



# Segurança no Trânsito – Grupo 09

Trabalho Interdisciplinar - Aplicações Web

**Integrantes do grupo:**

**Kaiky França da Silva**

**Carlos Eduardo Nascimento**

**Paula Maria**

Belo Horizonte, Julho/2023

### **Contexto (Problema, Objetivos, Público-alvo, Justificativa, Artefatos de Design Thinking):**

O problema da imprudência no trânsito é uma preocupação global que afeta não apenas o Brasil, mas também outros países ao redor do mundo. Dados da Organização Mundial da Saúde (OMS) mostram que os acidentes de trânsito são uma das principais causas de morte em todo o mundo, especialmente entre jovens de 15 a 29 anos.

A falta de atenção e o comportamento irresponsável ao volante são fatores que contribuem significativamente para esses acidentes. Segundo estudos, distrações ao dirigir, como o uso de celulares, são responsáveis por uma parcela considerável dos acidentes. Além disso, o desrespeito às leis de trânsito, o excesso de velocidade e a falta de uso do cinto de segurança são outros fatores que contribuem para a ocorrência de acidentes graves.

No contexto do projeto Steero, o foco inicial era desenvolver um aplicativo móvel que oferecesse recursos de automação e controle da velocidade do veículo. A intenção era abranger diversas funcionalidades para ajudar os motoristas a terem uma condução mais segura e consciente.

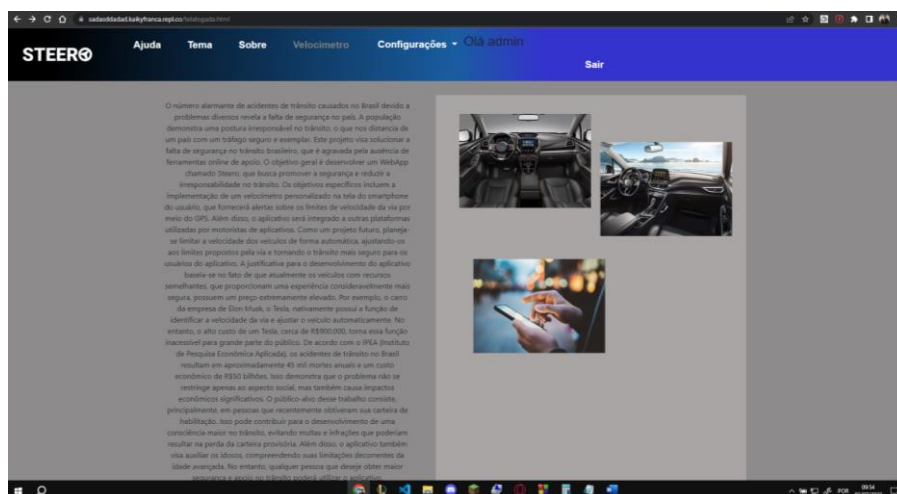
No entanto, devido a limitações e mudanças na equipe de desenvolvimento, algumas ideias tiveram que ser ajustadas e adaptadas ao longo do processo. Como resultado, o projeto evoluiu para um WebAPP que utiliza diversas APIs, incluindo as do Google, para aprimorar a experiência do usuário e oferecer funcionalidades que promovam a segurança no trânsito.

É importante ressaltar que a conscientização sobre os riscos no trânsito e a adoção de medidas de segurança são fundamentais para reduzir os acidentes. Além de tecnologias como o Steero, é essencial que haja uma mudança cultural, com a valorização de uma condução responsável e o respeito às leis de trânsito em todas as faixas etárias.

### **Projeto da Solução (Requisitos, projeto de interface/wireframes/protótipo, tecnologias):**

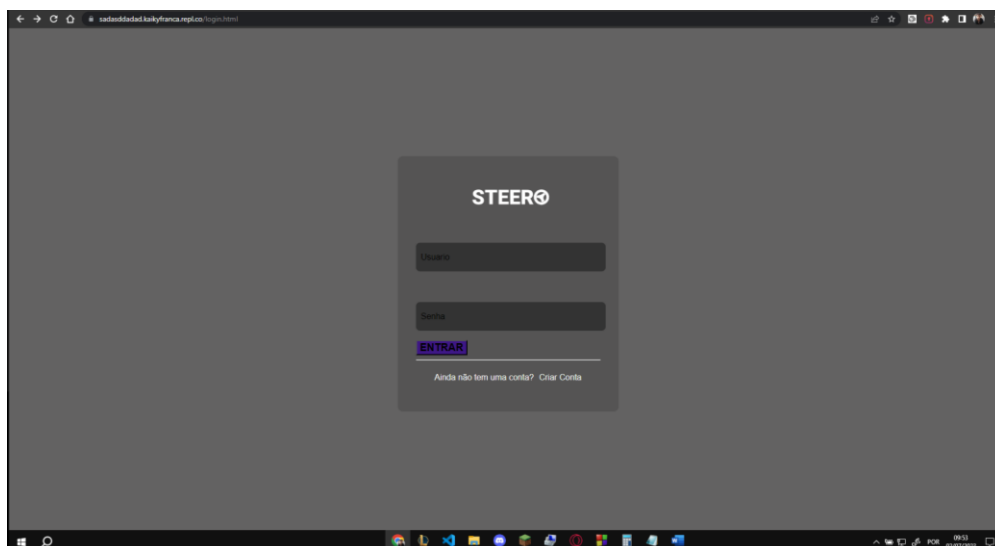
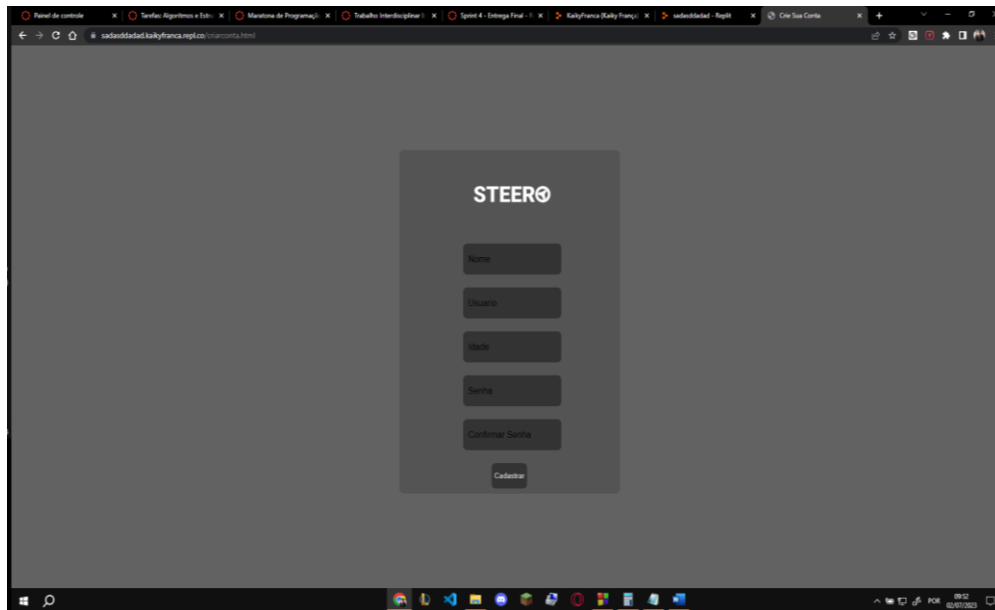
# Tela inicial

Principal com informações sobre o projeto STEERO, o usuário cadastrado e a funcionalidade principal, os demais botões apenas por design.



# Tela de login e Cadastro:

Telas de cadastro e login bem limpas e simples utilizando da funcionalidade LocalStorage para fazer a funcionalidade de login e senha.



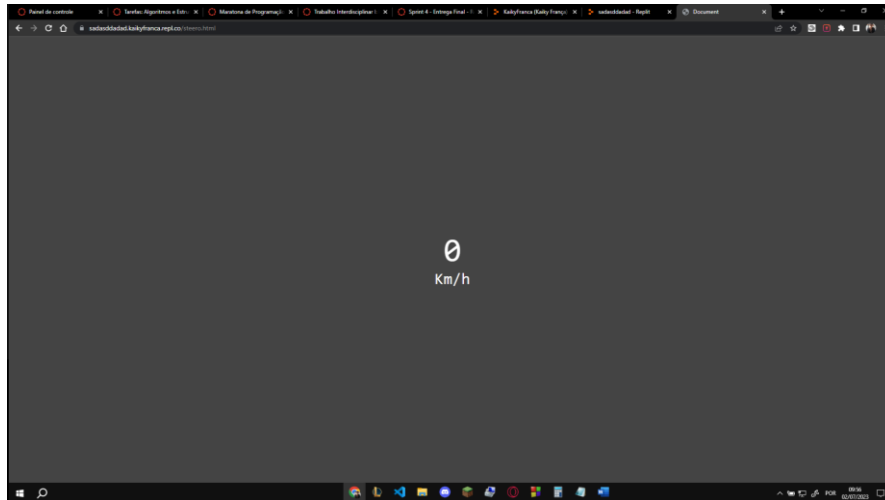
## Velocímetro:

A principal funcionalidade vem quando se clica em Velocímetro, onde o usuário é redirecionado para uma tela que marca sua principal velocidade utilizando seu GPS como um velocímetro de carro.

Ele utiliza da funcionalidade da GEOLocation API do google, que é uma API de localização do google MAPS que pelo GPS consegue medir a velocidade do usuário e ele transcreve isso para a tela.

Com a funcionalidade da HERE API também consegue identificar parcialmente a velocidade das rodovias presentes na rua, e quando ele percebe que está a cima da

velocidade indicada pela rodovia que ele está localizando, o app dá um alerta ficando com a tela vermelha.



## Requisitos Atendidos:

Colocamos alguns dos requisitos propostos como metas para serem atendidas, dentre eles, destaque aos:

RF-01 – Apresentar a proposta do APP em um site na WEB

RF-02 – Apresentação Geral do site

RF-03 – O site deve permitir uma descrição completa do grupo e seus integrantes

RNF-01 – Site em plataformas de compartilhamento (GitHub e Replit)

## Links para site e repositório:

### Site:

<https://steero-ti-front-end.kaikyfranca.repl.co/>

### Repositório GitHub:

<https://github.com/ICEI-PUC-Minas-PMGCC-TI/ti-1-pmg-cc-m-20231-tiaw-g09-seguranca-no-transito/tree/master>

# Códigos utilizados:

Utiliza da localização fornecida pela API e mede a velocidade do usuário imprimindo na tela, e também identifica as velocidades das vias para dizer se está a cima ou abaixo da velocidade recomendada.

```
3
4
5
6 const velocimetro = document.querySelector("#velocimetro")
7 const velocidade = document.querySelector("#velocidade")
8
9 let watchID = null
10 const option = {enableHighAccuracy:true}
11
12 velocimetro.addEventListener("click",()=>{
13     console.log(watchID)
14     if(!watchID){
15         watchID = navigator.geolocation.watchPosition(
16             updatePosition, handleError, option)
17     }
18     else{
19         navigator.geolocation.clearWatch(watchID)
20         watchID = null
21         velocidade.textContent = 0
22     }
23 })
24
25 //navigator.geolocation.watchPosition(
26 // updatePosition, handleError)
27
28 function updatePosition(position){
29     //console.log(position.coords.speed * 3.6)
30     if(position.coords.speed === null)
31         return
32     velocidade.textContent = (position.coords.speed * 3.6).toFixed(1)
33 }
34
35 var console: Console
36 func (error: Error){
37     console.log(error.message)
38 }
39
40
```

Json que diz a API que tipo de sinal utilizar para definir localização, no caso ela utiliza de GPS e dos sinais de internet do dispositivo.

```
kyFranca/Steero-TI-Front-End#steero.html
Run

js x login.html x telalogada.js x telalogada.html x steero.js x STEEROOOO.png x steero.html x
steero.html
19
20
21 <div id="velocidade">0</div>
22
23 <div>Km/h</div>
24
25 </div>
26
27
28
29 <script>
30
31 const apiKey = 'AIzaSyAYpUuS350IuiBWILuepeN0xJHVnt4Hf_o';
32
33 const request = {
34   "considerIp": true
35 };
36
37 const requestOptions = {
38   method: 'POST',
39   headers: {
40     'Content-Type': 'application/json',
41   },
42   body: JSON.stringify(request)
43 };
44
45 fetch('https://www.googleapis.com/geolocation/v1/geolocate?key=${apiKey}', requestOptions)
46   .then(response => response.json())
47   .then(data => {
48     console.log(data);
49     // Manipule a resposta da API Geolocation aqui
50   })
51   .catch(error => {
52     console.error('Erro:', error);
53   });
54 </script>
55
56 <script
57   src="https://www.googleapis.com/geolocation/v1/geolocate?key=AIzaSyAYpUuS350IuiBWILuepeN0xJHVnt4Hf_o">
58 </script>
59
60 <script src="steero.js">
61
62 </script>
63
64 </body>
65
66 </html>
```

## JavaScript da funcionalidade de Registro e Login:

A LocalStorage, funcionalidade padrão do JavaScript registra as informações que o usuário cadastra diretamente no site dentro de um Json que fica armazenado na memória do Site na Web como uma espécie de banco de dados porém temporário, logo após ele pega o registro e valida ele juntamente com a tela de Login.



```
login.js x criar.js x login.html x telalogada.js x telalogada.html x steero.js x STEERO
login.js > f entrar > f listaUser.forEach() callback > ...

function entrar() {
  let usuario = document.querySelector('#login');
  let userLabel = document.querySelector('#userLabel');

  let senha = document.querySelector('#senha');
  let senhaLabel = document.querySelector('#senhaLabel');

  let msgError = document.querySelector('#msgError');
  let listaUser = [];

  let userValid = {
    nome: '',
    user: '',
    senha: ''
  };

  listaUser = JSON.parse(localStorage.getItem('listaUser'));
  console.log(listaUser);

  if (listaUser) {
    listaUser.forEach((item) => {
      if (usuario.value == item.userCad && senha.value == item.senhaCad) {
        userValid = {
          nome: item.nomeCad,
          user: item.userCad,
          senha: item.senhaCad
        };
      }
    });
  }

  if (usuario.value == userValid.user && senha.value == userValid.senha) {
    window.location.href = 'telalogada.html'
    let token = Math.random().toString(16).substr(2) + Math.random().toString(16).substr(2)
    localStorage.setItem('token', token)
    localStorage.setItem('userLogado', JSON.stringify(userValid))
  } else {
    alert('Tente novamente');
  }

  console.log('Botão clicado')
}
```



```
login.js x criar.js x login.html x telalogada.js x telalogada.html x steero.js x STEEROOOO...
criar.js > labelconfirmSenha
1 //campos de validação
2
3 let nome = document.querySelector('#nome')
4 let labelNome = document.querySelector('#labelNome')
5 let validNome = false
6
7 let usuario = document.querySelector('#usuario')
8 let labelUsuario = document.querySelector('#labelUsuario')
9 let validUsuario = false
10
11 let idade = document.querySelector('#idade')
12 let labelIdade = document.querySelector('#labelIdade')
13 let validIdade = false
14
15 let senha = document.querySelector('#senha')
16 let labelSenha = document.querySelector('#labelSenha')
17 let validSenha = false
18
19 let confimsenha = document.querySelector('#confirmsenha')
20 let labelconfirmSenha = document.querySelector('#labelconfirmSenha')
21 let validconfirmSenha = false
22
23 let msgError = document.querySelector('#msgError')
24 let msgSuccess = document.querySelector('#msgSuccess')
25
26
27 nome.addEventListener('keyup',()=>{
28   if(nome.value.length <= 2){
29     labelNome.setAttribute('style','color: red')
30     labelNome.innerHTML = 'Minimo 3 caracteres'
31     nome.setAttribute('style','color: red')
32     validNome = false
33   }else {
34     labelNome.setAttribute('style','color: green')
35     labelNome.innerHTML = 'Nome'
36     nome.setAttribute('style','color: green')
37     validNome = true
38   }
39 })
40
41 usuario.addEventListener('keyup',()=>{
42   if(usuario.value.length <= 1){
43     labelUsuario.setAttribute('style','color: red')
44     labelUsuario.innerHTML = 'Minimo 2 caracteres'
45     usuario.setAttribute('style','color: red')
46     validUsuario = false
47   }else {
48     labelUsuario.setAttribute('style','color: green')
49     labelUsuario.innerHTML = 'Usuario'
50     usuario.setAttribute('style','color: green')
51   }
52 })
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
Ln 20, Col 1
```

```
login.js x criar.js x login.html x telalogada.js x telalogada.html x steero.js x STEEROOOO.p...
telalogada.js > ...
1 let userLogado = JSON.parse(localStorage.getItem('userLogado'));
2
3 let Logado = document.querySelector('#Logado');
4 Logado.innerHTML = 'Olá ' + userLogado.nome;
5
6 if (localStorage.getItem('token') == null) {
7   alert('Você precisa estar logado');
8 }
9
10 function sair() {
11   localStorage.removeItem('token');
12   window.location.href = 'login.html';
13 }
14
```

## **Metodologia de Trabalho (processo, atribuições, ferramentas):**

Entre os processos para criação do trabalho, pode-se destacar primeiramente a pesquisa intensa feita sobre as API's disponíveis e qual seria a melhor utilizável e aplicada para o APP.

Depois uma maior ênfase em JavaScript para melhoria da criação do Aplicativo que foi essencial.

Das ferramentas utilizadas são praticamente um WebSite feito em HTML e estilizado com CSS e Bootstrap, nas funcionalidades são utilizadas JavaScript com JSON e a GEOLocation API do Google e a HERE API.

## **Solução Implementada (funcionalidades de software)**

Conseguimos chegar em um resultado consistente com o velocímetro que marca muito bem a velocidade do carro e está o tempo todo funcionando, ainda estamos tentando implementar uma solução para que ele funcione sem precisar de internet coisa que é extremamente possível.

Para utilizar o dispositivo, basta entrar no site disponível no link:

<https://steero-ti-front-end.kaikyfranca.repl.co/>

Procurar pelo botão “Velocímetro” e a partir dele entrar, tocar uma vez na tela para iniciar, caso seja a primeira vez utilizando, tem que dar autorização para o acesso da localização, após isso é só aumentar a velocidade com o dispositivo que ele começa a funcionar.

Caso queira que ele pare por alguns instantes, basta dar outro toque na tela que imediatamente para de medir a velocidade e zera o velocímetro.

Na tela inicial para melhor experiência e recomendado criar um cadastro, clicando em “Login” e depois no botão para cadastrar seu Nome e Usuário.

(Por motivos de segurança da própria API ela não funciona sendo executada pelo Visual Studio Code usando LiveServer, apenas pelo link do Replit)

## **Testes de Software:**

Nos testes feitos individualmente pelos integrantes do grupo mostrou o velocímetro eficiente porém com um pouco de Delay no começo da movimentação dado por conta da API que leva menos de 1 segundo para estabilizar a verdadeira velocidade, a HERE

API que foi utilizada mostrou-se eficiente para o que promete, porém deixa a desejar quando o fato é estar a todo momento funcionando se mostrando fora do ar diversas vezes deixando algumas funcionalidades inutilizáveis, infelizmente não foram encontradas outras API que fizessem a mesma função que a HERE API que é encontrar informações das ruas que a pessoa está pelo GPS.

### **Planos de testes futuros e novas integrações:**

Para maiores testes futuros e definir melhores usabilidades utilizar em caminhadas e corridas, bicicletas e ver a maior precisão do velocímetro em relação a outros tipos de veículos que não sejam carros

Adicionar um sistema de notificações e produzir algo que consiga registrar e utilizar as velocidades das ruas para melhora do aplicativo e que ele se desprenda da HERE API.