

SEGURANÇA NO TRÂNSITO

Kaiky França da Silva, Carlos Eduardo Nascimento & Paula Maria

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

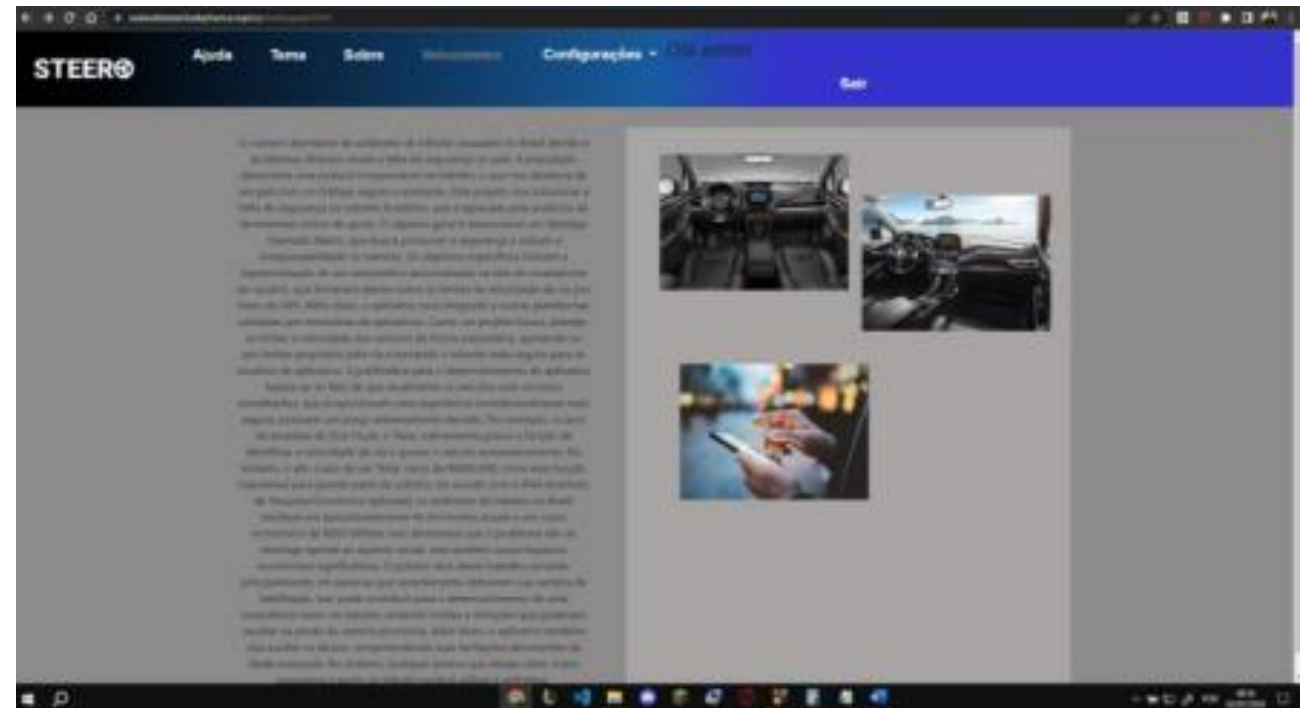
- O problema da imprudência no trânsito é uma preocupação global que afeta não apenas o Brasil, mas também outros países ao redor do mundo. Dados da Organização Mundial da Saúde (OMS) mostram que os acidentes de trânsito são uma das principais causas de morte em todo o mundo, especialmente entre jovens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



STEERO

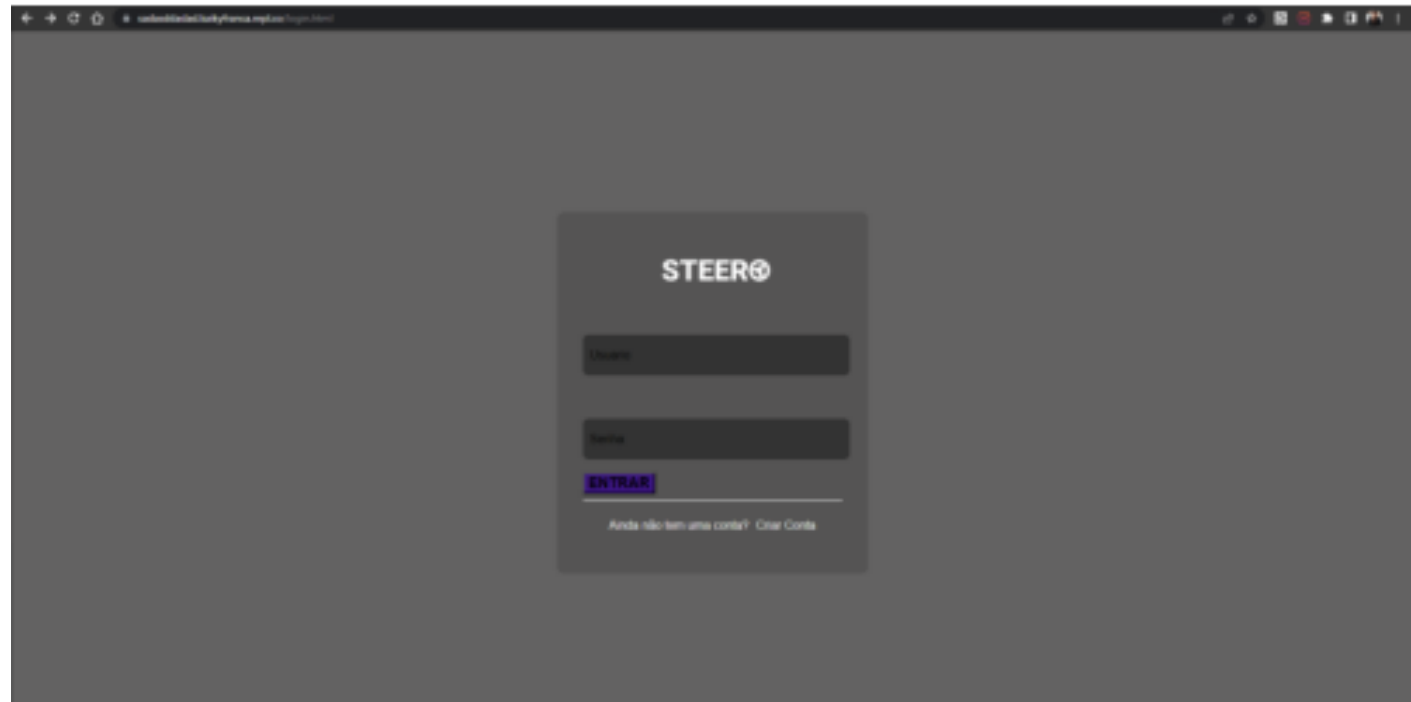
Tela inicial

- 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.



JavaScript da funcionalidade de Registro e Login:

```
login.js - entrar.js - login.html - telalogada.js - telalogada.html - steero.js - STEER
login.js - entrar.js - login.html - telalogada.js - telalogada.html - steero.js - STEER

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 listUser = [];

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

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

    if (listUser) {
        listUser.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 - entrar.js - login.html - telalogada.js - telalogada.html - steero.js - STEER
login.js - entrar.js - login.html - telalogada.js - telalogada.html - steero.js - STEER

function entrar() {
    let nome = document.querySelector('#nome');
    let userLabel = document.querySelector('#userLabel');
    let senha = document.querySelector('#senha');
    let senhaLabel = document.querySelector('#senhaLabel');
    let msgError = document.querySelector('#msgError');
    let listUser = [];

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

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

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

    if (nome.value == userValid.nome && userLabel.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 - entrar.js - login.html - telalogada.js - telalogada.html - steero.js - STEER
login.js - entrar.js - login.html - telalogada.js - telalogada.html - steero.js - STEER

function entrar() {
    let nome = document.querySelector('#nome');
    let userLabel = document.querySelector('#userLabel');
    let senha = document.querySelector('#senha');
    let senhaLabel = document.querySelector('#senhaLabel');
    let msgError = document.querySelector('#msgError');
    let listUser = [];

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

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

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

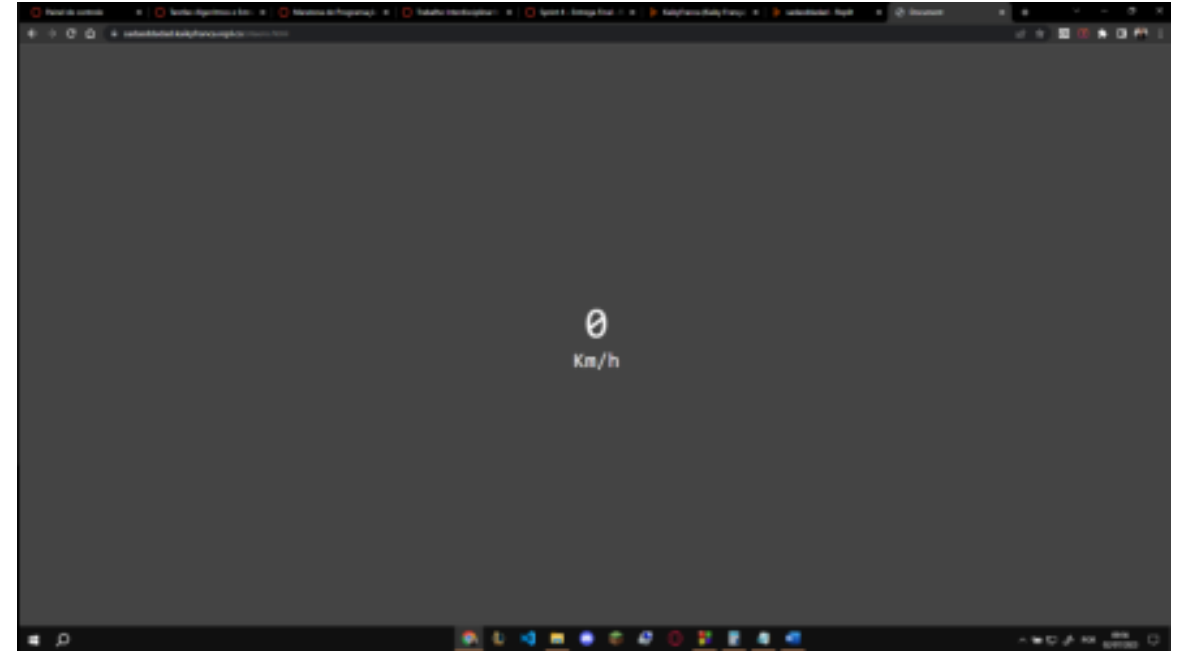
    if (nome.value == userValid.nome && userLabel.value == userValid.user && senha.value == userValid.senha) {
        window.location.href = 'telalogada.html';
        let token = Math.random().toString(16).substr(2) + Math.random().toString(16).substr(2);
        localStorage.setItem('token', token);
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

    console.log('Botão clicado');
}
```

- 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.

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.



- 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 acima ou abaixo da velocidade recomendada.
- 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.

```
const velocimetro = document.querySelector("#velocimetro")
const velocidade = document.querySelector("#velocidade")

let watchID = null
const option = {enableHighAccuracy:true}

velocimetro.addEventListener("click", ()=>{
  console.log(watchID)
  if(!watchID){
    watchID = navigator.geolocation.watchPosition(
      updatePosition, handleError, option)
  }
  else{
    navigator.geolocation.clearWatch(watchID)
    watchID = null
    velocidade.textContent = 0
  }
})

//navigator.geolocation.watchPosition(
//  updatePosition, handleError)

function updatePosition(position){
  console.log(position.coords.speed * 3.6)
  if(position.coords.speed === null){
    return
  }
  velocidade.textContent = (position.coords.speed * 3.6).toFixed(1)
}

var console = Console
function log(error){
  console.log(error.message)
}
```

[illegible]

Metodologia de Trabalho (processo, atribuições, ferramentas):

- A metodologia de trabalho adotada para a criação do aplicativo envolveu uma pesquisa intensa sobre as API's disponíveis, buscando a melhor opção para o projeto. Foi dada ênfase ao uso do JavaScript para aprimorar a criação do aplicativo. As ferramentas utilizadas incluem um website desenvolvido em HTML, estilizado com CSS e Bootstrap. As funcionalidades foram implementadas usando JavaScript, JSON e as APIs GEOLocation do Google e HERE.
- A solução implementada resultou em um velocímetro consistente, que marca com precisão a velocidade do carro e funciona continuamente. A equipe está buscando implementar uma solução para que o aplicativo funcione offline, o que é perfeitamente viável. Para utilizar o dispositivo, basta acessar o site pelo link fornecido e procurar pelo botão "Velocímetro". Ao tocar na tela, o velocímetro será iniciado, sendo necessário autorizar o acesso à localização. Para interromper a medição da velocidade, basta tocar novamente na tela, o que também reiniciará o velocímetro. Para uma melhor experiência, é recomendado criar um cadastro na tela inicial clicando em "Login" e em seguida no botão de cadastro. Vale ressaltar que a API não funciona quando executada pelo Visual Studio Code usando LiveServer, sendo necessário acessar o aplicativo pelo link do Replit por motivos de segurança.
- Para utilizar o dispositivo, basta entrar no site disponível no link: <https://steero-ti-front-end.kaikyfranca.repl.co/>

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.

