bearer
SITUAÇÃO (200 - SUCESSO)						
O usuário solicitante está autenticado? Erro 401						
SIM:		Continua para a próxima validação.				
NÃO:		Para a execução e retorna o erro informando que o usuário não foi autenticado.				
O usuário informado existe? Erro 401						

SIM:	Continua para a próxima validação.
NÃO:	Para a execução e retorna o erro informando que o usuário informado não existe.
O ID informado na requisição corresponde ao ID do usuário solicitante? Erro 401	
SIM:	O usuário informado existe? **(ADICIONAR NA BLACKLIST)** Erro 401
NÃO:	Para a execução e retorna o erro informando não é possível realizar a solicitação.

11	Logout					
CAMPO	**TIPO**	**MIN**	**MAX**	**EXTRAS**	**OBRIGATÓRIO?**	**ENVIADO POR**
id	integer	1	--	ID DO USUÁRIO	sim	RequestBody
token	JWT Bearer	--	--		sim	Header - Bearer

SITUAÇÃO (200 - SUCESSO)	
Validar o tipo do campo id, campo inválido? Erro 400	
SIM:	Continua para a próxima validação.
NÃO:	Para a execução e retorna o erro informando que os campos são inválidos.

O usuário solicitante está autenticado? Erro 401	
SIM:	Continua para a próxima validação.
NÃO:	Para a execução e retorna o erro informando que o usuário não foi autenticado.
O usuário informado existe? (URL) Erro 401	
SIM:	Continua para a próxima validação.
NÃO:	Para a execução e retorna o erro informando que o usuário informado não existe.
O ID informado no body da requisição corresponde ao ID do usuário solicitante (ID armazenado no token)? Erro 401	
SIM:	Realiza o logout do usuário. E coloca o token na Blacklist
NÃO:	Para a execução e retorna o erro informando que não é possível realizar a solicitação.