



INSTITUTO POLITÉCNICO
DE VIANA DO CASTELO

PROJETO#2

Tomás Pereira (30358) e Fábio Rocha (30240)



Instituto Politécnico de Viana do Castelo

**Escola Superior
de Tecnologia
e Gestão**

INSTITUTO POLITÉCNICO DE VIANA DO CASTELO
ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO

57, ERSC

2023/2024

2-A

Projeto#2

Unidade curricular: Programação Web

Discente (s): Tomás Machado Pereira e Fábio Sebastião da Rocha

Regente/orientador/Docente: António Lira Fernandes e Vasco Miranda

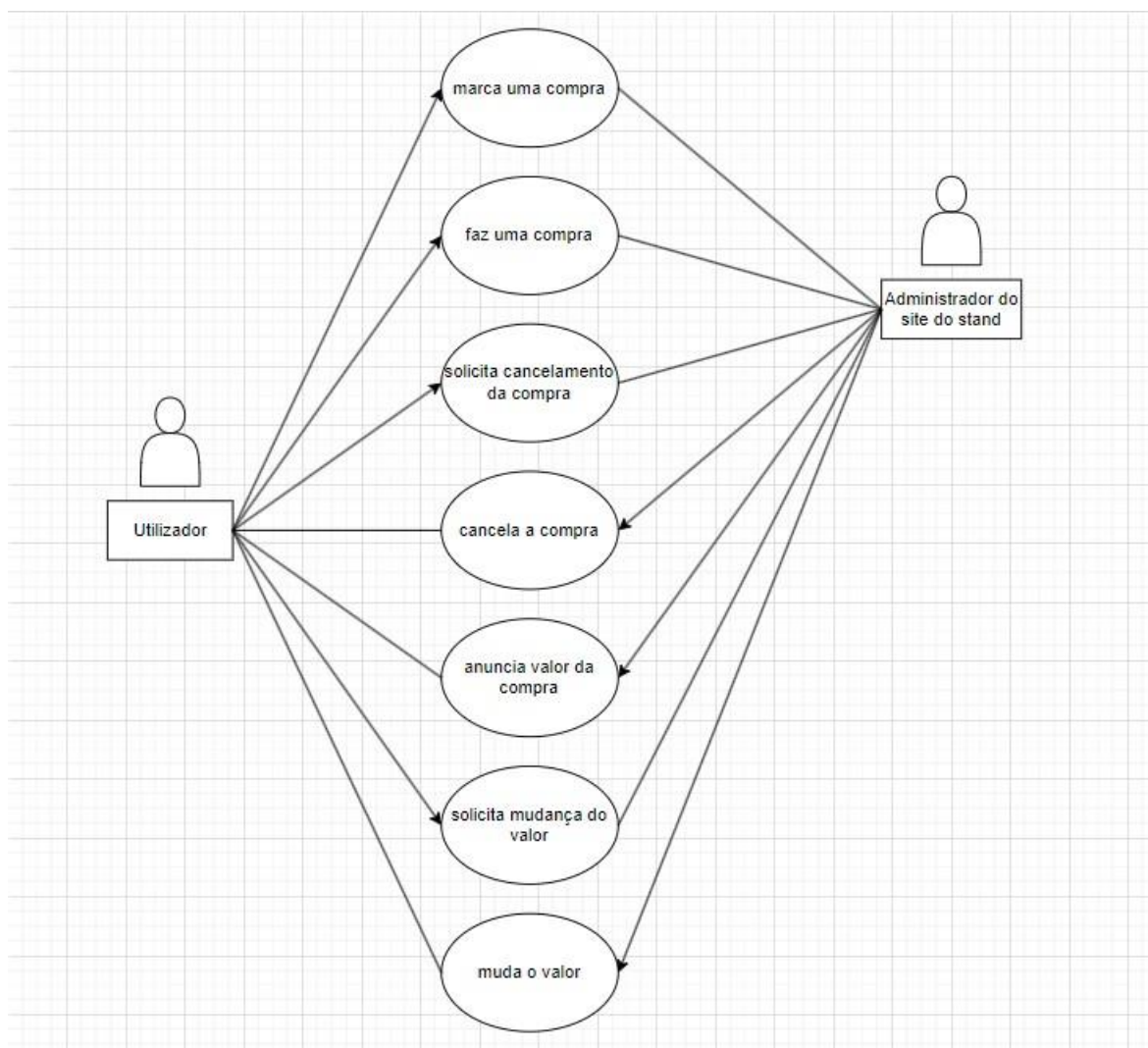
Viana do Castelo, junho, 2024

INTRODUÇÃO

Para a realização do projeto 1, na unidade curricular de Programação Web, foi-nos proposta a realização de um site funcional no qual teríamos de aplicar os conhecimentos adquiridos ao longo das aulas, tanto do professor António Lira como do professor Vasco Miranda. Para isso, decidimos recorrer a um stand virtual onde, para a conclusão com sucesso do mesmo, tentamos implementar todas os conceitos trabalhados e propostos para a realização deste trabalho.

Quanto ao relatório, este está dividido em várias partes. Começamos com a capa onde maior parte foi aproveitada pelo modelo disponibilizado no moodle pelo professor, depois fizemos uma introdução com uma breve explicação dos objetivos que tínhamos na realização deste trabalho. Logo depois temos a representação do desenho de casos de uso e a estrutura do site a explicar o que está representado em cada página do nosso site. Temos apresentação também, do desenho gráfico do nosso site e do nosso modelo relacional juntamente com o modelo E-R, já mais na parte final, temos as imagens do nosso site assim como o código utilizado no json e o código utilizado nalguns scripts. Para finalizar colocamos a imagem do nosso trello e a referência de alguns dos sites visitados para a resolução com sucesso do nosso trabalho.

DESENHO DE CASOS DE USO



ESTRUTURA DO SITE

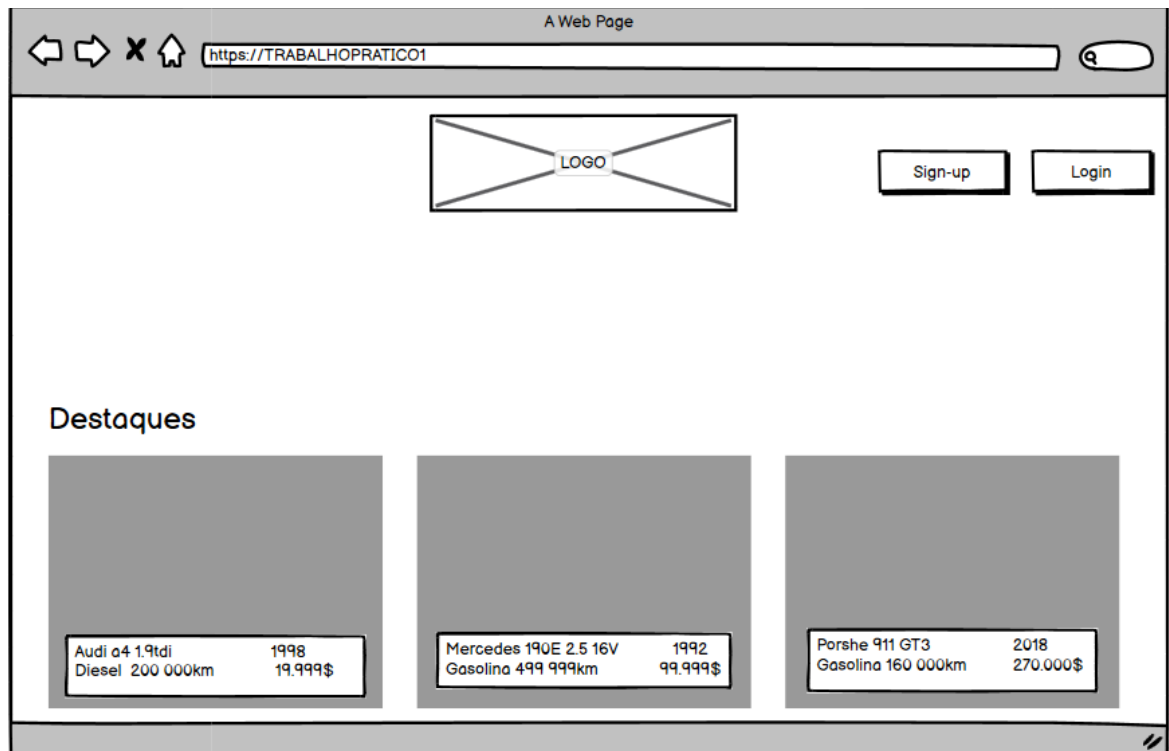
Ao entrar no nosso site vai ser possível encontrar uma página onde vai ter na parte de cima da página o logo da nossa empresa, por baixo temos os cards onde estão representados todos os carros que estão disponíveis no nosso site, sendo que ao clicar no carro é possível verificar as informações sobre o mesmo. No canto superior direito temos um botão para dar sign-up e um botão para fazer login que depois dá acesso a outra página.

Depois de dar sign-up ou login irá ser aberta outra página onde temos novamente o logo da nossa empresa no topo, tendo depois de um lado um botão back que nos permite voltar à página anterior e do lado direito temos dois botões. Um botão de gerir perfil que irá abrir outra página e um botão de gerir carros que ao seu pressionado encaminha-nos para outra página. Por baixo depois temos 4 botões (estatísticas, anúncios, mensagens e favoritos), onde ao clicarmos abre um modal ao lado onde será possível ver as suas estatísticas e etc.

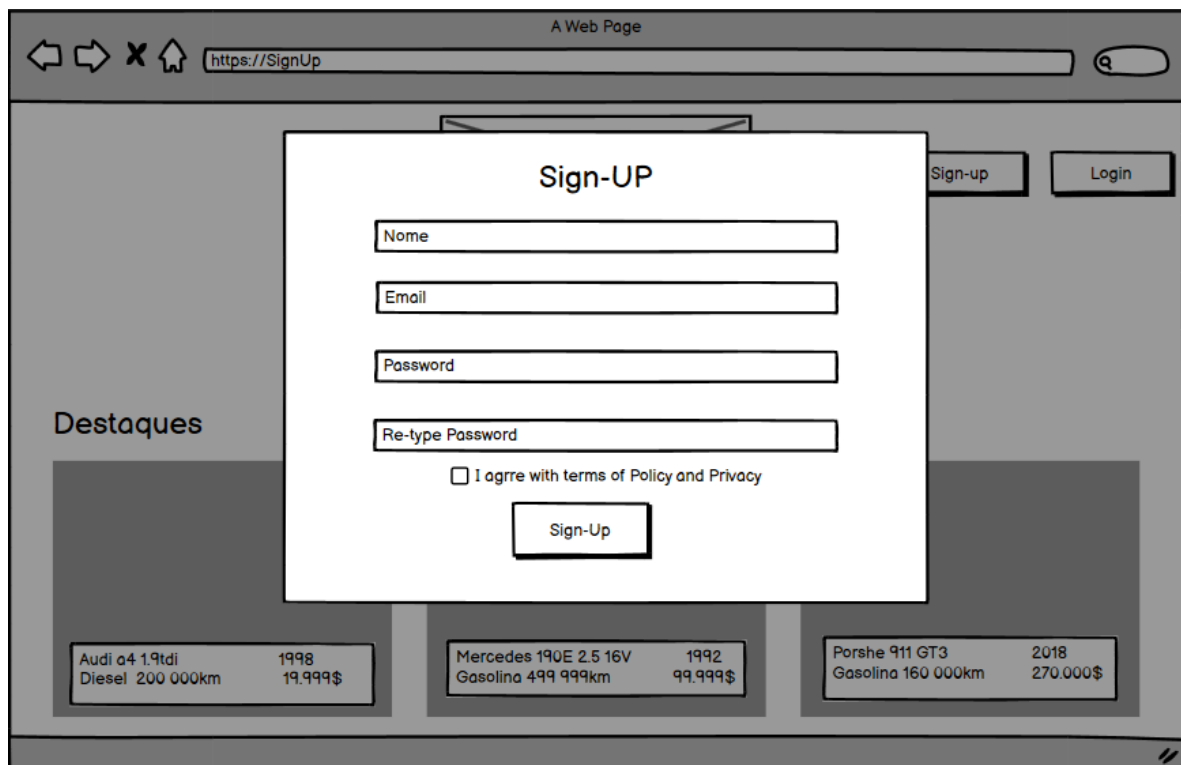
Ao pressionar o botão gerir perfil é encaminhado para uma página onde tem um botão back e a nossa foto de perfil no topo, tendo por baixo uma imagem que está a marcar o sítio onde se situaria a sua foto de perfil, tendo depois 4 campos, com o nome, o email, o país e a descrição. Já ao pressionar o botão gerir carros é encaminhado para uma página onde aparecem todos os carros presentes todos os carros presentes no nosso site e consequentemente todas as suas características e onde é possível também eliminar quaisquer carros, é possível também editar as características de cada carro e por fim é possível adicionar um carro. Apenas utilizadores com permissões administrativas é que tem acesso à secção gerir carros.

DESENHO DAS PÁGINAS

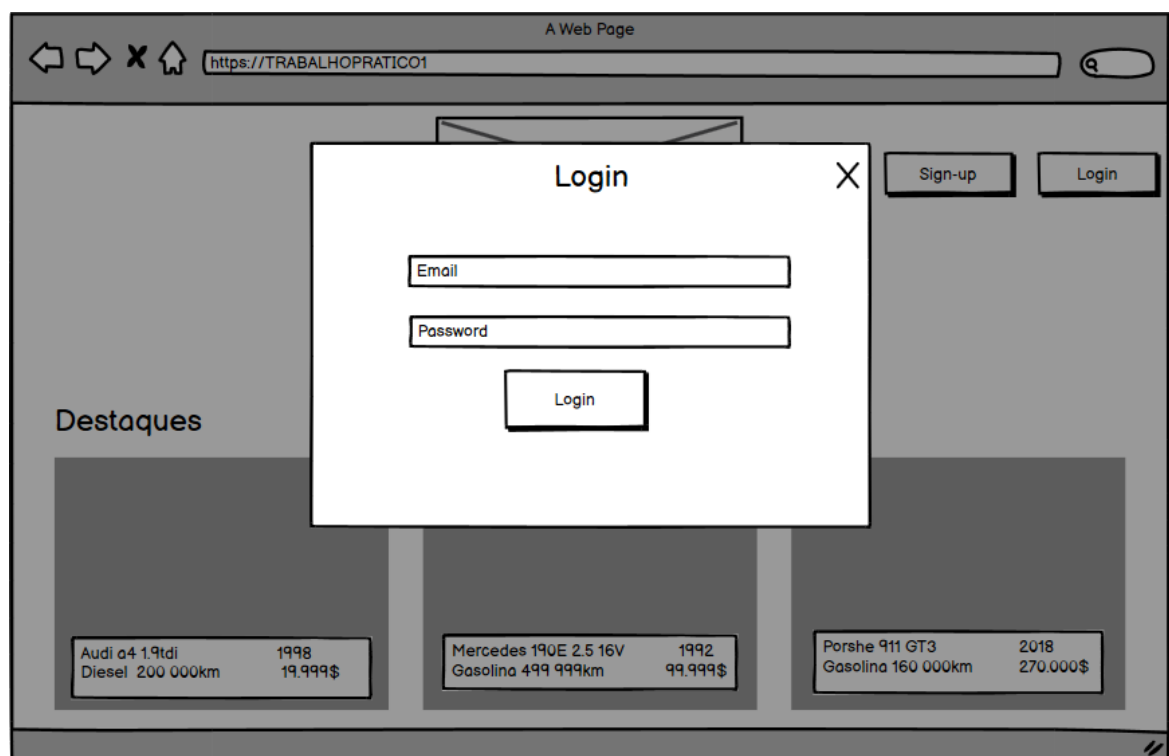
Para a realização do desenho gráfico das páginas do nosso site nós decidimos utilizar o balsamiq wireframes.



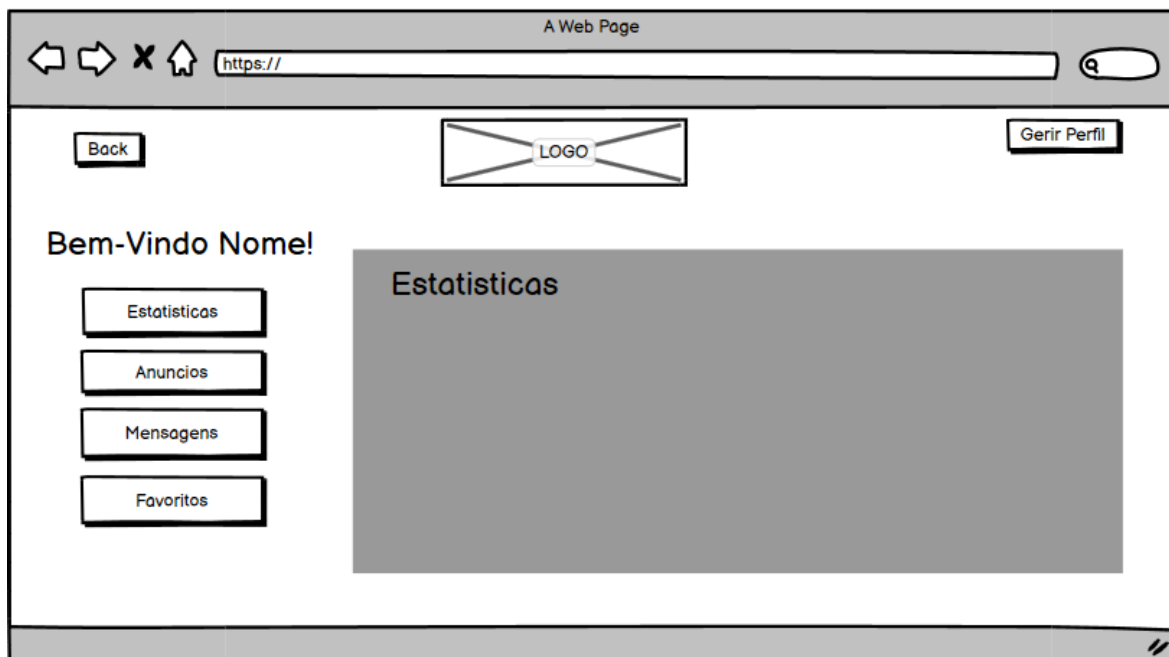
Representação página inicial



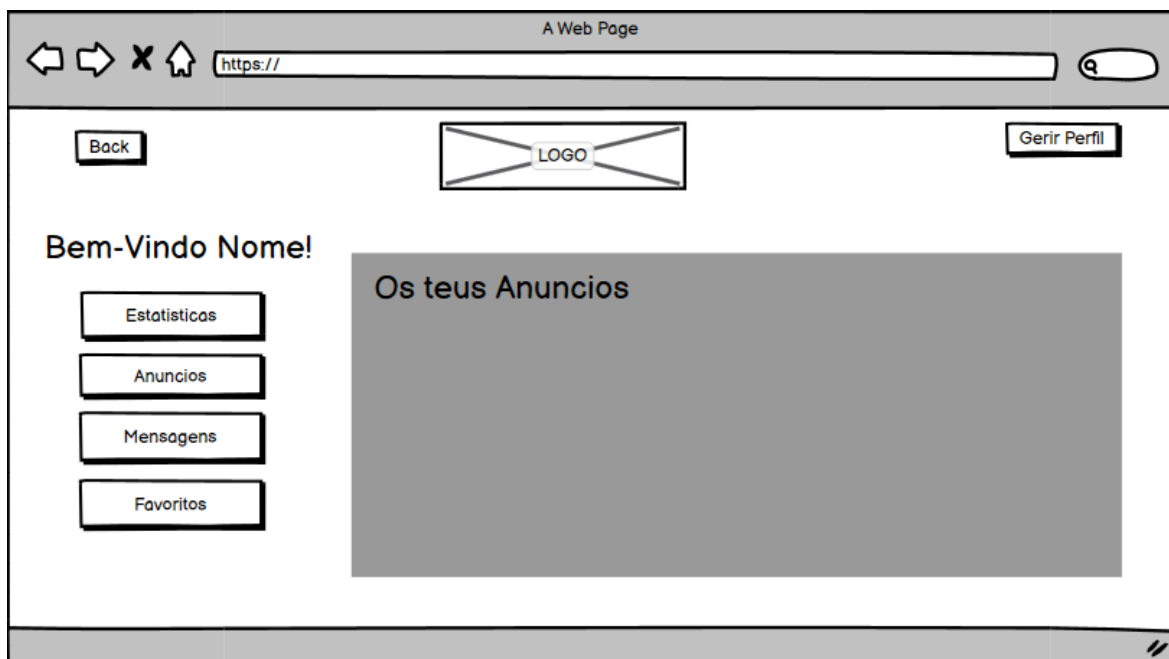
Representação depois de pressionar o botão sign-up



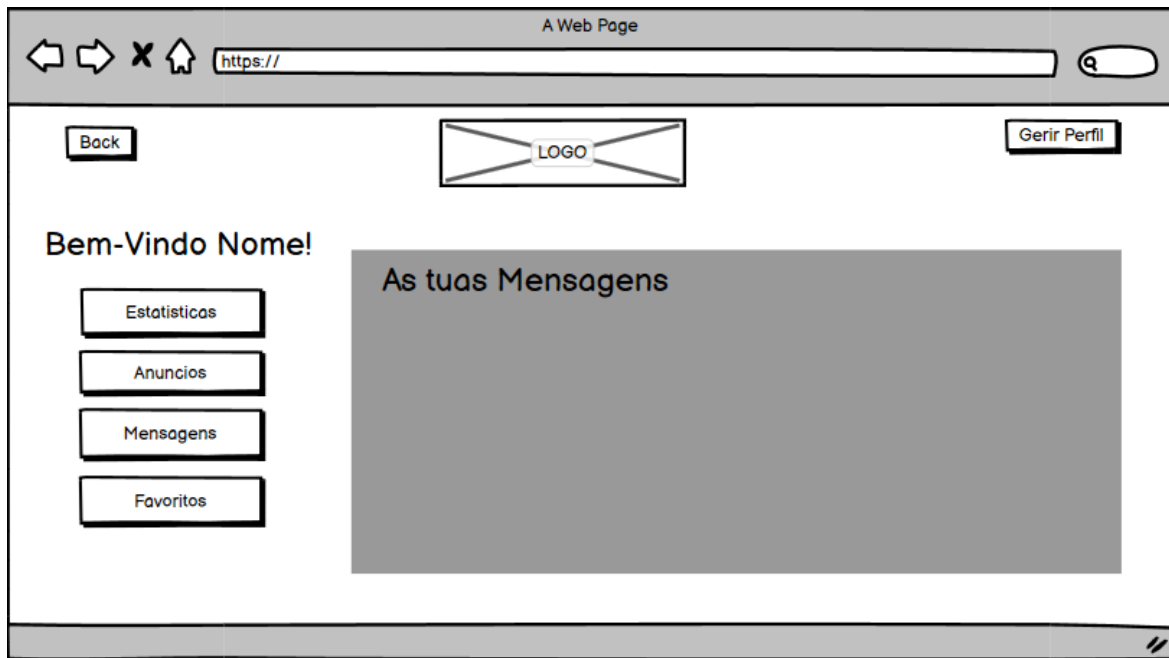
Representação depois de pressionar o botão login



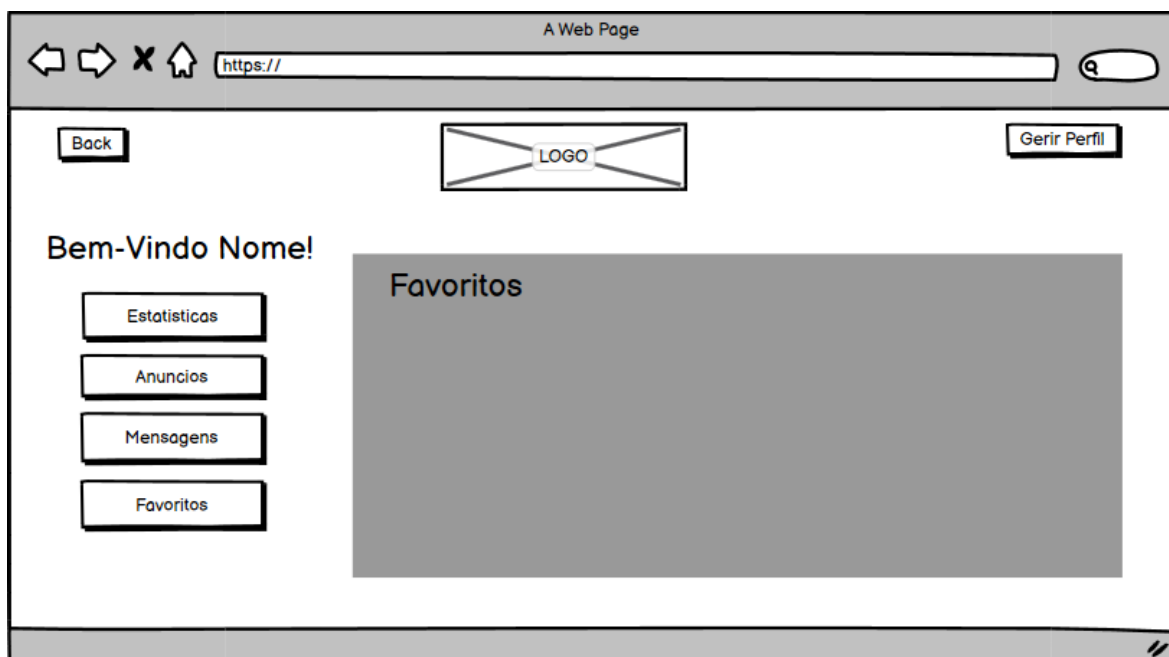
Representação da segunda página ao pressionar o botão estatísticas



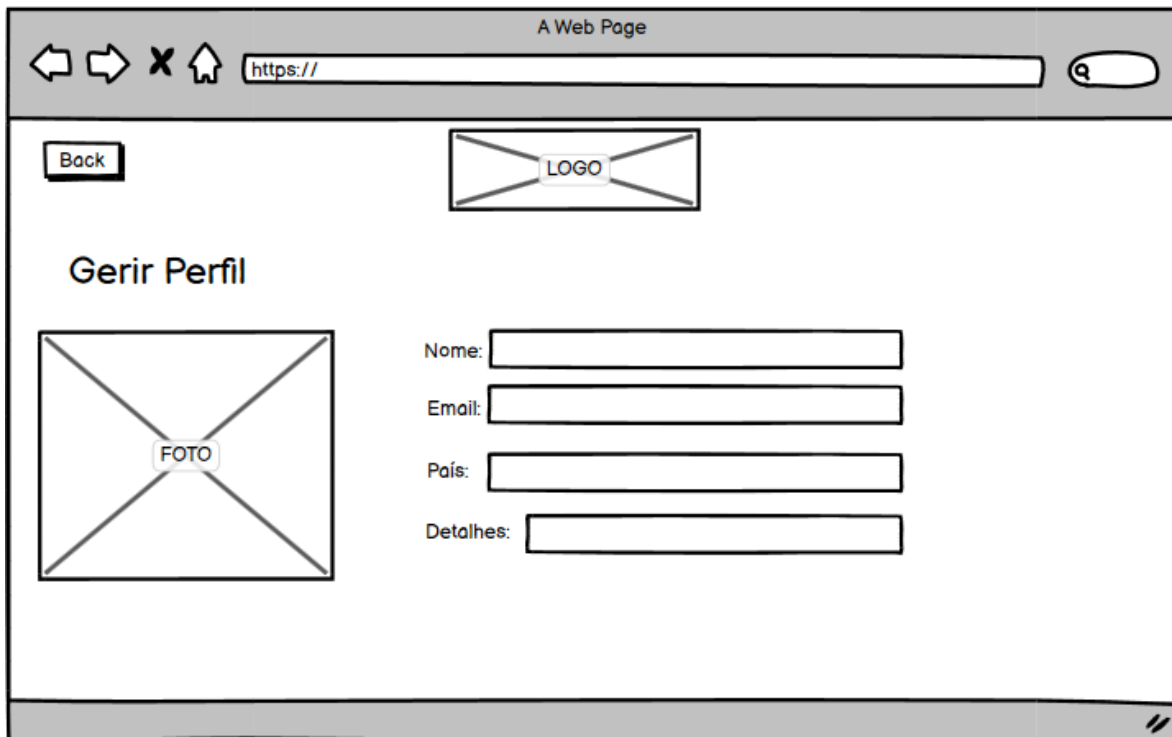
Representação ao pressionar o botão anúncios



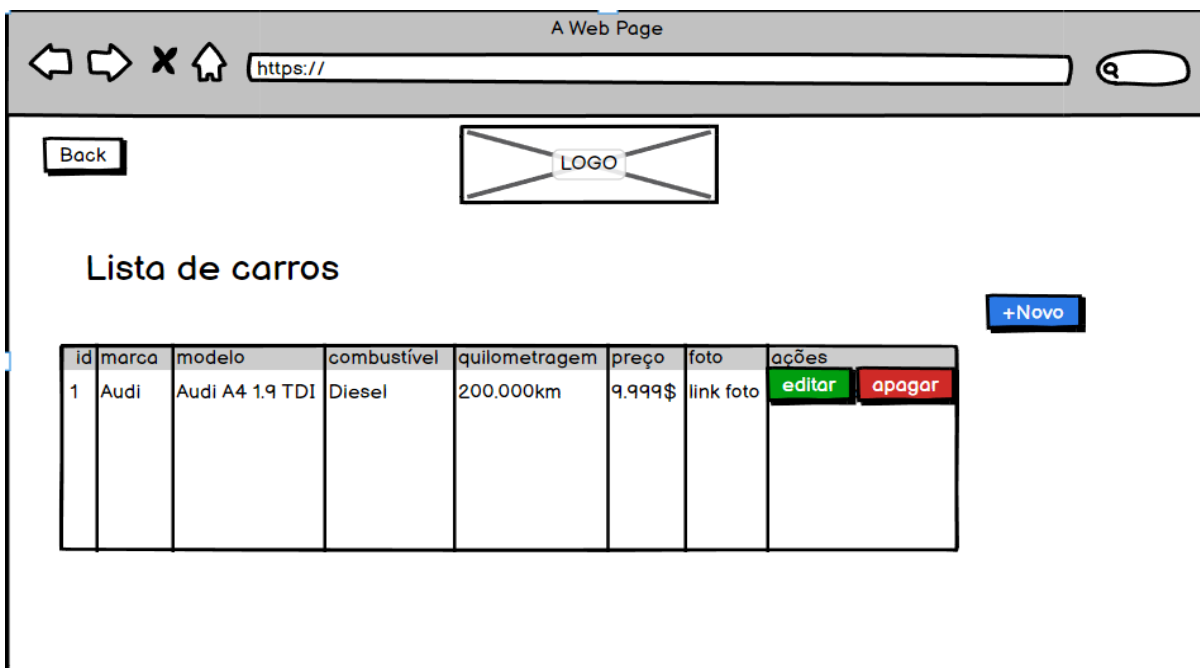
Representação ao pressionar o botão mensagens



Representação ao pressionar o botão favoritos



Representação da página depois de pressionar o botão gerir perfil

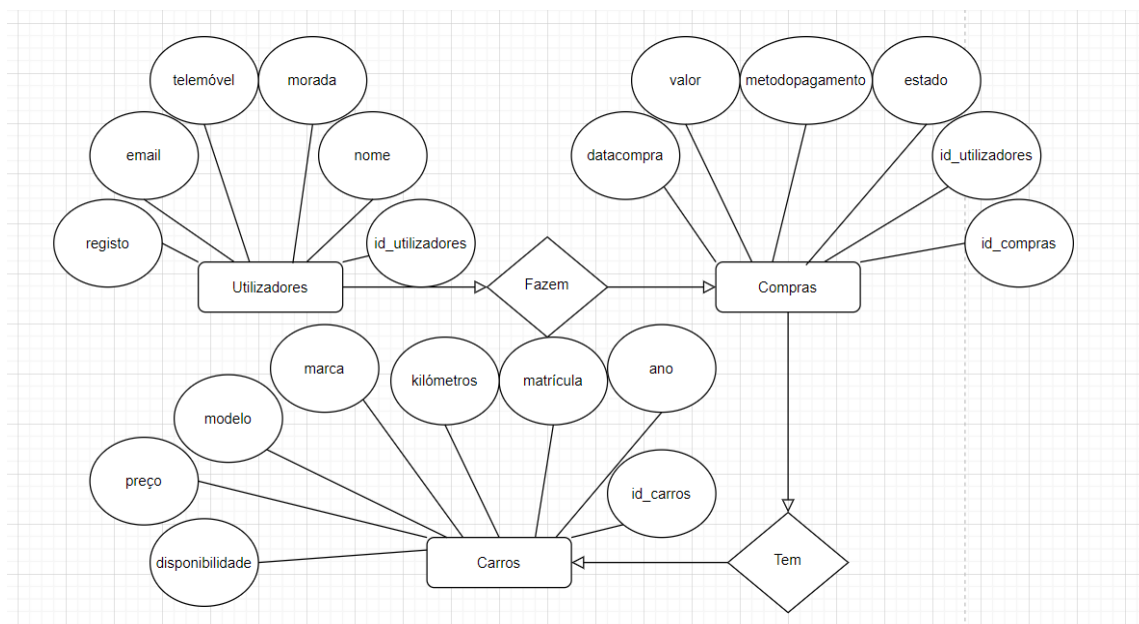


Representação da página depois de pressionar o botão gerir carros

MODELO E-R

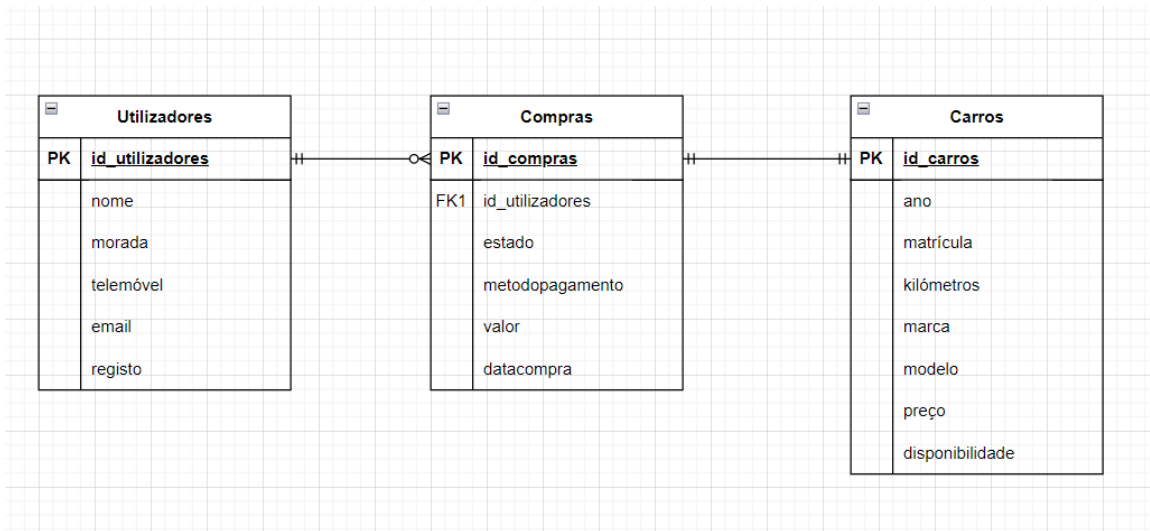
Para desenharmos o modelo Entidade-Relacionamento nós utilizamos o draw.io. No nosso modelo serão implementadas 3 tabelas, sendo elas as seguintes:

(Utilizadores-Compras-Carros).



DESENHO DO MODELO RELACIONAL

Para desenharmos o modelo relacional nós utilizamos o draw.io. No nosso modelo relacional nós implementamos 3 tabelas (utilizadores, compras e carros).



Aqui temos uma imagem do nosso Postgres, onde é possível observar as nossas 3 tabelas criadas e que neste caso, por exemplo, na tabela users, já é possível ver 4 utilizadores criados.

A imagem mostra a interface do pgAdmin 4. No painel esquerdo, a árvore de objetos mostra a estrutura do banco de dados. No painel central, uma consulta SQL foi executada:

```
1 SELECT id, name, email, password, "isAdmin", "Compras", "Favoritos"
2 FROM public."Users";
```

Os resultados da consulta são exibidos na aba "Data Output" na seguinte tabela:

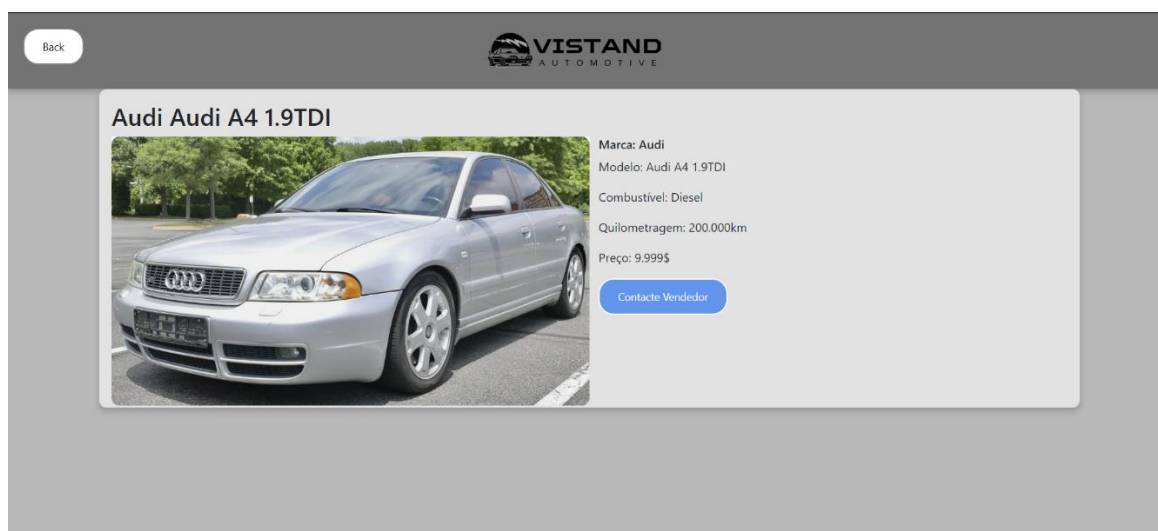
| | id [PK] Integer | name character varying (255) | email text | password text | isAdmin boolean | Compras Integer | Favoritos Integer | Me... |
|---|-----------------|------------------------------|---------------------------|---------------|-----------------|-----------------|-------------------|-------|
| 1 | 5 | paulo | fabiosrocha2119@gmail.com | Fabio2911# | false | 3 | 5 | Ve... |
| 2 | 6 | Chico | fabiosrocha100@gmail.com | Fabio2911# | false | 2 | 4 | Ve... |
| 3 | 1 | Fabio | fabiosrocha7@gmail.com | Fabio2911# | true | 1 | 11 | Ve... |
| 4 | 7 | Fábio | fabiosrocha29@hotmail.com | Fabio2911# | true | 0 | 1 | Ve... |

Na barra de status inferior, há uma mensagem de sucesso: "Successfully run. Total query runtime: 94 msec. 4 rows affected."

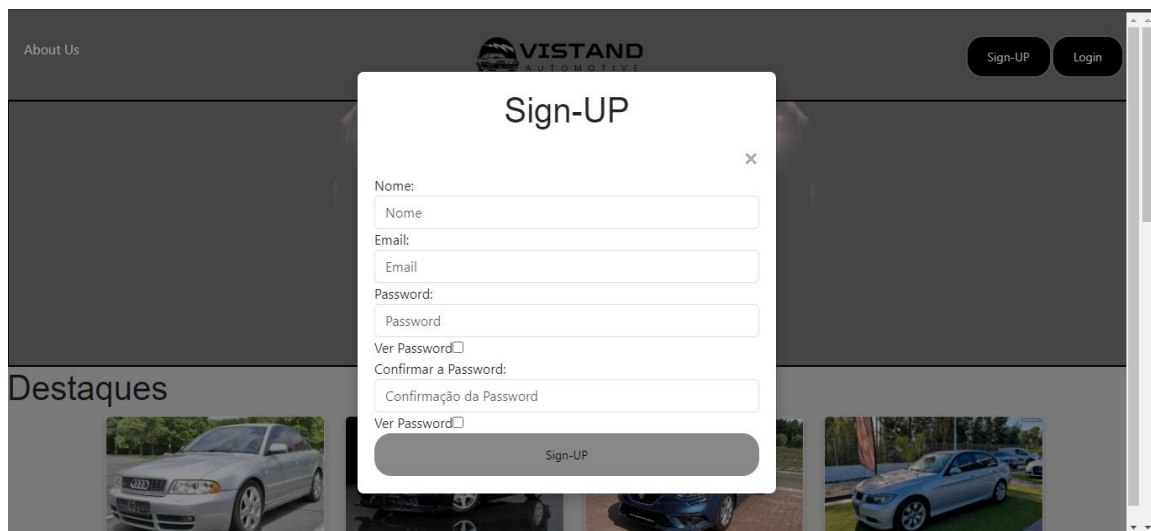
IMAGENS DAS PÁGINAS WEB



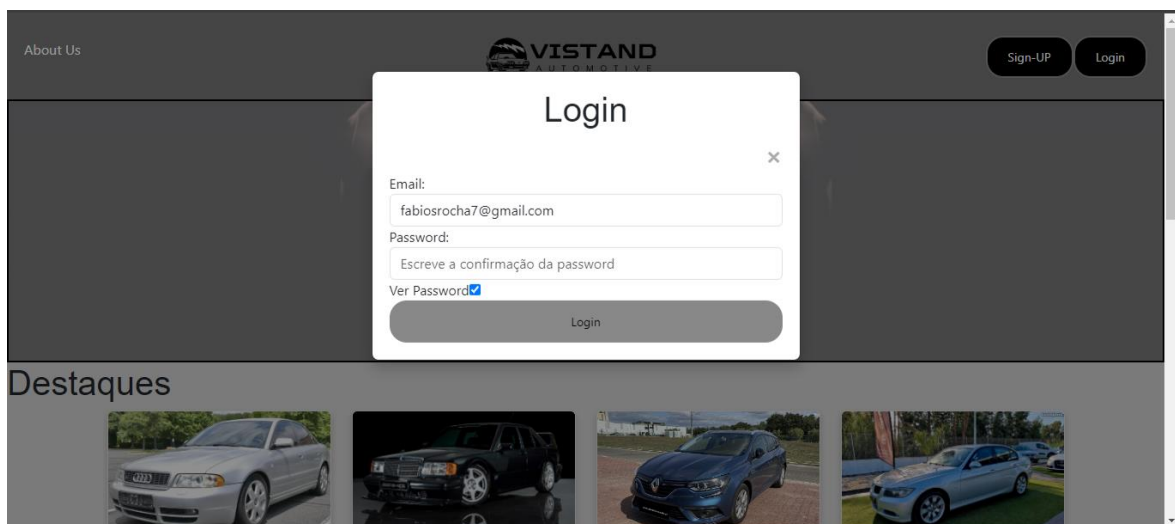
Página inicial



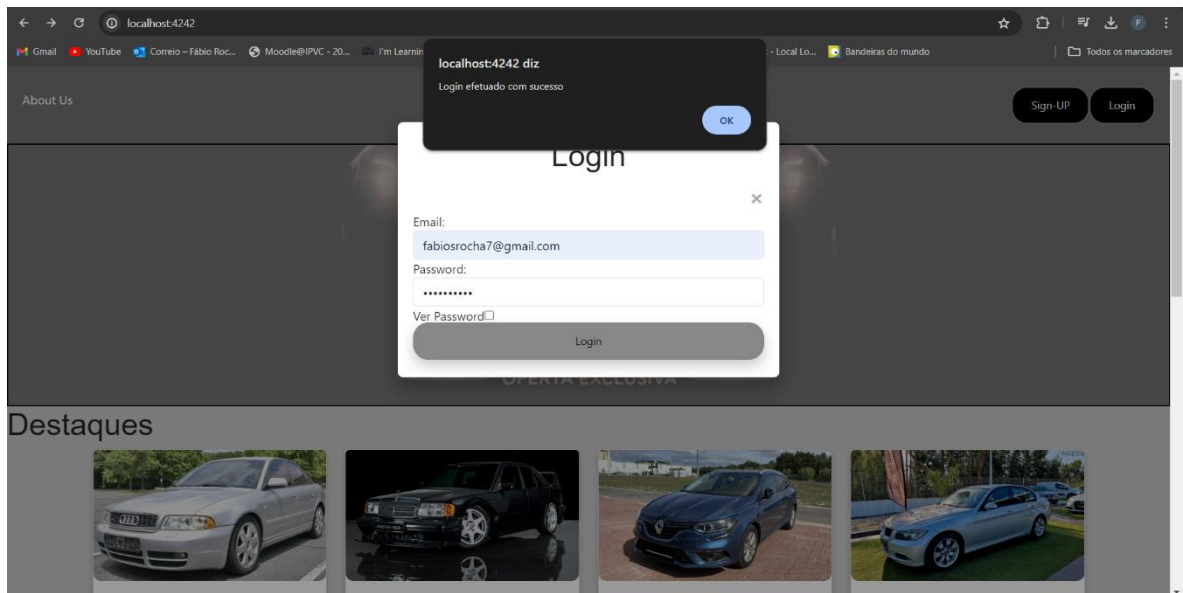
Página ao pressionar um veículo



Modal do sign-up



Modal do login



Login bem efetuado

Bem-vindo, Fabio

- Estatísticas
- Anúncios
- Mensagens
- Favoritos

Página ao dar login ou sign-up

Bem-vindo, Fabio

- Estatísticas
- Anúncios
- Mensagens
- Favoritos

Estatísticas

Nome: Fabio
Administrador: Sim
Email: undefined
Compras: undefined

Modal das estatísticas

Back

Gerir Perfil

Nome:

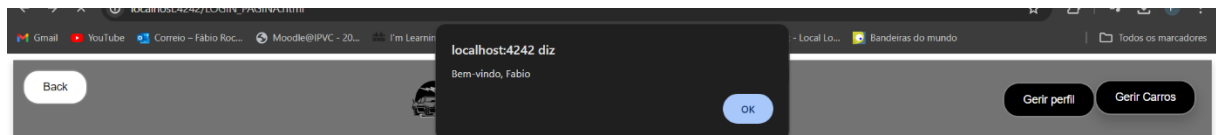
Email:

País:

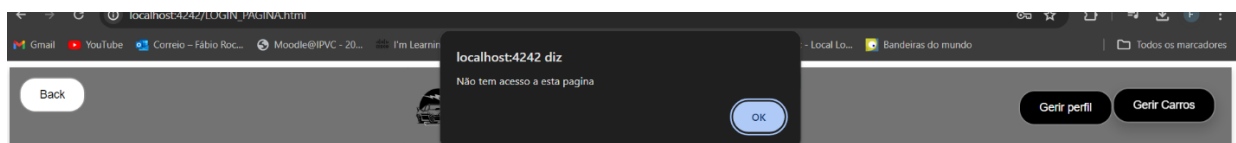
Descrição:

Salvar Alterações

Página de gerir perfil



Utilizador com permissões para entrar em gerir carros



Utilizador sem permissões para entrar em gerir carros



Lista de Carros

+ Novo

| ID | Marca | Modelo | Combustível | Quilometragem | Preço | Foto | Ações |
|----|----------|-----------------------|-------------|---------------|-----------|------------------------------|---|
| 1 | Audi | Audi A4 1.9TDI | Diesel | 200.000km | 9.999\$ | Link da Foto | <button>Editar</button> <button>Apagar</button> |
| 2 | Mercedes | Mercedes 190e 2.5 16V | Gasolina | 499.999km | 49.999\$ | Link da Foto | <button>Editar</button> <button>Apagar</button> |
| 3 | Renault | Megane 1.5dci | Diesel | 220.000km | 13.999\$ | Link da Foto | <button>Editar</button> <button>Apagar</button> |
| 4 | BMW | 320d e90 | Diesel | 320.000km | 10.999\$ | Link da Foto | <button>Editar</button> <button>Apagar</button> |
| 5 | Porsche | 911 turbo 3.3 | Gasolina | 320.000km | 140.000\$ | Link da Foto | <button>Editar</button> <button>Apagar</button> |
| 6 | Mercedes | C63 5.8 V8 | Gasolina | 90.000km | 100.000\$ | Link da Foto | <button>Editar</button> <button>Apagar</button> |
| 7 | Audi | A3 8P 2.0TDI | undefined | 5.000km | 80.000\$ | Link da Foto | <button>Editar</button> <button>Apagar</button> |
| 8 | Porsche | 911 GT3 Coupe | Gasolina | 100.000km | 270.000\$ | Link da Foto | <button>Editar</button> <button>Apagar</button> |

Página de gerir carros

CÓDIGO JSON

Código json utilizado na criação do nosso site.

```
127.0.0.1:5500/trab1/JSON_TRABPRAT1.json

Gmail YouTube Instagram LotsOfJoker

Aplicar estilos de formatação

[
  {
    "marca": "Audi",
    "modelo": "Audi A4 1.9TDI",
    "combustivel": "Diesel",
    "quilometragem": "200.000km",
    "preco": "9.999$",
    "imagem": "Fotos/Carro1.jpg"
  },
  {
    "marca": "Mercedes",
    "modelo": "Mercedes 190e 2.5 16V",
    "combustivel": "Gasolina",
    "quilometragem": "499.999km",
    "preco": "49.999$",
    "imagem": "Fotos/Carro2.jpg"
  },
  {
    "marca": "Renault",
    "modelo": "Megane 1.5dci",
    "combustivel": "Diesel",
    "imagem": "Fotos/Carro3.jpg",
    "preco": "13.999$",
    "quilometragem": "220.000km"
  },
  {
    "marca": "BMW",
    "modelo": "320d e90",
    "imagem": "Fotos/Carro4.jpg",
    "quilometragem": "320.000km",
    "combustivel": "Diesel",
    "preco": "10.999$"
  },
  {
    "marca": "Porsche",
    "modelo": "911 turbo 3.3",
    "combustivel": "Gasolina",
    "imagem": "Fotos/Carro5.jpg",
    "quilometragem": "320.000km",
    "preco": "140.000$"
  }
]
```

```
← → ↻ 127.0.0.1:5500/trab1/JSON_TRABPRAT1.json
Gmail YouTube Instagram LotsOfJoker
Aplicar estilos de formatação □
preco": 140.000$,
},
{
  "marca": "Mercedes",
  "modelo": "C63 5.8 V8",
  "quilometragem": "90.000km",
  "preco": "100.000$",
  "imagem": "Fotos/Carro6.jpg",
  "combustivel": "Gasolina"
},
{
  "marca": "Audi",
  "modelo": "A3 8P 2.0TDI",
  "quilometragem": "5.000km",
  "preco": "80.000$",
  "imagem": "Fotos/Carro7.jpg"
},
{
  "marca": "Porsche",
  "modelo": "911 GT3 Coupe",
  "quilometragem": "100.000km",
  "preco": "270.000$",
  "imagem": "Fotos/Carro8.jpg",
  "combustivel": "Gasolina"
},
{
  "marca": "Ford",
  "modelo": "Focus 3 1.5TDCI",
  "quilometragem": "200.000km",
  "preco": "13.999$",
  "imagem": "Fotos/Carro9.jpg",
  "combustivel": "Diesel"
},
{
  "marca": "Volkswagen",
  "modelo": "Golf 1.6TDI",
  "quilometragem": "130.000km",
  "preco": "13.200$",
  "imagem": "Fotos/Carro10.jpg",
  "combustivel": "Diesel"
}
]
```

Imagem das principais scripts que utilizamos no nosso trabalho

Nestas duas fotos está representada a script utilizada por nós, em que contem as funções utilizadas para abrir e fechar os nossos modais, como também a script que recebe o ficheiro json convertendo-o para .json, e que em seguida para todos os carros presentes no ficheiro vai criar um card de cada carro com imagem,nome,marca, quilómetros e preço.

```
<script>

function SignUp(){
  var modal= document.getElementById("ModalSign");
  modal.style.display="block";
}
function fecharSign(){
  var modal = document.getElementById("ModalSign");
  modal.style.display = "none";
}

function Login(){
  var modal= document.getElementById("ModalLogin");
  modal.style.display="block";
}
function fecharLogin(){
  var modal = document.getElementById("ModalLogin");
  modal.style.display = "none";
}
function login(){
  window.location.href="LOGIN_PAGINA.html"
}

//funcao para carregar dados pelo json
function carregarCarros(){
  fetch('JSON_TRABPRAT1.json')//carrega o json
  .then(resposta => resposta.json())//converte a resposta para json
  .then(data => {
    //para cada carro
    data.forEach(carro => {
      //Cria a div de cada
      const cardDiv = document.createElement('div');
      cardDiv.classList.add('col-md-3');

      //Cria a div de cada
      const cardDiv = document.createElement('div');
      cardDiv.classList.add('col-md-3');
      //Cria o código HTML com os cards
      cardDiv.innerHTML = `
        <div class="card" onclick="paginaCarro('${carro.marca}')">
          <div class="img-container">
            
          </div>
          <div class="card-body mt-3">
            <h4 class="card-title">${carro.marca}</h4>
            <p class="card-text">${carro.modelo ? carro.modelo : ''}</p>
            <p class="card-text">${carro.quilometragem ? carro.quilometragem : ''} | ${carro.preco ? carro.preco : ''}</p>
          </div>
        </div>
      `;
      //adiciona o card à div com id "cardCarros"
      document.getElementById('cardCarros').appendChild(cardDiv);
    });
  });
}

//funcao para abrir uma nova pagina com cada carro
function paginaCarro(marca){
  window.location.href = `CARRO_PAGINA.html?marca=${marca}`;
}

//Carrega os dados dos carros quando a pagina carrega
window.onload = carregarCarros;

</script>
```

Esta script lê qual dos 4 botões foi pressionado e mostra no modal o conteúdo em relação ao botão pressionado.

```
<script>
// Função para abrir o novo modal com o conteúdo específico
function openMeuModal(content) {
  var modal = document.getElementById("meuModal");
  var modalContent = document.getElementById("modalConteudo");
  var titulo = "";

  // Definindo o título e o conteúdo com base no botão clicado
  switch(content) {
    case 'Estatísticas':
      titulo = "As minhas estatísticas";
      modalContent.innerHTML = "";
      break;
    case 'Anuncios':
      titulo = "Os meus anúncios";
      modalContent.innerHTML = "";
      break;
    case 'Mensagens':
      titulo = "As minhas mensagens";
      modalContent.innerHTML = "";
      break;
    case 'Favoritos':
      titulo = "Favoritos";
      modalContent.innerHTML = "";
      break;
  }

  // Exibe o novo modal com título e conteúdo
  modalContent.innerHTML = "<h2>" + titulo + "</h2>" + modalContent.innerHTML;
  modal.style.display = "block";
}
</script>
```

Este script obtém uma lista de carros em formato JSON, e depois exibe esses carros em uma tabela HTML.

```
131 <script>
133 .then(response => response.json())
134 .then(data => {
135   const tabela = document.getElementById('carrosTabela');
136   data.forEach(carro => {
137     const tr = document.createElement('tr');
138     tr.innerHTML = `
139       <td>${carro.id}</td>
140       <td>${carro.marca}</td>
141       <td>${carro.modelo}</td>
142       <td>${carro.combustivel}</td>
143       <td>${carro.quilometragem}</td>
144       <td>${carro.preco}</td>
145       <td><a href="${carro.imagem}" target="_blank">Link da Foto</a></td>
146     <td>
147       <button class="btn btn-success btn-sm" onclick="abrirEditarCarro(${carro.id})">Editar</button>
148       <button class="btn btn-danger btn-sm" onclick="apagarCarro(${carro.id})">Apagar</button>
149     </td>
150   `;
151   tabela.appendChild(tr);
152   });
153 }
154 .catch(error => console.error('Erro ao carregar carros:', error));
155
```

Este script vê se o utilizador tem permissões administrativas para entrar na secção gerir carros e se o mesmo não tiver mostra uma mensagem de alerta.

```
39  <script>
46  }
47
48  const gerir_carros = async () => {
49    var dados = {
50      token: localStorage.getItem("token"),
51    };
52    console.log(JSON.stringify(dados));
53    const response= await fetch("http://localhost:4242/api/pgs/auth/letoken", {
54      method: "POST",
55      headers: {
56        "Content-Type": "application/json",
57      },
58      body: JSON.stringify(dados),
59    })
60    console.log(response);
61    dados= await response.json()
62    console.log(dados);
63    if(!dados.isAdmin){
64      alert("Não tem acesso a esta pagina");
65    }
66    else{
67      window.location.href='Gerir_Carros.html';
68      alert(`Bem-vindo, ${dados.name}`);
69    }

```

API

A nossa API foi projetada para permitir que os utilizadores possam realizar operações CRUD numa base de dados de carros. Para além disso, ela inclui funcionalidades de autenticação e autorização para garantir que apenas utilizadores autorizados possam ter acesso e modificar certo tipo de dados.

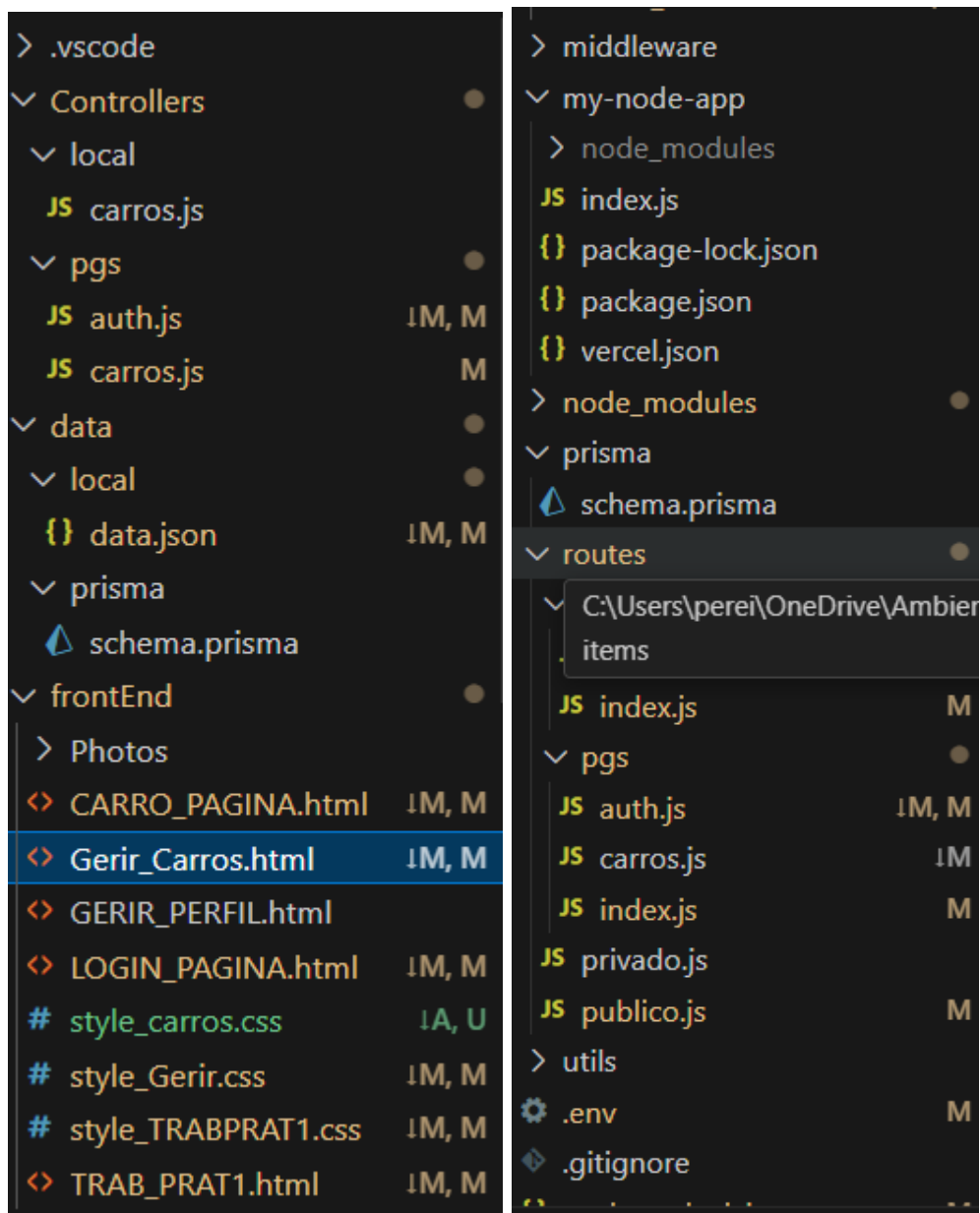
Aqui temos alguns exemplos de endpoints presentes na nossa API:

```
const response= await fetch("http://localhost:4242/api/pgs/auth/signin", {  
  method: "POST",  
  headers: {  
    "Content-Type": "application/json",  
  },  
});
```

```
fetch("http://localhost:4242/api/pgs/auth/signup", {  
  method: "POST",  
  headers: {"Content-Type": "application/json"},  
  body: JSON.stringify(dados),  
});
```

```
fetch('http://localhost:4242/api/pgs/carros')  
  .then(response => response.json())  
  .then(data => {  
    const tabela = document.getElementById('carrosTabela');  
    data.forEach(carro => {  
      const tr = document.createElement('tr');  
      tr.innerHTML = `  
        <td>${carro.id}</td>  
        <td>${carro.marca}</td>  
        <td>${carro.modelo}</td>  
        <td>${carro.combustivel}</td>  
        <td>${carro.quilometragem}</td>  
        <td>${carro.preco}</td>  
        <td><a href="${carro.imagem}" target="_blank">Link da Foto</a></td>  
        <td>  
          <button class="btn btn-success btn-sm" onclick="abrirEditarCarro(${carro.id})">Editar</button>  
          <button class="btn btn-danger btn-sm" onclick="apagarCarro(${carro.id})">Apagar</button>  
        </td>  
      `;  
      tabela.appendChild(tr);  
    });  
  });
```


Aqui temos as nossas rotas para a conclusão deste trabalho.



VERCEL

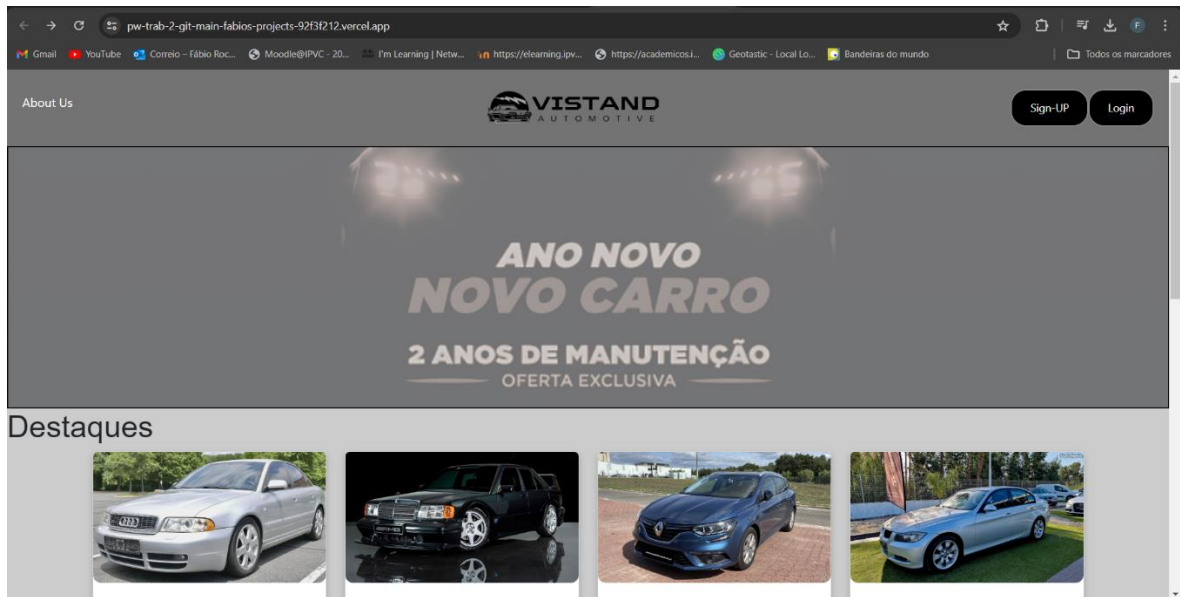
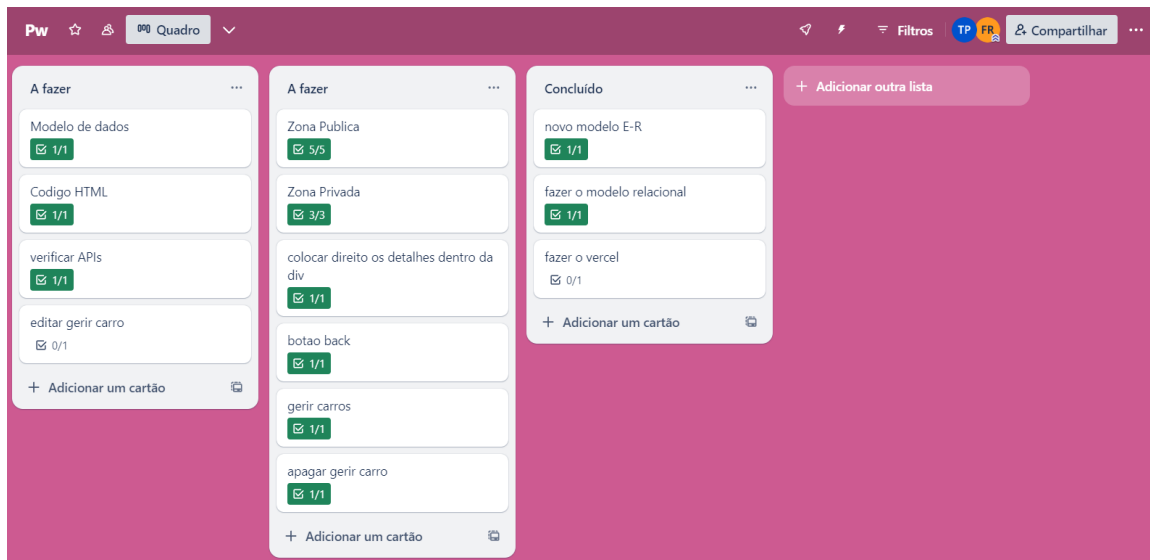


IMAGEM DO NOSSO TRELLO



REFERÊNCIAS

- <https://www.devmedia.com.br/o-que-e-uml-e-diagramas-de-caso-de-uso-introducao-pratica-a-uml/23408>
- https://www.w3schools.com/html/html_scripts.asp
- <https://www.w3schools.com/css/>
- PowerPoints das aulas[António Lira Fernades](online)