ESCOLA SENAI

“PROF. DR. EURYCLIDES DE JESUS ZERBINI”

Camila Grossi Gomes

Felipe de Abreu Monteiro

Felipe Guimarães dos Reis Lira

Johnny Oliveira Mendes

Thiago Marcelino Aguiar

**SBrakes - Security Brakes**

Campinas SP

2022

Camila Grossi Gomes

Felipe de Abreu Monteiro

Felipe Guimarães dos Reis Lira

Johnny Oliveira Mendes

Thiago Marcelino Aguiar

**SBrakes - Security Brakes**

Projeto apresentado à Escola SENAI “Prof. Dr. Euryclides de Jesus Zerbini” para obtenção do certificado de conclusão do Curso Técnico de Desenvolvimento de Sistemas.

Orientador: Paulo Henrique Pansani

Campinas SP

2022

Camila Grossi Gomes

Felipe de Abreu Monteiro

Felipe Guimarães dos Reis Lira

Johnny Oliveira Mendes

Thiago Marcelino Aguiar

**SBrakes - Security Brakes**

Trabalho de conclusão de curso aprovado como requisito parcial para obtenção do grau de técnico, do curso Técnico de Desenvolvimento de Sistemas da Escola SENAI “Prof. Dr. Euryclides de Jesus Zerbini”.

**BANCA EXAMINADORA**

1º Examinador

2º Examinador

3º Examinador

Campinas SP

2022

A nossas famílias, por terem nos ajudado a chegar até aqui e continuarem a nos suportar.

**AGRADECIMENTOS**

Gostaríamos de agradecer a todos os nossos professores por ter nos dado o suporte que precisávamos para realizar esse trabalho.

Ao professor Paulo Henrique Pansani, pelo apoio e esclarecimento de nossas dúvidas. Ao professor Douglas de Cassio Quinzani Gaspar, por nos socorrer na resolução dos erros de nossos programas.

E aos professores Astarote Borrego e Renato de Mattos Onofre, agradecemos o encorajamento e o auxílio recebidos para desenvolver esse projeto.

Também, aos nossos amigos e familiares, dizemos nosso muito obrigado por ter tornado nossa jornada mais agradável e divertida, além da ajuda necessária para que pudéssemos completar essa importante etapa de nossas vidas.

**“A tarefa não é tanto ver aquilo que ninguém viu, mas pensar o que ninguém ainda pensou sobre aquilo que todo mundo vê.”**

**- Arthur Schopenhauer**

**SUMÁRIO**

[1 INTRODUÇÃO 9](#_Toc113543364)

[2 JUSTIFICATIVA 10](#_Toc113543365)

[3 OBJETIVOS 11](#_Toc113543366)

[3.1. Objetivos Gerais 12](#_Toc113543367)

[3.1.1. Site 12](#_Toc113543368)

[3.1.2. Mobile 12](#_Toc113543369)

[3.1.3. Arduino 13](#_Toc113543370)

[3.2. Objetivos Específicos 13](#_Toc113543371)

[4 PRODUCT BACKLOG 15](#_Toc113543372)

[4.1 Desenvolvimento Mobile 15](#_Toc113543373)

[4.2 Arduíno 15](#_Toc113543374)

[4.3 Desenvolvimento Web 16](#_Toc113543375)

[5 REQUISITOS NÃO FUNCIONAIS 16](#_Toc113543376)

[6 PREMISSAS 17](#_Toc113543377)

[7 RESTRIÇÕES 18](#_Toc113543378)

[8 ANÁLISE DE RISCOS DE UM PROJETO 19](#_Toc113543379)

[*Tabela 1 – Tabela de análise de riscos* 19](#_Toc113543380)

[8.1. Nível e Planos de Ação para os Riscos 19](#_Toc113543381)

[8.2. Planos de ação 20](#_Toc113543382)

[9 Sprints 21](#_Toc113543383)

[9.1. Primeiro Sprint (Mês de Agosto) 21](#_Toc113543384)

[9.1.1. Product Backlog 21](#_Toc113543385)

[9.1.2. Sprint Backlog 21](#_Toc113543386)

[9.1.3. Burn Down Chart 21](#_Toc113543387)

[9.1.4. Diagramas 22](#_Toc113543388)

[9.1.5. Plano de testes 22](#_Toc113543389)

[9.1.5.1. Resultados 22](#_Toc113543390)

[9.1.6. Kanban e Retrospectiva 23](#_Toc113543391)

[9.2. Segundo Sprint (Mês de Setembro) 24](#_Toc113543392)

[9.2.1. Product Backlog 24](#_Toc113543393)

[9.2.2. Sprint Backlog 24](#_Toc113543394)

[9.2.3. Burn Down Chart 24](#_Toc113543395)

[9.2.4. Diagramas 24](#_Toc113543396)

[9.2.5. Plano de testes 24](#_Toc113543397)

[9.2.5.1. Resultados 25](#_Toc113543398)

[9.2.6. Kanban e Retrospectiva 25](#_Toc113543399)

[10 Modelo de Dados 26](#_Toc113543400)

[10.1. Diagrama de Entidade e Relacionamento 26](#_Toc113543401)

[10.2. Modelo lógico do banco de dados 26](#_Toc113543402)

[10.3. Dicionário de dados 26](#_Toc113543403)

[11 PRINCIPAIS TELAS DO SISTEMA 27](#_Toc113543404)

[12 CONCLUSÃO 28](#_Toc113543405)

[12.1. Escreva os resultados obtidos 28](#_Toc113543406)

[12.2. Constatações 28](#_Toc113543407)

[12.3. Sugestões de possíveis aperfeiçoamentos técnicos 28](#_Toc113543408)

[13 REFERÊNCIAS 29](#_Toc113543409)

[14 GLOSSÁRIO 30](#_Toc113543410)

[15 ANEXOS 31](#_Toc113543411)

# INTRODUÇÃO

Os caminhões são um fundamentais para realizar o transporte e distribuição de riquezas pelo Brasil. No entanto, convém salientar que, apesar dos aspectos positivos à sociedade, ao utilizarem os caminhões como instrumento logístico, ambos podem ser prejudiciais quando utilizados durante longos períodos, pois podem apresentar defeitos não perceptíveis que possam comprometer a vida do motorista e de outras pessoas a sua volta.

Destarte, este trabalho tem o objetivo de solucionar a colisão de caminhões em marcha ré que, devido ao seu peso e inclinação da via, os motoristas não conseguem parar o veículo a tempo de evitar acidentes. Portanto, a finalidade deste projeto é desenvolver de um sistema capaz de garantir uma maior segurança à sociedade, isto é, tornar o trânsito brasileiro mais seguro e menos passível de acidentes.

Sendo assim, foi proposto um freio de emergência (SBrakes – Security Brakes), no qual consiste em um braço mecânico dobrado em sua maior parte do tempo. Na ponta do braço mecânico, há uma curva utilizada para calçar o pneu e, na parte posterior, há um cravo que garante a parada do automóvel quando ele está em baixa velocidade. Convém salientar que esse cravo exercerá sua função devido ao peso do caminhão e seu formato pontiagudo, permitindo com que o peso o enfinque no asfalto e pare o caminhão completamente.

# JUSTIFICATIVA

Como salientado, os caminhões são um meio fundamental para realizar o transporte e distribuição de riquezas pelo Brasil. Segundo a Associação Brasileira de Caminhoneiros (Abcam), cerca de 60% do transporte de cargas são realizados pela malha rodoviária. Logo, nota-se a forte influência e demanda desse setor no desenvolvimento da economia do país.

Como dito anteriormente, o caminhão é um meio logístico fundamental para o transporte de mercadorias. Embora apresentem força em questão de tração, ainda assim há problemas quando este veículo possui a necessidade de subir ruas íngremes e, devido ao seu peso, a inclinação das vias somada com outros fatores, tais como chuva, areia e óleo, viabilizam a descida de ré do caminhão mesmo com os freios acionados ou até mesmo acelerando no intuito de evitar uma colisão. Consequentemente, veículos, pessoas, construções públicas e imóveis podem ser danificados.

Além disso, os possíveis encargos atribuídos às empresas e aos motoristas são de um valor monetário elevado. Por esse motivo, é de suma importância que esse empecilho seja solucionado a fim de evitar transtornos aos sistemas corporativos.

# OBJETIVOS

Nesse sentido, será utilizado alguns sensores, dentre eles:

* Acelerômetro: sua função é verificar a inclinação e a velocidade em que o caminhão está se movendo;
* Sensor de corrente: utilizado para verificar a corrente do fio que conecta na luz de ré.

Além disso, será necessário um conector de Bluetooth para fazer a conexão com o aplicativo. Esse aplicativo, por sua vez, servirá como uma forma de facilitar a visualização do *status* do freio (ativado ou desativado). No aplicativo, também será possível modificar esse *status* quando bem entender.

E, como medida de segurança, o motorista também poderá acionar o freio por meio de 2 botões que serão colocados no painel da cabine do caminhão, um acionará e o outro desativará o freio.

Todos esses dispositivos serão conectados a um Arduino, no qual será inserido um programa que analisará os dados enviados dos sensores e acionará o freio quando necessário.

Vale ressaltar que, quando houver intervenção humana, isto é, quando o motorista ativar ou desativar o freio, essa ação não passará pela parte lógica contida no Arduíno.

Esse programa ativará o freio de emergência quando:

* O sistema receber os dados dos sensores e avaliar que é necessário acionar os freios;
* O Arduino receber pelo Bluetooth o comando para acionar o freio;

Fora as situações em que o próprio motorista poderá apertar os botões para acionar os freios.

# Objetivos Gerais

## 3.1.1. Site

Com relação ao site que será desenvolvido, ele será a ferramenta que os encarregados poderão adicionar e remover os funcionários que devem ter acesso ao aplicativo. Eles não terão a capacidade de alterar os dados dos funcionários, como o login e a senha.

Além disso, o site será uma forma de informar as pessoas sobre o nosso produto, com uma explicação detalhada sobre o produto e seus componentes. Também contará com um “FAQ” que responderá a algumas perguntas possíveis, “Quem Somos?” que explicará um pouco mais sobre a equipe que produziu o projeto e a página de Login, que permitirá apenas que pessoas que trabalhem na empresa e que possuam funções administrativas, possam realizar modificações no cadastro de funcionários.

Também contará com um conjunto de páginas que auxiliarão na navegação do usuário. Dentre as páginas *web,* temos:

1. **Dúvidas:** página responsável por apresentar um conjunto das principais dúvidas que o usuário poderá ter em relação ao nosso produto;
2. **Projeto:** detalhamento dos responsáveis pela elaboração do projeto bem como suas respectivas funções;
3. **Empresa:** explicação sobre o produto desenvolvido e os objetivos ao desenvolver o projeto;
4. **Contato:** contém nossos principais canais de informação como redes sociais, telefone e e-mail;
5. **Login**: área destinada apenas funcionários que possuem funções administrativas dentro da empresa;
6. **Dashboard:** controle e monitoramento de todos os funcionários e veículos da corporação.

Ademais, os encarregados poderão selecionar qual caminhão será alocado para qual motorista. Fazendo com que apenas o motorista tenha acesso a um único fre, isto é, aquele que o encarregado o designou. Fora a possibilidade de cadastrar novos caminhões e removê-los.

## 3.1.2. Mobile

Essa solução contém também um aplicativo que terá uma tela de Login que será preenchida pelo motorista com os dados passados a ele pelo encarregado. Depois, ele poderá trocar a senha na página do usuário, no qual terá acesso ao seu login e sua senha atual. Ao clicar no “Alterar senha”, que fica embaixo do campo da senha, ele poderá trocá-la por alguma outra de sua preferência.

Também neste aplicativo, temos a possibilidade de verificar o estado do freio e de ativá-lo ou desativá-lo. Mas para isso será necessário ter conexão com o Bluetooth, e este deve estar conectado no freio obrigatoriamente. Caso isso não ocorra, ele mandará um aviso para o usuário que o freio não foi encontrado.

Além disso o aplicativo terá a funcionalidade de mandar uma notificação para o celular do motorista para o avisar do estado do freio, ela será ativada APENAS quando o freio estiver ativado e, quando ele for desativado, também será enviada uma notificação concluindo que o freio foi desativado. Ao clicar nessa notificação o motorista será direcionado para a página principal do aplicativo, na qual ele poderá desativar ou ativar o freio.

E por último, mas não menos importante, haverá uma página que permite que ele recupere a senha, caso ele tenha se esquecido de qual era e esteja na página de login. Ele fará isso por meio de uma API de e-mail, que enviará uma senha gerada aleatoriamente pelo sistema.

## 3.1.3. Arduino

O Arduino será programado no Arduino IDE para fazer a programação. E ele servirá para coletar os dados dos sensores e analisar em qual momento ele deve ativar o freio de emergência, isto é, sem intervenção humana. Ele não interferirá quando o motorista apertar o botão (da cabine) para ativar o freio, esse botão fará com que o freio seja ativado ou desativado imediatamente.

Com relação ao Bluetooth, o Arduino receberá o comando do celular conectado e executará o comando, sem passar pela análise.

# Objetivos Específicos

* Criar um aplicativo na plataforma Android Studio;
* Criar um site na plataforma Visual Studio Code;
* Criar a programação do Arduino na plataforma Arduino IDE;
* Usar corretamente a metodologia ágil SCRUM;
* Desenvolver um site que permita o cadastro, seleção e exclusão dos funcionários;
* Desenvolver um aplicativo que permita a alteração de senha, o login, e a recuperação de senha, além de permitir a ativação ou desativação do freio pelo Bluetooth;
* Desenvolver a programação de forma a possibilitar a ativação do freio quando comandado ou quando for constatado pela própria programação através dos dados colhidos pelos sensores.

# PRODUCT BACKLOG

Refere-se às funcionalidades que o *software* deverá possuir com o objetivo de satisfazer as necessidades analisadas:

## 4.1 Desenvolvimento Mobile

* **RF01 -** O aplicativo fará o login dos funcionários que foram cadastrados no banco pelo site
* **RF02 -** O aplicativo permitirá que os funcionários que o acessarem ativem/desativem o freio do caminhão que eles usarão para a viajem
* **RF03 -** O aplicativo conectará apenas ao Bluetooth do caminhão, caso ele não esteja disponível ele mostrará que não foi possível encontrar o freio
* **RF04 -** O aplicativo mostrará o status do freio (Ativado ou desativado) no aplicativo
* **RF05 -** O aplicativo permitirá que caso a senha seja esquecida, ela possa ser recuperada pelo e-mail cadastrado no banco
* **RF06 -** O aplicativo permitirá que o funcionário possa mudar a senha caso ele deseje fazer isso

## 4.2 Arduíno

* **RF01 -** O Arduino avaliará a inclinação do caminhão
* **RF02 -** O Arduino avaliará se a luz de ré está ativada ou não
* **RF03 -** O Arduino ativará o freio quando for comandado, ou quando for decidido que é necessário
* **RF04 -** O Arduino avaliará se o freio precisa ser puxado de forma automática (sem intervenção humana)
* **RF05 -** O Arduino receberá a ordem de desativar/ativar o freio do bluetooth do celular
* **RF06 -** O Arduino passará o status do freio para o celular pelo bluetooth
* **RF07 -** O Arduino ativará o freio quando os botões que estão presentes na cabine forem pressionados

## 4.3 Desenvolvimento Web

* **RF01** – Usuários administradores
  + **RF01.1** - O site permite com que usuários administradores possam realizar login e acessar a área administrativa (*Dashboard)*
  + **RF01.2** - O site permite que usuários administradores possam atualizar, remover, cadastrar e selecionar usuários para possíveis alterações de dados
  + **RF01.3** - O site permite que usuários administradores possam redefinir sua senha. A modificação será realizada por meio de uma API.
* **RF02** – Usuários em geral
  + **RF02.1 –** O site disponibilizará uma explicação detalhada do projeto e da equipe que está atuando no desenvolvimento do *software*
  + **RF02.2 –** O site permite que os usuários tirem dúvidas a respeito do projeto por meio de um *chatbot* e por meio do FAQ desenvolvido
  + **RF02.3** – O site permite com que usuários entre em contato com a equipe das mais variadas formas possíveis: redes sociais, e-mail e telefones das centrais de atendimento ao cliente

# REQUISITOS NÃO FUNCIONAIS

Essas são os requisitos dos sistemas que se referem às propriedades dos sistemas e não às suas funcionalidades:

* **RNF01 –** Interface agradável e de fácil manuseio
* **RNF01 –** Rápida comunicação com o banco para a troca de informações

# PREMISSAS

As premissas são os fatores do projeto que são assumidos como verdadeiros, reais ou certos sem a necessidade de prova ou demonstração.

* **PRE01 –** O projeto será realizado por 5 alunos.
* **PRE02 –** O projeto será desenvolvido dentro e fora do ambiente escolar.
* **PRE03 –** O trabalho será desenvolvido utilizando a metodologia Ágil Scrum.
* **PRE04 –** O trabalho será desenvolvido com o Visual Studio Code, Android Studio, Arduino IDE e o Word.
* **PRE05 –** As tarefas serão distribuídas e monitoradas por cada integrante, contudo, é de responsabilidade de todos o conhecimento por completo do trabalho.
* **PRE06 –** O trabalho será feito nas segundas, terças e sextas, exceto feriados. Também poderão ser realizados em períodos fora do SENAI quando for necessário.
* **PRE07 –** Os arquivos serão mandados para o GitHub ao final de cada aula
* **PRE08 –** O projeto deverá ser finalizado, testado e apresentado até dia 02 de dezembro de 2022.

# RESTRIÇÕES

* **RES01 -** O aplicativo deve ser desenvolvido para Android. Dispositivos IOS não terão compatibilidade com o sistema nesse primeiro momento.
* **RES02 -** O site deve ser editado/desenvolvido usando React e o NodeJS no Visual Studio Code.
* **RES03 -** Para a desenvoltura do projeto faz-se necessário o uso da internet.
* **RES04 -** O aplicativo precisará de uma versão posterior ou igual ao Android 7.0.0.

# ANÁLISE DE RISCOS DE UM PROJETO

A tabela a seguir contém a análise de riscos que, possivelmente, podem ocorrer durante o projeto. Ademais, encontra-se o seu nível de impacto, probabilidade e risco.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TABELA DE ANÁLISE DOS RISCOS** | | | | |
| **ID** | **Ameaças** | **Impacto** | **Probabilidade** | **Risco** |
| 1 | Atraso do projeto | 4 | 3 | 12 |
| 2 | Tecnologia desconhecida | 5 | 4 | 20 |
| 3 | Falta de energia | 5 | 2 | 10 |
| 4 | Falta de um integrante | 3 | 3 | 9 |
| 5 | Mudança de planejamento | 4 | 2 | 8 |
| 6 | Alterações inesperadas no cronograma | 3 | 2 | 6 |
| 7 | Complexidade do projeto não mensurada | 5 | 1 | 5 |

# *Tabela 1 – Tabela de análise de riscos*

# Nível e Planos de Ação para os Riscos

Foi definida uma hierarquia do nível dos riscos, do menos grave para o mais grave. Assim, damos uma maior atenção às ameaças com maior impacto e probabilidade de acontecer.

**

|  |  |
| --- | --- |
| **NÍVEL DE RISCOS** | |
| **ID** | **Riscos** |
| 2 | 20 |
| 1 | 12 |
| 3 | 10 |
| 4 | 9 |
| 5 | 8 |
| 6 | 6 |
| 7 | 5 |

*Tabelas 2, 3 e 4 – Tabelas de riscos, probabilidade e nível de riscos respectivamente*

# Planos de ação

Esses são os planos que pensados caso algum risco ocorra, para que assim possamos lidar com eles de forma a não atrasar a entrega do projeto final.

|  |  |
| --- | --- |
| **ID** | **SOLUÇÃO** |
| 1 | Trabalhar no projeto em casa para compensar o que não foi feito no dia letivo. |
| 2 | Estudar sobre a tecnologia fora dos horários destinado para o projeto para que isso não o atrase. |
| 3 | Compensar o dia perdido no contraturno. |
| 4 | Continuar as partes do projeto faltante que não necessitem do integrante, ou caso isso não seja possível, tentar fazer a comunicação com essa pessoa para que se possa prosseguir com o projeto. |
| 5 | Se reorganizar e reorganizar o *planner* tendo em mente o tempo restante do projeto. |
| 6 | Focar no que essencial para o cumprimento do projeto de forma a deixá-lo funcional. |
| 7 | O grupo se reunirá e realocará as tarefas para que a tarefa não fique encarregada a apenas um único integrante da equipe |

*Tabela 5 – Tabela do plano de ação*

# Sprints

# Primeiro Sprint (Mês de Agosto)

No primeiro Sprint, foram separadas as tarefas a serem feitas por cada integrante. Decidiu-se, então, começar pela montagem dos *wireframes* das telas *mobile e web*, de forma conjunta, junto com a decisão do nome e o design do logotipo que serão utilizados nas duas plataformas.

Também, foi possível completar nesse Sprint as telas do aplicativo Mobile, faltando apenas a parte lógica dele, além de ter sido feito o Banco de Dados que será utilizado, tanto para o site quanto para o aplicativo.

# Product Backlog

[Não](http://www.devmedia.com.br/curso/introducao-a-uml/128) houve alterações desde o Product Backlog inicial

# Sprint Backlog

* **Mobile:** 
  + **RF01 -** O aplicativo fará o login dos funcionários que foram cadastrados no banco pelo site
* **Site:**
  + **RF01** – Usuários administradores
    - **RF01.1** - O site permite com que usuários administradores possam realizar login e acessar a área administrativa (*Dashboard)*
    - **RF01.2** - O site permite que usuários administradores possam atualizar, remover, cadastrar e selecionar usuários para possíveis alterações de dados
    - **RF01.3** - O site permite que usuários administradores possam redefinir sua senha. A modificação será realizada por meio de uma API.

# Burn Down Chart

[Esse](http://www.devmedia.com.br/curso/introducao-a-uml/128) gráfico representa a quantidade de horas trabalhadas durante esse Primeiro Sprint, contando que trabalhamos, aproximadamente, quatro horas diárias, durante trinta e dias a partir do dia 01 de agosto, até o dia 01 de setembro. Totalizando aproximadamente 110,4 horas previstas e, terminado no último dia com quatro horas trabalhadas, cumprindo então as horas programadas.

*Imagem 1 - Burn Down Chart de agosto*

# Diagramas

Nesse Sprint não foram feitos nenhum Diagramas.

# Plano de testes

Nesse Sprint não foram feitos nenhum Teste

# Resultados

Não houve resultados.

# Kanban e Retrospectiva

*Imagem 2 - Trello do 1ºMês*

Nesse Sprint, foi necessário dar ênfase a construção das telas e programar o banco, bem como começar a documentação. No Android, além dos *wireframes*, foi realizado a programação lógica. Porém, ao tentar modificar uma das telas do aplicativo, houve um problema e foi necessário atrasar o desenvolvimento do *software* para que o professor pudesse auxiliar a equipe na tentativa de solucionar o problema. Com relação ao site, iniciou-se a realização das telas e a conexão com o banco de dados. O site já possui o cadastramento de funcionários e permite o login dos administradores. Mas em geral não houve maiores problemas e a equipe se encaixa dentro do cronograma esperado.

# Segundo Sprint (Mês de Setembro)

No segundo sprint, a equipe apresentou alguns imprevistos o que levou o reiniciar o aplicativo *mobile.* Devido ao *software* utilizado (Android Studio), o grupo apresentou problemas na construção da aplicação. Além disso, o integrante da equipe responsável por idealizar as telas do Android, não entregou o produto a tempo, comprometendo a entrega no prazo estimado.

# Product Backlog

Mudança na linguagem de programação para construção do aplicativo. A partir de agora, será utilizado o Visual Studio Code como ferramenta para construção da aplicação e será utilizada o *framework* Flutter e a linguagem de programação Dart.

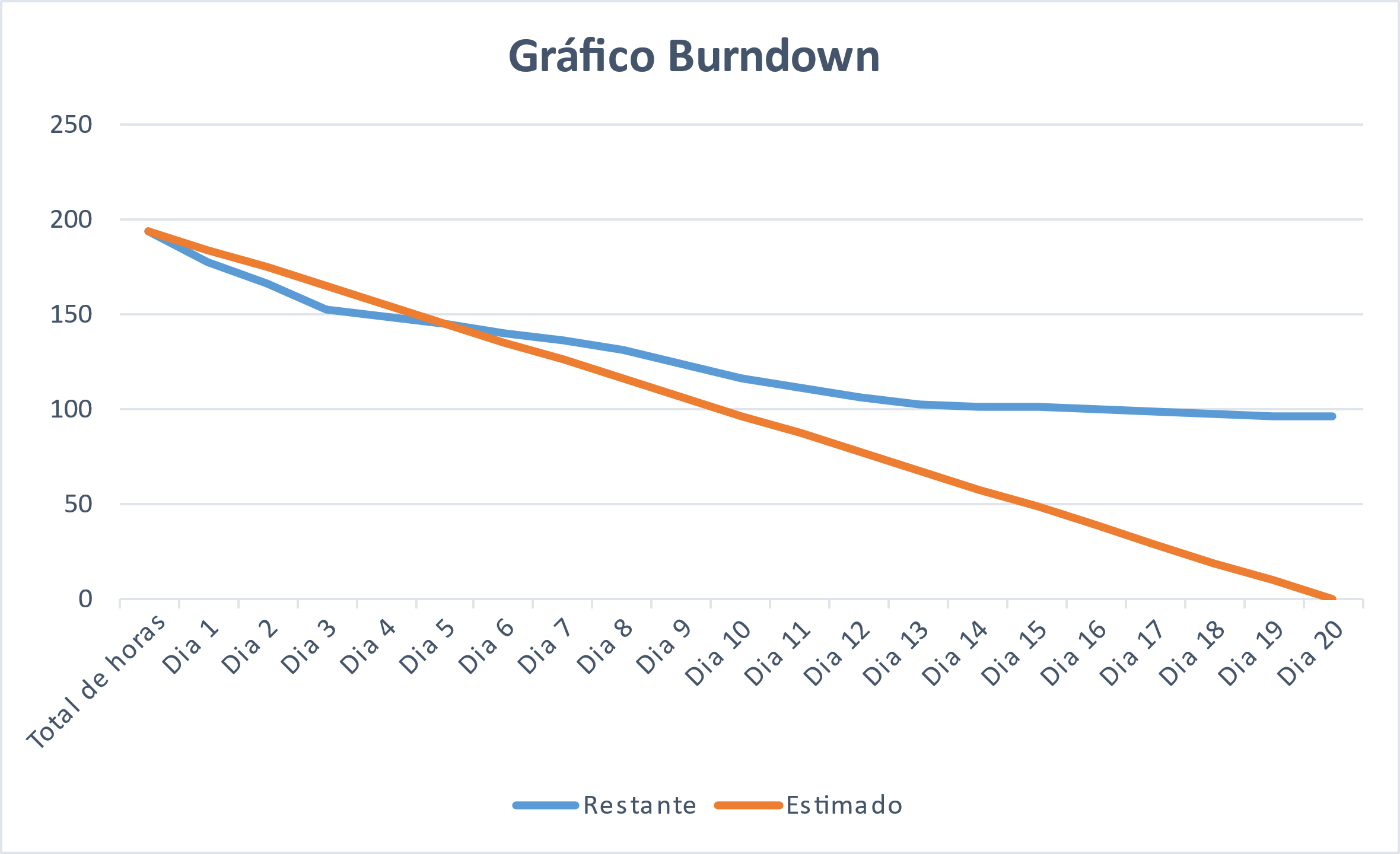
Diante do atraso na entrega das telas, os integrantes, Thiago Aguiar e Camila Gomes, optaram por desenvolver o aplicativo em uma outra plataforma. Ulteriormente, o encarregado da produção das telas, se juntou a equipe para finalizá-las.

# Sprint Backlog

Além disso, foi realizado nesse sprint:

* **Site:** 
  + **RF02** – Usuários em geral
    - **RF02.1 –** O site disponibilizará uma explicação detalhada do projeto e da equipe que está atuando no desenvolvimento do *software*
    - **RF02.2 –** O site permite que os usuários tirem dúvidas a respeito do projeto por meio de um *chatbot* e por meio do FAQ desenvolvido
    - **RF02.3** – O site permite com que usuários entre em contato com a equipe das mais variadas formas possíveis: redes sociais, e-mail e telefones das centrais de atendimento ao cliente

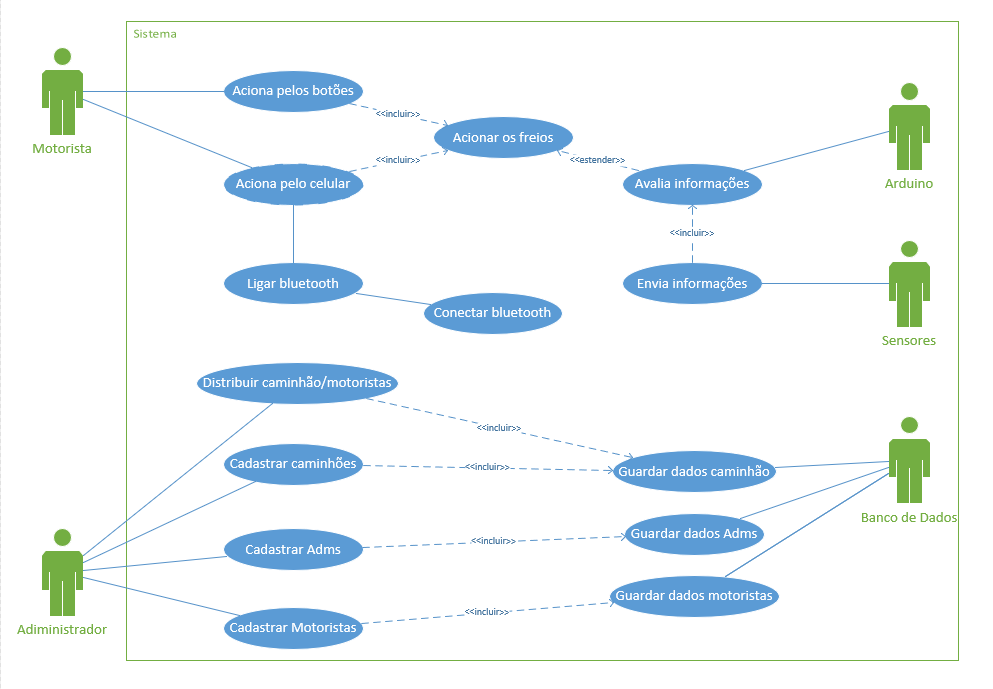
# Burn Down Chart



*Imagem 3 - Burn Down Chart de setembro*

# Diagramas

Nesse sprint foi realizado o diagrama dos casos de usos:



*Imagem 4 – Diagrama do caso de uso*

|  |
| --- |
| **NARRATIVA DE CASO DE USO** |
| **Sigla/Nome**: Acionar pelos botões  **Objetivo**: Acionar o freio por meio dos botões  **Frequência estimada**: Alta  **Ator Principal**: Motorista |
| **CENÁRIO PRINCIPAL** |
| 1. O caso inicia quando o motorista aperta o botão verde que está no painel da cabine do caminhão 2. Isso gera o comando para acionar o freio |
| **CENÁRIO ALTERNATIVO** |
| 1. O caso alternativo começa se o motorista aperta o botão vermelho no painel da cabine do caminhão    * 1. Nesse caso o botão desativará o freio, caso ele esteja ativado      2. Caso o freio não esteja ativado não acontecerá nada |
| **CENÁRIOS DE EXCEÇÃO** |
| 1. Caso o botão quebre    * 1. Caso isso ocorra, o freio poderá ser ativado pelo aplicativo ou pelo próprio Arduino, mas mesmo assim é altamente recomendado que os botões sejam trocados assim que possível |
| **PRÉ-CONDIÇÕES** |
| 1. O caminhão estar em baixa velocidade, no início de sua decida |
| **PÓS-CONDIÇÕES** |
| 1. A parada do caminhão |

|  |
| --- |
| **NARRATIVA DE CASO DE USO** |
| **Sigla/Nome:**  Acionar pelo celular  **Objetivo:**  Acionar o freio pelo celular  **Frequência estimada:**  Mediana  **Ator Principal:** Motorista |
| **CENÁRIO PRINCIPAL** |
| 1. O motorista abrirá o aplicativo 2. Nesse aplicativo há um botão que permitirá que o freio seja ativado ou desativado 3. Quando o motorista clicar nele, ele mudará o status do freio de ativado para desativado (vice-versa) 4. Passando essa mudança para o Arduino para que o freio seja acionado (Vide Acionar os freios) |
| **CENÁRIO ALTERNATIVO** |
| - |
| **CENÁRIOS DE EXCEÇÃO** |
| 1. Bluetooth não está conectado    * 1. O aplicativo mostrará que não é possível realizar a operação e que é necessário se conectar novamente com o freio         1. Caso o problema persista, o aplicativo não permitirá a mudança de status por ele, mas ainda será possível que ela seja feita por outros meios, como por meio dos botões (Vide Acionar pelos botões), ou pelo próprio Arduino (Vide Avalia informações) |
| **PRÉ-CONDIÇÕES** |
| 1. Que o bluetooth esteja ligado (Vide Ligar o bluetooth) 2. Que o bluetooth esteja conectado com o freio designado para ele pelo administrador (Vide Conectar o bluetooth) |
| **PÓS-CONDIÇÕES** |
| 1. O freio mudará de status, permitindo com que se tenha uma maior visibilidade da situação. Assim, não será necessário que o motorista saia do caminhão para saber em qual situação o freio se encontra |

|  |
| --- |
| **NARRATIVA DE CASO DE USO** |
| **Sigla/Nome:**  Ligar o bluetooth  **Objetivo:**  Ativar o bluetooth do celular para que ele consiga conectar com o freio  **Frequência estimada:**  Mediana  **Ator Principal:** Motorista |
| **CENÁRIO PRINCIPAL** |
| 1. O bluetooth será ativado após a permissão do motorista pelo aplicativo 2. Com isso, o bluetooth será ligado e poderá, então, ser conectado ao freio (Vide Conectar o bluetooth) |
| **CENÁRIO ALTERNATIVO** |
| - |
| **CENÁRIOS DE EXCEÇÃO** |
| 1. Aplicativo não tenha a permissão para o bluetooth    * 1. Caso o aplicativo não tenha obtido a permissão, será requisitado novamente, pois sem essa permissão o aplicativo não terá como mudar o status do freio, e o motorista será impedido de fazer qualquer mudança no aplicativo para que isso não venha afetar o funcionamento do sistema. |
| **PRÉ-CONDIÇÕES** |
| 1. Um celular que tenha o hardware que permite a conexão bluetooth |
| **PÓS-CONDIÇÕES** |
| 1. A permissão para que o aplicativo possa se conectar com o freio |

|  |
| --- |
| **NARRATIVA DE CASO DE USO** |
| **Sigla/Nome:** Conectar o bluetooth  **Objetivo:** Realizar a conexão do bluetooth no celular e no freio  **Frequência estimada:** Mediana  **Ator Principal:** Motorista |
| **CENÁRIO PRINCIPAL** |
| 1. Com a permissão obtida e o bluetooth ligado (Vide Ligar o bluetooth) 2. Ele será conectado automaticamente com o freio do caminhão que foi designado para ele, não sendo necessária nada além de sua permissão |
| **CENÁRIO ALTERNATIVO** |
| - |
| **CENÁRIOS DE EXCEÇÃO** |
| 1. Freio não encontrado    * 1. Caso isso ocorra, o aplicativo exibirá uma mensagem na tela explicando que o freio designado não está por perto e impedirá que o motorista faça alguma mudança no status do freio, a fim de evitar que o funcionamento do sistema seja danificado.      2. O sistema incentivará o usuário a se mover para mais perto do freio, até que ele se conecte com ele |
| **PRÉ-CONDIÇÕES** |
| 1. O bluetooth estar ligado e ter as permissões necessárias |
| **PÓS-CONDIÇÕES** |
| 1. O freio será conectado e poderá ser utilizado |

|  |
| --- |
| **NARRATIVA DE CASO DE USO** |
| **Sigla/Nome:** Envia informações  **Objetivo:** Enviar as informações dos sensores para o Arduino  **Frequência estimada:** Alta  **Ator Principal:** Sensores |
| **CENÁRIO PRINCIPAL** |
| 1. Os sensores coletarão dados do ambiente 2. Eles os enviarão para o Arduino para que eles sejam analisados (Vide Avalia informações) |
| **CENÁRIO ALTERNATIVO** |
| - |
| **CENÁRIOS DE EXCEÇÃO** |
| 1. Caso os sensores parem de funcionar    * 1. Os freios podem ser acionados pelos outros dois métodos, mas não será possível que eles se acionem sozinhos, já que não haverá as informações necessárias para o Arduino fazer a verificação |
| **PRÉ-CONDIÇÕES** |
| 1. Os sensores devem estar funcionando corretamente 2. Os sensores devem estar conectados com o Arduino |
| **PÓS-CONDIÇÕES** |
| 1. O Arduino receberá as informações necessárias para avaliar se o freio deverá ser puxado automaticamente, ou se ele deve permanecer na sua forma de repouso. |

|  |
| --- |
| **NARRATIVA DE CASO DE USO** |
| **Sigla/Nome:** Avalia informações  **Objetivo:** Avaliar as informações recebidas dos sensores e avaliar se o freio deve ser puxado ou não  **Frequência estimada:** Alta  **Ator Principal:** Arduino |
| **CENÁRIO PRINCIPAL** |
| 1. O Arduino receberá as informações dos sensores 2. Ele avaliará esses sinais, eles podem ser classificados como:    * + 1. Normal        2. Anormal 3. Caso seja classificado como normal ele não acionará o freio, e ele permanecerá no seu estado de repouso 4. O Arduino recebe o comando de ativar ou de desativar o freio. Nesse caso o Arduino executará o comando que ele recebeu |
| **CENÁRIO ALTERNATIVO** |
| 1. Caso os sinais enviados dos sensores sejam classificados como “Anormal” 2. Nesse caso o freio será acionado, fazendo com que o caminhão pare e impedindo que ele colida com outros veículos, pessoas e imóveis |
| **CENÁRIOS DE EXCEÇÃO** |
| 1. Caso o Arduino erre em sua avaliação e não acione o freio    * 1. Os freios ainda podem ser acionados pela cabine e pelo aplicativo 2. Caso o Arduino erre em sua avaliação e acione o freio quando não é necessário    * 1. Nesse caso o motorista será obrigado a desativar o freio pela cabine ou aplicativo |
| **PRÉ-CONDIÇÕES** |
| 1. Os sensores devem passar as informações para o Arduino fazer a verificação delas |
| **PÓS-CONDIÇÕES** |
| 1. O freio será acionado automaticamente 2. O freio responderá ao que o motorista comandou pelo aplicativo |

|  |
| --- |
| **NARRATIVA DE CASO DE USO** |
| **Sigla/Nome:** Acionar os freios  **Objetivo:** Acionar os freios de emergência do caminhão para fazer uma parada de emergência  **Frequência estimada:** Mediana  **Ator Principal:** Sistema |
| **CENÁRIO PRINCIPAL** |
| 1. O sistema recebe o comando de acionamento pelos botões (Vide aciona pelos botões) 2. Nesse caso, o freio é ativado de acordo com o comando do motorista, seja ele para ativá-lo como desativá-lo. |
| **CENÁRIO ALTERNATIVO** |
| 1. O sistema recebe o comando de acionamento pelo celular (Vide aciona pelo celular). 2. Nesse caso, o freio é ativado de acordo com o comando do motorista, seja ele para ativá-lo como desativá-lo. 3. O sistema recebe o comando de acionamento por meio do Arduino (Vide avalia informações). 4. Nesse caso, o freio será ativado se a resposta proveniente da avaliação dos sensores for positiva |
| **CENÁRIOS DE EXCEÇÃO** |
| - |
| **PRÉ-CONDIÇÕES** |
| 1. Receber a permissão dos três meios específicos para acionar ou não os freios |
| **PÓS-CONDIÇÕES** |
| 1. Execução da ordem realizada |

|  |
| --- |
| **NARRATIVA DE CASO DE USO** |
| **Sigla/Nome:** Cadastrar Motoristas  **Objetivo:** Cadastraros motoristas dos caminhões  **Frequência estimada:** Alta  **Ator Principal:**  Administradores |
| **CENÁRIO PRINCIPAL** |
| 1. Primeiramente, será necessário a realização do login do administrador para acessar o *dashboard* 2. Preencher os dados corretamente referentes ao usuário 3. Nome 4. Credencial 5. Foto de Perfil 6. E-mail 7. Senha 8. Administrador (necessário estar desativado) 9. Status na empresa 10. O administrador precisa clicar no botão de “Enviar Dados” para finalizar o cadastro |
| **CENÁRIO ALTERNATIVO** |
| - |
| **CENÁRIOS DE EXCEÇÃO** |
| 1. Caso o administrador deixar o campo “Administrador” ativado, será necessário deletar o usuário do banco e realizar a ação novamente. |
| **PRÉ-CONDIÇÕES** |
| 1. Inserir os dados corretamente em seus respectivos campos |
| **PÓS-CONDIÇÕES** |
| 1. Administrador cadastrou um motorista com sucesso |

|  |
| --- |
| **NARRATIVA DE CASO DE USO** |
| **Sigla/Nome:** Cadastrar Administradores (Adm)  **Objetivo:** Cadastrar Adm para queeles possam ter acesso ao portal e possam cadastrar os motoristas, adm, caminhões e atribuir um caminhão a um motorista  **Frequência estimada:** Mediana  **Ator Principal:** Administradores |
| **CENÁRIO PRINCIPAL** |
| 1. Primeiramente, será necessário a realização do login do administrador para acessar o *dashboard* 2. Preencher os dados corretamente referentes ao usuário 3. Nome 4. Credencial 5. Foto de Perfil 6. E-mail 7. Senha 8. Administrador (necessário estar ativado) 9. Status na empresa 10. O administrador precisa clicar no botão de “Enviar Dados” para finalizar o cadastro |
| **CENÁRIO ALTERNATIVO** |
| - |
| **CENÁRIOS DE EXCEÇÃO** |
| Caso o administrador deixar o campo “Administrador” desativado, será necessário editar os dados do usuário do banco e realizar a ação novamente. |
| **PRÉ-CONDIÇÕES** |
| 1. Inserir os dados corretamente em seus respectivos campos |
| **PÓS-CONDIÇÕES** |
| 1. Administrador cadastrado com sucesso |

|  |
| --- |
| **NARRATIVA DE CASO DE USO** |
| **Sigla/Nome:** Cadastrar caminhões  **Objetivo:** Cadastrar caminhões para que eles possam ser atribuídos aos motoristas e conectados no freio com o bluetooth do celular  **Frequência estimada:** Mediana  **Ator Principal:** Administradores |
| **CENÁRIO PRINCIPAL** |
| 1. Primeiramente, será necessário a realização do login do administrador para acessar o *dashboard* 2. Preencher os dados corretamente referentes ao usuário 3. Placa 4. Modelo 5. Marca 6. O administrador precisa clicar no botão de “Enviar” para finalizar o cadastro do veículo |
| **CENÁRIO ALTERNATIVO** |
| - |
| **CENÁRIOS DE EXCEÇÃO** |
| - |
| **PRÉ-CONDIÇÕES** |
| 1. Inserir os dados corretamente em seus respectivos campos |
| **PÓS-CONDIÇÕES** |
| 1. Caminhão cadastrado com sucesso |

|  |
| --- |
| **NARRATIVA DE CASO DE USO** |
| **Sigla/Nome:** Distribuir caminhões/motoristas  **Objetivo:** Distribuiros caminhões disponíveis para os motoristas disponíveis  **Frequência estimada:** Alta  **Ator Principal:** Administradores |
| **CENÁRIO PRINCIPAL** |
| 1. A pessoa selecionará, dentre as opções disponíveis, os caminhões disponíveis. 2. Na página de alteração de dados do caminhão, será possível designar o veículo para um dos funcionários disponíveis. |
| **CENÁRIO ALTERNATIVO** |
| - |
| **CENÁRIOS DE EXCEÇÃO** |
| - |
| **PRÉ-CONDIÇÕES** |
| 1. Selecionar o veículo correspondente; 2. Selecionar o funcionário correspondente. |
| **PÓS-CONDIÇÕES** |
| 1. Veículo designado corretamente ao funcionário. |

|  |
| --- |
| **NARRATIVA DE CASO DE USO** |
| **Sigla/Nome:** Guardar dados caminhão  **Objetivo:** Guardar os dados relacionado ao caminhão  **Frequência estimada:** Mediana  **Ator Principal:** Banco de dados |
| **CENÁRIO PRINCIPAL** |
| 1. Armazenamento de dados recebidos pelo formulário de cadastrado de veículos; 2. Utilizando o sistema CRUD (*Create, Read, Update, Delete),* funções do banco de dados, para armazenamento e requisição dos dados. |
| **CENÁRIO ALTERNATIVO** |
| - |
| **CENÁRIOS DE EXCEÇÃO** |
| 1. Problemas relacionados às funções do banco de dados o que inviabilizará o armazenamento de dados de forma correta e segura. |
| **PRÉ-CONDIÇÕES** |
| 1. Receber corretamente os dados de cadastro de caminhões pelo administrador. |
| **PÓS-CONDIÇÕES** |
| 1. Dados armazenados com sucesso |

|  |
| --- |
| **NARRATIVA DE CASO DE USO** |
| **Sigla/Nome:** Guardar dados administrador  **Objetivo:** Guardar os dados relacionados aos administradores  **Frequência estimada:** Mediana  **Ator Principal:** Banco de dados |
| **CENÁRIO PRINCIPAL** |
| 1. Armazenamento de dados inseridos pelo administrador no formulário de cadastro de usuários; 2. Selecionar a opção “Administrador” para que o funcionário possa acessar suas funções administrativas. 3. Utilizando o sistema CRUD (*Create, Read, Update, Delete),* funções do banco de dados, para armazenamento e requisição dos dados. |
| **CENÁRIO ALTERNATIVO** |
| - |
| **CENÁRIOS DE EXCEÇÃO** |
| 1. Problemas relacionados às funções do banco de dados o que inviabilizará o armazenamento de dados de forma correta e segura. |
| **PRÉ-CONDIÇÕES** |
| 1. Receber corretamente os dados de cadastro de caminhões pelo administrador. |
| **PÓS-CONDIÇÕES** |
| 1. Dados armazenados com sucesso |

|  |
| --- |
| **NARRATIVA DE CASO DE USO** |
| **Sigla/Nome:** Guardar dados motoristas  **Objetivo:** Guardar os dados relacionados com os motoristas dos caminhões  **Frequência estimada:** Mediana  **Ator Principal:** Banco de dados |
| **CENÁRIO PRINCIPAL** |
| 1. Armazenamento de dados inseridos pelo administrador no formulário de cadastro de usuários; 2. A opção “Administrador” não poderá ser ativada; 3. Utilizando o sistema CRUD (*Create, Read, Update, Delete),* funções do banco de dados, para armazenamento e requisição dos dados. |
| **CENÁRIO ALTERNATIVO** |
| - |
| **CENÁRIOS DE EXCEÇÃO** |
| 1. Problemas relacionados às funções do banco de dados o que inviabilizará o armazenamento de dados de forma correta e segura |
| **PRÉ-CONDIÇÕES** |
| 1. Receber corretamente os dados de cadastro de caminhões pelo administrador. |
| **PÓS-CONDIÇÕES** |
| 1. Dados armazenados com sucesso |

*Tabelas 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18 – Tabelas sobre casos de uso sobre a funcionalidade do sistema*

# Plano de testes

Não foram realizados planos de teste nesse *Sprint*.

# Resultados

Nesse *Sprint,* não houve resultados dos planos de testes.

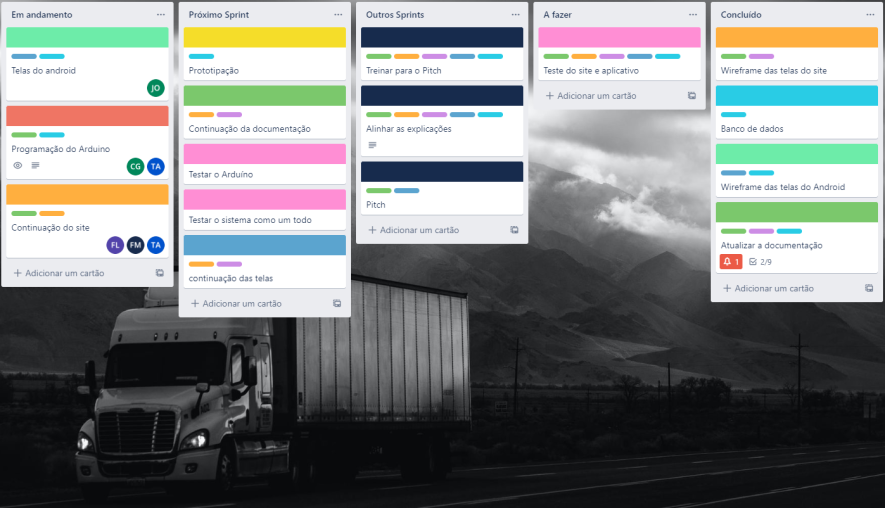
# 

# Kanban e Retrospectiva

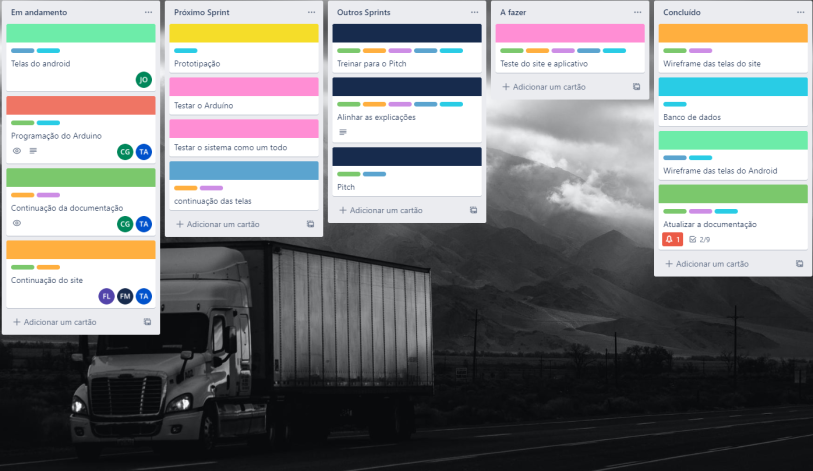
*Imagem 5 – Trello da 1ª Semana de Setembro*

Interface gráfica do usuário, Site

Descrição gerada automaticamente*Imagem 6 – Trello da 2ª Semana de Setembro*

**

*Imagem 7 – Trello da 3ª Semana de Setembro*

*Imagem 8 – Trello da 4ª Semana de Setembro*

Como mostrado anteriormente, nesse sprint fizemos boa parte do site e mesmo com o eventual atraso da aplicação Android conseguimos, com a ajuda da equipe, manter o progresso do projeto.

# Terceiro Sprint (Mês de Outubro)

Nesse Sprint decidimos que três integrantes trabalhariam juntos para completar o Android, como a utilização da tecnologia anteriormente selecionada demandaria uma grande quantidade de tempo para que os alunos as aprendessem e implementassem no projeto. Portanto decidimos que o grupo focaria na parte de Android com o intuito de terminar essa parte do projeto, mas que a parte do arduino também seria usada para fazer testes de funcionamento do Android.

# Product Backlog

Como salientado anteriormente voltamos ao plano original de usar o Android para a realização do aplicativo Android.

# Sprint Backlog

# Desenvolvimento Mobile

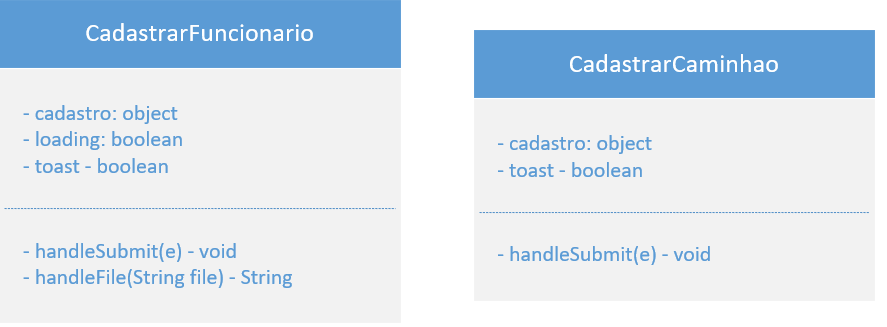
* + **RF01 -** O aplicativo fará o login dos funcionários que foram cadastrados no banco pelo site
  + **RF02 -** O aplicativo permitirá que os funcionários que o acessarem ativem/desativem o freio do caminhão que eles usarão para a viajem
  + **RF03 -** O aplicativo conectará apenas ao Bluetooth do caminhão, caso ele não esteja disponível ele mostrará que não foi possível encontrar o freio
  + **RF04 -** O aplicativo mostrará o status do freio (Ativado ou desativado) no aplicativo
  + **RF05 -** O aplicativo permitirá que caso a senha seja esquecida, ela possa ser recuperada pelo e-mail cadastrado no banco
  + **RF06 -** O aplicativo permitirá que o funcionário possa mudar a senha caso ele deseje fazer isso
* **Arduíno**
* **RF03 -** O Arduino ativará o freio quando for comandado, ou quando for decidido que é necessário
* **RF05 -** O Arduino receberá a ordem de desativar/ativar o freio do bluetooth do celular
* **RF06 -** O Arduino passará o status do freio para o celular pelo bluetooth
* **RF07 -** O Arduino ativará o freio quando os botões que estão presentes na cabine forem pressionados

# Burn Down Chart

*Imagem 9 – Gráfico Burndown de outubro*

# Diagramas

Nesse Sprint foi feito o seguinte diagrama de classes:



*Imagem 10 – Diagrama de classe das classes CadastrarFuncionario e CadastrarCaminhao*

# Plano de testes

|  |  |
| --- | --- |
| **Plano de testes - 01** | |
| **Funcionalidade: Cadastro de Funcionário** | |
| **Passos** | **Ações** |
| 1 | Cadastrar um funcionário sem a foto |
| 2 | Cadastrar um funcionário sem a credencial |
| 3 | Cadastrar um funcionário sem a senha |
| 5 | Cadastrar um usuário sem um e-mail |
| 6 | Cadastrar um funcionário sem um e-mail válido |
| 7 | Cadastrar um funcionário sem um nome |
| **Expectativa** | |
| O cadastro não será efetuado | |
| **Data** | |
| 17/10/2022 | |
| **Criador** | |
| **Camila** | |

|  |  |
| --- | --- |
| **Plano de testes - 02** | |
| **Funcionalidade: Cadastro de Caminhões** | |
| **Passos** | **Ações** |
| 1 | Cadastrar um funcionário sem a marca |
| 2 | Cadastrar um funcionário sem o modelo |
| 3 | Cadastrar um funcionário sem a placa |
| 5 | Cadastrar um usuário sem bluetooth |
| **Expectativa** | |
| O cadastro não será efetuado | |
| **Data** | |
| 17/10/2022 | |
| **Criador** | |
| **Camila** | |

*Tabelas 19, 20 – Tabelas dos planos de testes para cadastros de Caminhões e Funcionários*

# Resultados

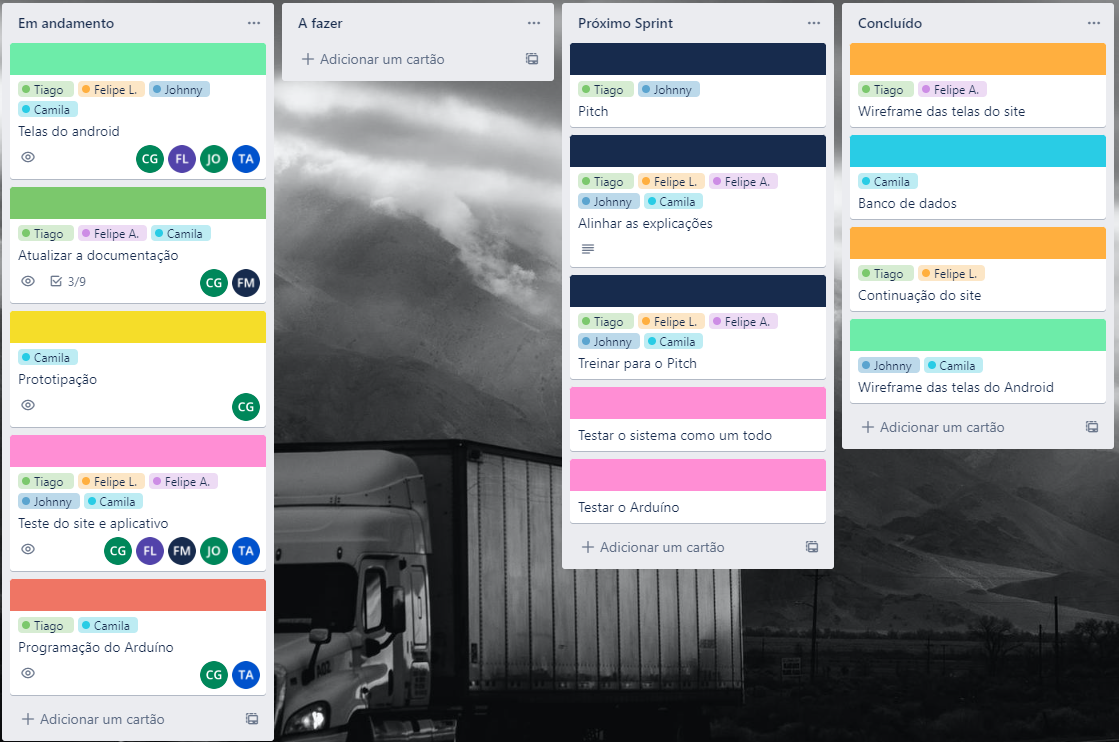
|  |  |  |  |
| --- | --- | --- | --- |
| **Resultado do teste - 01** | | | |
| **Funcionalidade: Cadastro de Funcionário** | | | |
| **Teste** | **Executor** | **Data** | **Resultado** |
| 1 | Tiago | 17/10/2022 | Sucesso |
| 2 | Tiago | 17/10/2022 | Sucesso |
| 3 | Tiago | 17/10/2022 | Sucesso |
| 5 | Tiago | 17/10/2022 | Sucesso |
| 6 | Tiago | 17/10/2022 | Sucesso |
| 7 | Tiago | 17/10/2022 | Sucesso |

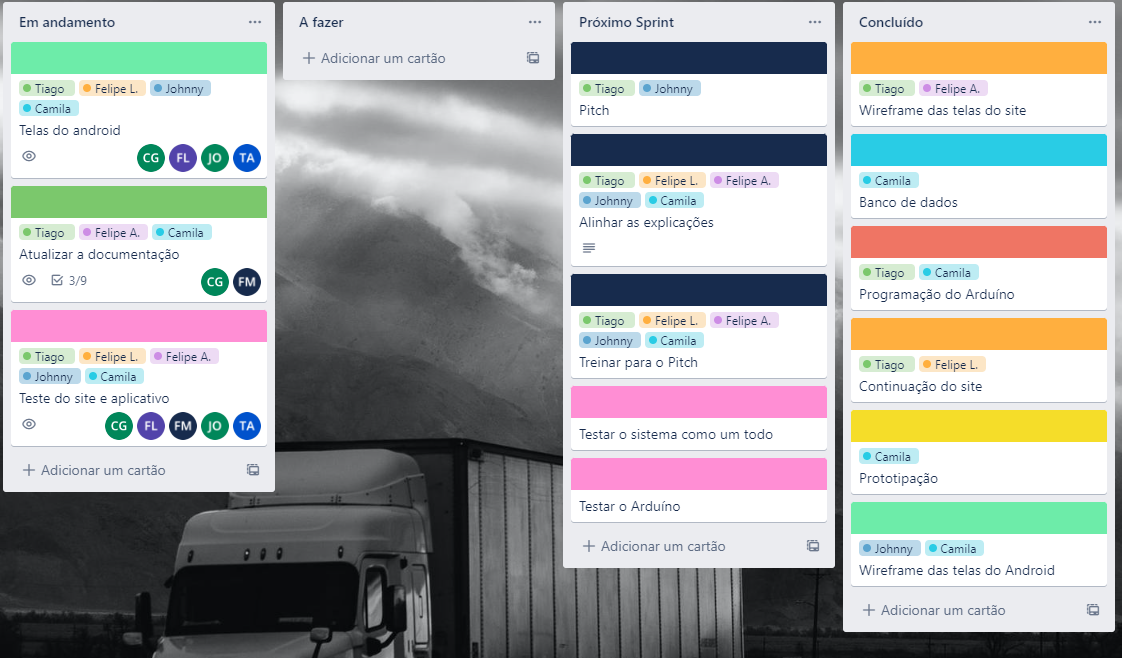
|  |  |  |  |
| --- | --- | --- | --- |
| **Resultado do teste - 02** | | | |
| **Funcionalidade: Cadastro de Caminhões** | | | |
| **Teste** | **Executor** | **Data** | **Resultado** |
| 1 | Tiago | 17/10/2022 | Sucesso |
| 2 | Tiago | 17/10/2022 | Sucesso |
| 3 | Tiago | 17/10/2022 | Sucesso |
| 5 | Tiago | 17/10/2022 | Sucesso |

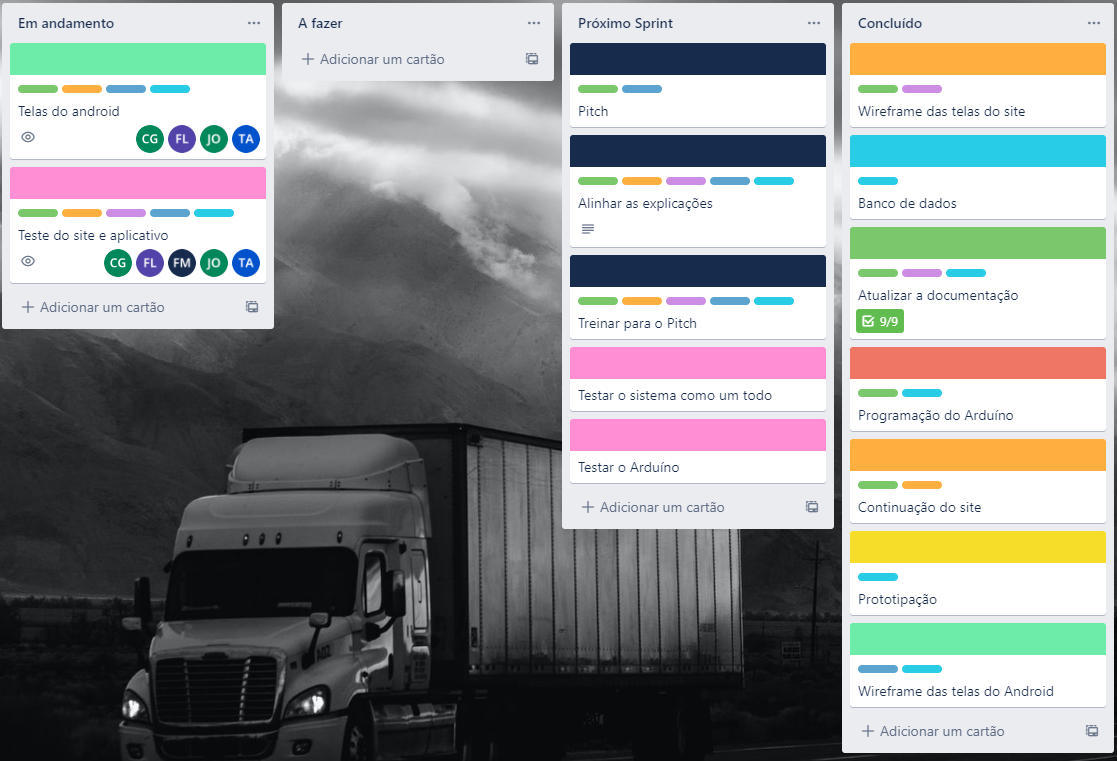
*Tabelas 20, 21 – Tabelas dos resultados dos testes 01 e 02*

# Kanban e Retrospectiva

*Imagem 11 – Trello da 1ª Semana de Outubro*

*Imagem 12 – Trello da 2ª Semana de Outubro*

*Imagem 13 – Trello da 3ª Semana de Outubro*

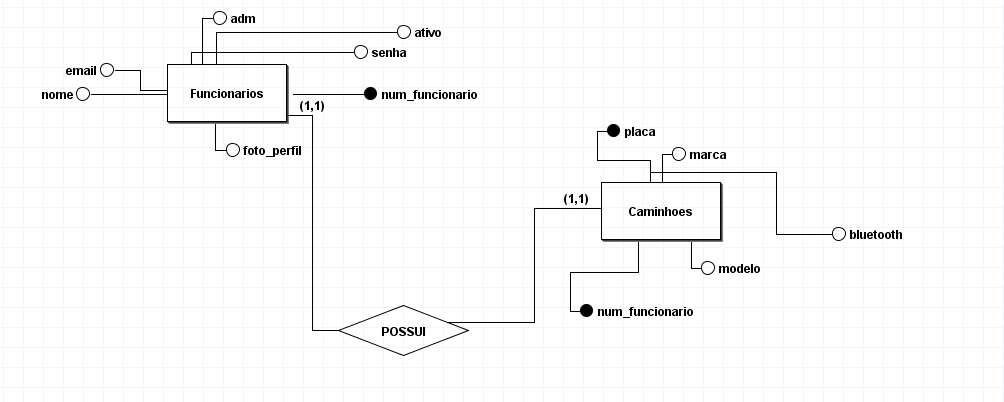
*Imagem 14 – Trello da 4° Semana de Outubro*

Nesse sprint conseguimos concluir uma boa parte do planejado para esse sprint , embora tenha ocorrido alguns imprevistos, como a falta de internet e palestras que atrapalharam o nosso desenvolvimento nesse mês, ainda conseguimos completar boa parte do conteúdo proposto pela equipe.

# Modelo de Dados

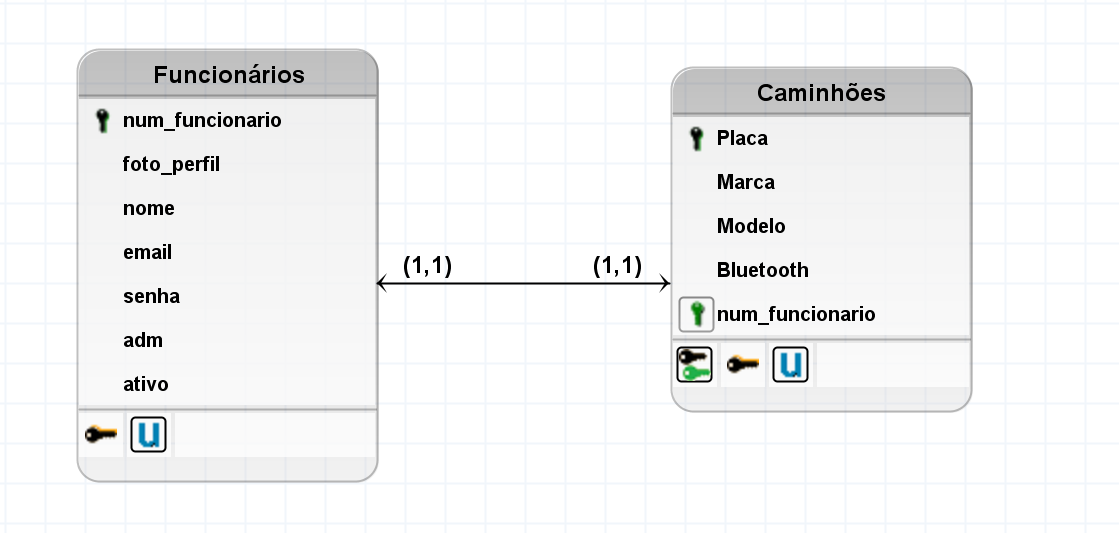
Nós optamos pelo uso do banco MySQL, uma vez que precisamos de um banco capaz de armazenar dados de forma relacionada. Por isso decidimos que a melhor opção no nosso caso era usar esse tipo de armazenamento, também usamos o SQLite para otimizar o login do usuário no aplicativo, já que evitaria que o usuário fosse obrigado a fazer o login toda a vez que precisasse usar o aplicativo.

# Diagrama de Entidade e Relacionamento



*Imagem 15 – Diagrama de entidade e relacionamento do banco de dados usado no projeto*

# Modelo lógico do banco de dados



*Imagem 16 – Modelo lógico do banco de dados*

# Dicionário de dados

|  |  |  |
| --- | --- | --- |
| **Dicionário de dados - Funcionários** | | |
| **Atributos** | **Tipos de dados** | **Descrição** |
| num\_funcionario | String | O número usado na empresa para identificar um funcionário específico |
| foto\_perfil | LongText | Campo usado para guardar imagens convertidas em base 64 |
| Nome | String | Guarda o nome da pessoa |
| Email | String | Guarda o email do funcionário |
| senha | String | Guarda a senha do funcionário |
| adm | Bool | Guarda se o funcionário é um administrador (verdadeiro), ou se ele é apenas um usuário comum (falso) |
| ativo | Bool | Grava se um usuário não foi desligado da empresa, o valor inicial dele é verdadeiro, e quando o funcionário for desligado o valor passa a ser falso e ele perde o acesso ao aplicativo/site |

|  |  |  |
| --- | --- | --- |
| **Dicionário de dados - Caminhões** | | |
| **Atributos** | **Tipos de dados** | **Descrição** |
| placa | String | Armazena a placa do veículo |
| marca | String | Usado para guardar a marca do veículo |
| modelo | String | Guarda o modelo do veículo |
| bluetooth | String | Guarda o nome do bluetooth do veículo |
| ativo | Bool | Grava se um veículo não foi desligado da empresa, o valor inicial dele é verdadeiro, e quando o veículo for desligado o valor passa a ser falso e ele perde o acesso ao aplicativo/site |

*Tabelas 22, 23 - Dicionário de dados do banco de dados*

# PRINCIPAIS TELAS DO SISTEMA

Descreve de maneira simples as principais telas do sistema

# CONCLUSÃO

# Escreva os resultados obtidos

Resultados obtidos

# Constatações

Constatações

# Sugestões de possíveis aperfeiçoamentos técnicos

Sugestões

# REFERÊNCIAS

TRIBUNA DE MINAS. **Caminhão não consegue subir via íngreme, desce de ré e quase acerta residência**. Disponível em: <<https://tribunademinas.com.br/noticias/cidade/13-10-2017/caminhao-carregado-de-cimento-desde-de-re-rua-antonio-dias.html>>. Acesso em: 26 set. 2022.

LADEIRA. **Carreta não consegue subir ladeira e causa acidente em Itatiba; vídeo**. Disponível em: <<https://g1.globo.com/sp/sorocaba-jundiai/noticia/2019/10/25/carreta-nao-consegue-subir-ladeira-e-causa-acidente-em-itatiba-video.ghtml>>. Acesso em: 26 set. 2022.

**Associação Brasileira dos Caminhoneiros - Home**. Disponível em: <[http://www.abcam.org.br/index.php/pt/>.](http://www.abcam.org.br/index.php/pt/%3E.) Acesso em: 26 set. 2022.

‌

‌

# GLOSSÁRIO

Se houver necessidade

# ANEXOS

Se houver necessidade