



Instituto Politécnico de Viseu
Escola Superior de Tecnologia e Gestão de Viseu
Departamento de Informática

Unidade Curricular: Aplicações para a Internet I

Relatório Relativo ao Projeto Prático

Tema: MyRecipeList

Realizado por: Maurício Florentino - 27417

Hélder Albuquerque - 27409

Duarte Girão - 27457

Viseu, 2025

Instituto Politécnico de Viseu
Escola Superior de Tecnologia e Gestão de Viseu
Departamento de Informática

Relatório relativo ao Projeto Prático

Curso de Licenciatura em Engenharia Informática

Unidade Curricular de Aplicações para a Internet I

MyRecipeList

Ano Letivo 2025/26

Viseu, 2025

ÍNDICE

| | |
|---|-----------|
| 1. Introdução | 1 |
| 2. Questões de análise | 2 |
| 3. Desenvolvimento do código | 3 |
| 3.1. Página Inicial- Secções, Menus, Submenus e Slider | 3 |
| 3.2. Formulário de Contactos | 9 |
| 3.3. Lista de receitas | 11 |
| 3.3.1. Estrutura e Componentes | 11 |
| 4. Valorização do trabalho | 13 |
| 4.1. Dinamismo do Website | 13 |
| 4.2. Página de Contactos | 14 |
| 5. Conclusões | 16 |
| 6. Bibliografia | 17 |

Índice de Figuras

| | |
|---|----|
| Figura 1 - Página Inicial | 3 |
| Figura 2 - Código da navbar | 3 |
| Figura 3 - Estilo da Navbar..... | 4 |
| Figura 4 - Estilo da Navbar (cont) | 4 |
| Figura 5 - Código do Slider de imagens | 5 |
| Figura 6 - Estilo do slider /imagens do slider..... | 5 |
| Figura 7 - Código das Receitas em Destaque | 6 |
| Figura 8 - Código das Receitas em Destaque | 6 |
| Figura 9 - Código das Receitas em Destaque (cont) | 7 |
| Figura 10 – Categorias..... | 7 |
| Figura 11 - Código das Categorias | 8 |
| Figura 12 - Página de Contactos..... | 9 |
| Figura 13 - Código da Validação dos dados..... | 9 |
| Figura 14 - Código da Validação dos dados (cont) | 10 |
| Figura 15 - Código da Validação dos dados (cont.) | 10 |
| Figura 16 - Lista de Receitas | 12 |
| Figura 17- Página de Receitas no Computador | 13 |
| Figura 18- Página de Receitas em um Dispositivo Móvel | 13 |
| Figura 19 - Estrutura do Formulário feito com React | 14 |
| Figura 20 - Código que preenche a página com o Formulário | 15 |

1. Introdução

Este relatório tem como objetivo apresentar o projeto realizado para a Unidade Curricular de Aplicações para a Internet I, com recurso às linguagens de HTML, CSS, JavaScript e React. Para tal, MyRecipeList - uma página dedicada à apresentação de receitas clássicas, tradicionais e simpels - foi a solução desenvolvida pelo grupo.

Este documento está estruturado em cinco capítulos que se seguem a esta introdução.

No segundo capítulo, são apresentadas as respostas às questões de análise presentes no enunciado deste projeto.

Segue-se o capítulo três, onde são mostrados alguns excertos do código e a sua respetiva visualização no website.

O capítulo quatro aborda mecanismos relevantes no assunto do trabalho e a sua valorização.

Termina-se com o capítulo cinco, onde se apresentam as conclusões deste trabalho.

2. Questões de análise

Este capítulo aborda as respostas dadas às questões de análise, colocadas no enunciado deste projeto.

- **Quem é o público-alvo? Que características definem o público-alvo?**

O público-alvo são pessoas que procuram receitas simples e saborosas para fazerem em casa, especialmente aquelas que têm pouco tempo disponível.

- **O que é que o público-alvo procura no website? Que tarefas tem de realizar no website? Quais as suas expectativas?**

O público-alvo procura no website receitas básicas e saborosas, assim como ideias e dicas de preparação. As principais tarefas são pesquisar receitas, consultar ingredientes e seguir instruções simples. As expectativas são encontrar conteúdos claros, práticos e fáceis de usar.

- **Que objetivos quer o vosso grupo atingir ao implementar o website? São objetivos quantitativos ou qualitativos?**

O nosso grupo pretende criar um website útil e acessível, que ajude os utilizadores a encontrar receitas simples e saborosas. Os objetivos são maioritariamente qualitativos, focando-se na facilidade de uso, clareza da informação.

- **Que atitude se pretende conferir ao website?**

Pretende-se conferir ao website uma atitude educativa, incentivando a aprendizagem de receitas novas na cozinha.

- **Quais são as cores dominantes do website?**

As cores dominantes serão o branco e o laranja.

- **Que funcionalidades e conteúdos gostaria de ver implementadas no site?**

Queríamos adicionar uma página de admin onde o administrador poderia adicionar, remover e editar receitas e reviews das receitas pelos utilizadores.

- **Como podemos medir o sucesso do website? Como sabemos se os objetivos foram atingidos? É possível medir objetivamente (quantitativamente) o sucesso do website? Como?**

O sucesso do website pode ser medido pelo número de receitas, as suas popularidades e a satisfação.


```

.navbar {
  background: var(--cream);
  color: var(--muted);
  border-bottom: 1px solid #59, 47, 0.06;
  box-shadow: 0 1px 0 #0, 0, 0, 0.02;
  padding: 10px 20px;
}

.navbar .container {
  display: flex;
  align-items: center;
  justify-content: center;
  position: relative;
  width: 100%;
}

.navbar .container ul {
  display: flex;
  margin: 0;
  padding: 0;
  align-items: center;
}

.main-nav {
  display: flex;
  align-items: center;
  gap: 12px;
}

.main-nav ul {
  list-style: none;
  margin: 0;
  padding: 0;
  display: flex;
  justify-content: center;
  align-items: center;
}

```

Figura 3 - Estilo da Navbar

```

.navbar a {
  color: var(--text);
  text-decoration: none;
  font-weight: 600;
  padding: 8px 10px;
  border-radius: 6px;
  transition: background 0.18s ease, color 0.18s ease;
}

.navbar .container ul li {
  display: inline-flex;
  align-items: center;
}

.navbar a {
  display: inline-flex;
  align-items: center;
  gap: 6px;
}

.navbar a:hover,
.navbar a:focus {
  color: #fff;
  background: var(--primary);
  text-decoration: none;
}

.navbar .fa-house {
  color: var(--text);
}

.nav-toggle {
  display: none;
  background: transparent;
  border: 0;
  color: inherit;
  cursor: pointer;
  padding: 8px;
}

```

Figura 4 - Estilo da Navbar (cont)

O código presente nas Figuras 2, 3 e 4 serve para todas as páginas do website, dado que o a navbar é igual em todas as páginas.


```
<div class="image-slider">  
    
    
    
</div>
```

Figura 5 - Código do Slider de imagens

```
.image-slider {  
  width: 100%;  
  max-width: none;  
  margin: 0;  
  left: auto;  
  transform: none;  
  position: relative;  
  overflow: hidden;  
  border-radius: 0;  
  box-shadow: var(--shadow);  
  background: var(--background);  
  height: var(--slider-height);  
}  
  
.slider-img {  
  width: 100%;  
  height: var(--slider-height);  
  object-fit: cover;  
  position: absolute;  
  left: 0;  
  top: 0;  
  opacity: 0;  
  transition: opacity 0.7s ease;  
  z-index: 0;  
  pointer-events: none;  
}  
  
.slider-img.active {  
  opacity: 1;  
  z-index: 1;  
  pointer-events: auto;  
}
```

Figura 6 - Estilo do slider /imagens do slider

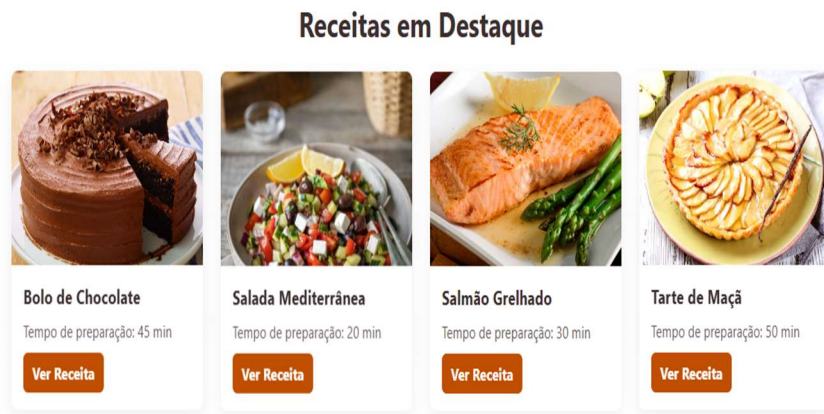


Figura 7 - Código das Receitas em Destaque

Esta secção foi criada para apresentar, de forma mais apelativa, as receitas que consideramos mais atrativas e populares. Ao existir uma secção dedicada, torna-se mais fácil para o utilizador encontrar rapidamente estas receitas.

```
<section id="receitas-destaque" class="recipes-featured">
  <div class="container">
    <h2 class="section-title">Receitas em Destaque</h2>
    <div class="recipes-grid">
      <article class="recipe-card">
        <div class="card-media">
          
        </div>
        <div class="card-body">
          <h3 class="recipe-name">Bolo de Chocolate</h3>
          <p class="recipe-time">Tempo de preparação: 45 min</p>
          <a class="btn" href="receita-detalle.html?id=BoloDeChocolate">Ver Receita</a>
        </div>
      </article>

      <article class="recipe-card">
        <div class="card-media">
          
        </div>
        <div class="card-body">
          <h3 class="recipe-name">Salada Mediterrânea</h3>
          <p class="recipe-time">Tempo de preparação: 20 min</p>
          <a class="btn" href="receita-detalle.html?id=SaladaMediterranea">Ver Receita</a>
        </div>
      </article>
    </div>
  </div>
</section>
```

Figura 8 - Código das Receitas em Destaque

```

<article class="recipe-card">
  <div class="card-media">
    
  </div>
  <div class="card-body">
    <h3 class="recipe-name">Salmão Grelhado</h3>
    <p class="recipe-time">Tempo de preparação: 30 min</p>
    <a class="btn" href="receita-detalle.html?id=SalmãoGrelhado">Ver Receita</a>
  </div>
</article>

<article class="recipe-card">
  <div class="card-media">
    
  </div>
  <div class="card-body">
    <h3 class="recipe-name">Tarte de Maçã</h3>
    <p class="recipe-time">Tempo de preparação: 50 min</p>
    <a class="btn" href="receita-detalle.html?id=TarteDeMaca">Ver Receita</a>
  </div>
</article>
</div>
</section>

```

Figura 9 - Código das Receitas em Destaque (cont)

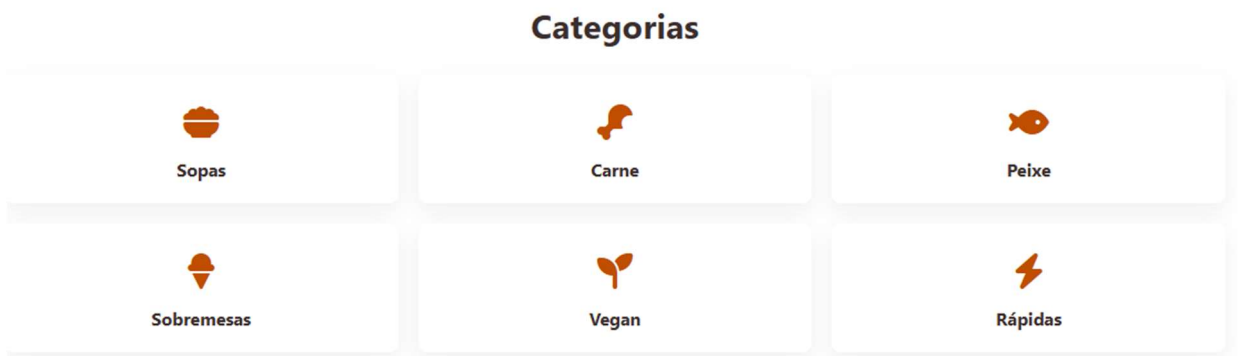


Figura 10 – Categorias

Todas as receitas têm a sua categoria, estando esta secção também presente na página principal.

```

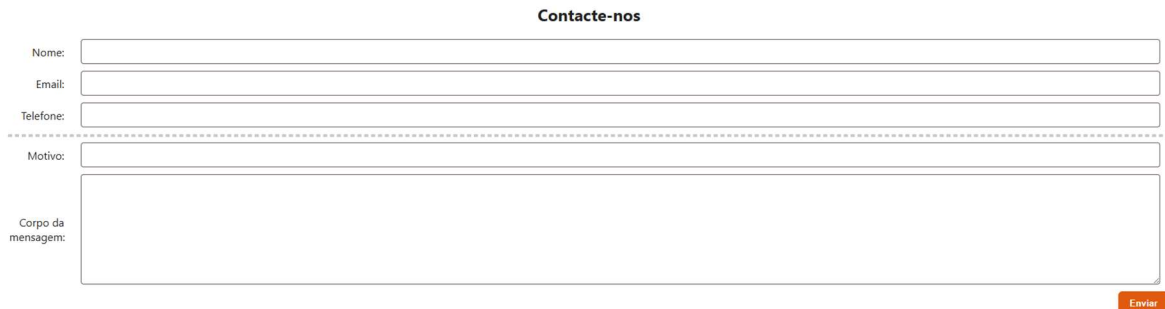
<section id="categorias" class="categories-section">
  <div class="container">
    <h2 class="section-title">Categorias</h2>
    <div class="categories-grid">
      <a class="category-card" href="receitas.html#sopa" aria-label="Sopas">
        <div class="cat-icon"><i class="fa-solid fa-bowl-food" aria-hidden="true"></i></div>
        <div class="cat-name">Sopas</div>
      </a>
      <a class="category-card" href="receitas.html#carne" aria-label="Carne">
        <div class="cat-icon"><i class="fa-solid fa-drumstick-bite" aria-hidden="true"></i></div>
        <div class="cat-name">Carne</div>
      </a>
      <a class="category-card" href="receitas.html#peixe" aria-label="Peixe">
        <div class="cat-icon"><i class="fa-solid fa-fish" aria-hidden="true"></i></div>
        <div class="cat-name">Peixe</div>
      </a>
      <a class="category-card" href="receitas.html#sobremesas" aria-label="Sobremesas">
        <div class="cat-icon"><i class="fa-solid fa-ice-cream" aria-hidden="true"></i></div>
        <div class="cat-name">Sobremesas</div>
      </a>
      <a class="category-card" href="receitas.html#vegan" aria-label="Vegan">
        <div class="cat-icon"><i class="fa-solid fa-seedling" aria-hidden="true"></i></div>
        <div class="cat-name">Vegan</div>
      </a>
      <a class="category-card" href="receitas.html#rapidas" aria-label="Rápidas">
        <div class="cat-icon"><i class="fa-solid fa-bolt" aria-hidden="true"></i></div>
        <div class="cat-name">Rápidas</div>
      </a>
    </div>
  </div>
</section>

```

Figura 11 - Código das Categorias

3.2. Formulário de Contactos

Um dos requisitos necessários presente no enunciado é a criação de um formulário, capaz de receber dados como nome, motivo de contacto, email, telefone e o respetivo texto de contacto, com a capacidade de validar tal dados.



Formulário de contacto com o título "Contacte-nos". O formulário contém campos para "Nome:", "Email:", "Telefone:", "Motivo:" e um campo de texto grande para "Corpo da mensagem:". Um botão laranja "Enviar" está no canto inferior direito.

Figura 12 - Página de Contactos

O formulário da Figura 12 recebe os dados introduzidos pelo utilizador, e armazena-os no local storage do browser. Além disso, todos os dados inseridos são validados, a nível do formato e da obrigatoriedade. Todas estas funcionalidades são garantidas através de React.

```

1 //Função que valida dados
2 //Na prática, os dados também devem ser validados no lado do servidor
3 function validarDados(nome, email, telefone, motivo, texto)
4 {
5     if(nome == "" || email == "" || telefone == "" || motivo == "" || texto == "" ||
6        nome == null || email == null || telefone == null || motivo == null || texto == null
7     )
8     {
9         console.log("Dados por preencher");
10        return "Ainda tem dados por preencher!";
11    }
12
13    let output = "";
14
15    //-----Validações nome
16    if(nome.length < 2 || nome.length > 50)
17    {
18        console.warn("Erro no nome!");
19        output += "Nome inválido!\n";
20    }
21
22    //-----Validações email
23    if(email.length < 5 || email.length > 50)
24    {
25        console.warn("Erro no email!");
26        output += "Email inválido!\n";
27    }
28    else
29    {
30        let val = false;
31        let j = 0;
32
33        for(let i = 1; i < email.length-3; i++) //O '@' não pode ser nenhum dos últimos 3 caracteres,
34        { // por isso não vale a pena ver o email até ao fim
35
36            if(email[i] == '@')
37            {
38                val = true;
39                j = i; //O 'j' vai ser utilizado para encontrar o ponto final, que só pode existir depois do '@' e outro caractere
40            }
41        }
42
43        if(val)
44        {
45            val = false;
46            for(j+2; j < email.length-1; j++)
47            {
48                if(email[j] == '.')

```

Figura 13 - Código da Validação dos dados

```

42
43     if(val)
44     {
45         val = false;
46         for(j+2; j < email.length-1; j++)
47         {
48             if(email[j] == '.')
49             {
50                 val = true;
51             }
52         }
53     }
54
55
56     if(!val)
57     {
58         console.warn("Erro no email!");
59         output += "Email inválido! (erro no '@' ou no ponto final)\n";
60     }
61 }
62
63 //-----Validações telefone
64 if(telefone.length != 9)
65 {
66     console.warn("Erro no telefone!");
67     output += "Número de telefone inválido! (número de dígitos maior ou menor que 9)\n";
68 }
69 else
70 {
71     for(let i = 0; i < 9; i++)
72     {
73         if(telefone[i] < '0' || telefone[i] > '9')
74         {
75             console.warn("Erro no telefone!");
76             output += "Dígito inválido no telefone ["+telefone[i]+"]\n";
77         }
78     }
79 }
80
81 //-----Validações motivo
82 if(motivo.length < 2 || motivo.length > 50)
83 {
84     console.warn("Erro no motivo!");
85     output += "Motivo inválido!\n";
86 }
87
88 //-----Validações texto

```

Figura 14 - Código da Validação dos dados (cont)

```

79     }
80
81     //-----Validações motivo
82     if(motivo.length < 2 || motivo.length > 50)
83     {
84         console.warn("Erro no motivo!");
85         output += "Motivo inválido!\n";
86     }
87
88     //-----Validações texto
89     if(texto.length < 2 || texto.length > 250)
90     {
91         console.warn("Erro na mensagem!");
92         output += "Corpo da mensagem inválido!\n";
93     }
94
95     return output;
96 }
97

```

Figura 15 - Código da Validação dos dados (cont.)

3.3. Lista de receitas

Na secção de receitas, é apresentada uma lista de pratos, com o objetivo de oferecer opções diversificadas para diferentes gostos e ocasiões. Cada receita inclui informações essenciais, como imagem ilustrativa e tempo de preparação, facilitando a escolha do utilizador.

A estrutura da lista de receitas foi desenvolvida recorrendo a HTML (para a organização e semântica da página, garantindo acessibilidade), CSS (para a estilização e adaptação a diferentes dispositivos) e JavaScript (para interatividade e funcionalidades adicionais).

3.3.1. Estrutura e Componentes

A secção das receitas está organizada de forma intuitiva, com os seguintes elementos principais:

- Cabeçalho (Header), que contém o logotipo e a navegação principal, incluindo menus responsivos.

- Secção das Receitas , constituído por:
 - Estrutura em layout flexível para apresentar as receitas de forma organizada e responsiva.
 - Agrupamento das receitas por categoria, facilitando a navegação.
 - Um bloco de informações com o nome da receita e uma breve descrição.
 - Uma imagem ilustrativa, ajustada com CSS
 - Indicação do tempo de preparação, acompanhada de um ícone, para rápida identificação
- Rodapé (Footer), que contém links rápidos para as redes sociais e para outras secções do website, apresentando também a data e a hora de forma dinâmica..

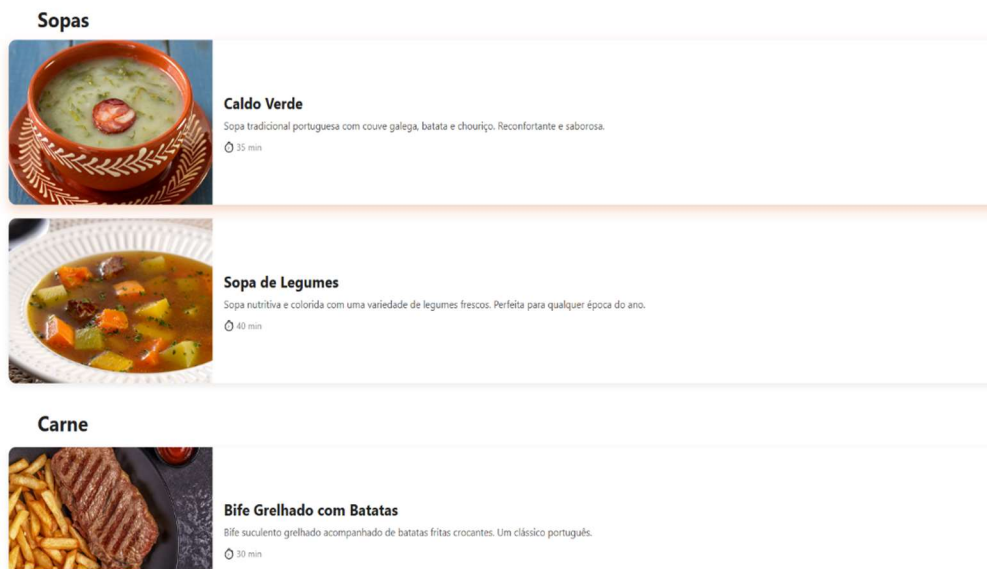


Figura 16 - Lista de Receitas

4. Valorização do trabalho

Neste capítulo, são apresentadas duas funcionalidades que valorizam o trabalho, desenvolvidas com o objetivo de melhorar a experiência do utilizador.

4.1. Dinamismo do Website

O website foi desenvolvido de forma flexível, permitindo ser utilizado tanto em dispositivos móveis como em computadores.

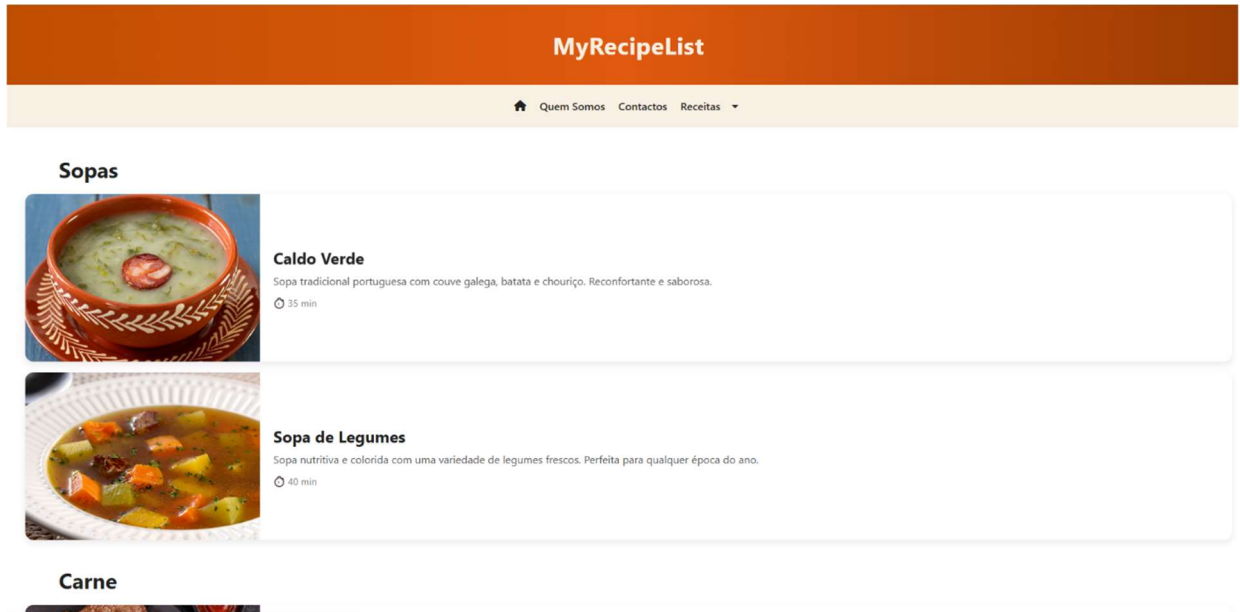


Figura 17- Página de Receitas no Computador

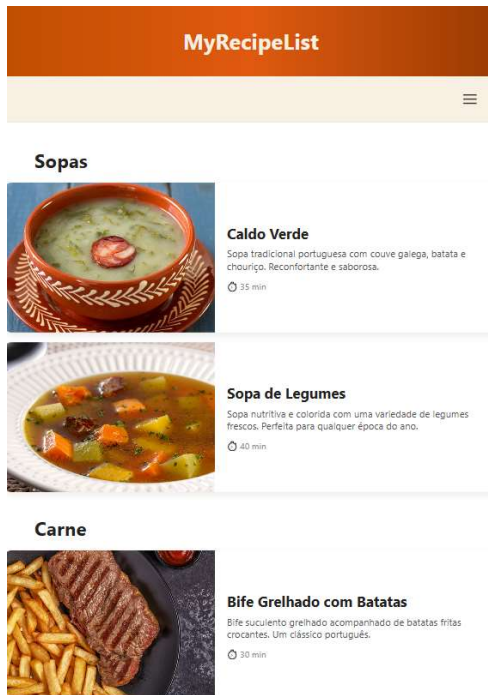


Figura 18- Página de Receitas em um Dispositivo Móvel

4.2. Página de Contactos

A página de contactos foi feita inteiramente em React.

Isto incluiu o HTML

```
135
136 //Função que cria os elementos visuais do formulário
137 function formulario(props){
138
139   return React.createElement("form", {onSubmit: submissao},
140     React.createElement("h2",{class: "tituloContactos"},"Contacte-nos"),
141
142     React.createElement("div",{class:"divInput"},
143       React.createElement("label",{class:"labelInput"},"Nome: "),
144       React.createElement("input",{ type: 'text', id: "nome", required: true, defaultValue:props.nome},null),
145       React.createElement("p",{ },null)
146     ),
147
148     React.createElement("div",{class:"divInput"},
149       React.createElement("label",{class:"labelInput"},"Email: "),
150       React.createElement("input",{ type: 'email', id: "email", required: true, defaultValue:props.email},null),
151       React.createElement("p",{ },null)
152     ),
153
154     React.createElement("div",{class:"divInput"},
155       React.createElement("label",{class:"labelInput"},"Telefone: "),
156       React.createElement("input",{ type: 'tel', id: "telefone", required: true, defaultValue:props.telefone},null),
157       React.createElement("p",{ },null)
158     ),
159
160     React.createElement("hr",{class:"linhasSimples"},null),
161
162     React.createElement("div",{class:"divInput"},
163       React.createElement("label",{class:"labelInput"},"Motivo: "),
164       React.createElement("input",{ type: 'text', id: "motivo", required: true, defaultValue:props.motivo},null),
165       React.createElement("p",{ },null)
166     ),
167
168     React.createElement("div",{class:"divInput"},
169       React.createElement("label",{class:"labelInput"},"Corpo da mensagem: "),
170       React.createElement("textarea",{ id: "texto", required: true, defaultValue:props.texto},null),
171       React.createElement("p",{ },null)
172     ),
173
174     React.createElement("div",{class:"divInput"},
175       React.createElement("button",{ type: 'submit', id: "enviar", class:"ButaoEnviar"}, "Enviar")
176     )
177   );
178
179 }
```

Figura 19 - Estrutura do Formulário feito com React

```
214 // Obter referência ao elemento root
215 const root = ReactDOM.createRoot(document.getElementById("conteudo"));
216
217 const output = validarDados(localStorage.getItem("nome"), localStorage.getItem("email"),
218 localStorage.getItem("telefone"), localStorage.getItem("motivo"), localStorage.getItem("texto"));
219
220 //Validar dados guardados
221 if(!output)
222 {
223     console.log("Criar formulário a partir de local storage");
224     form = formulario_localstorage();
225 }
226
227 else
228 {
229     console.warn("Criar formulário vazio!");
230     console.error(output)
231     form = formulario_vazio();
232
233     //não é necessário, mas apaga-se a base de dados caso exista algum dado errado,
234     // porque os dados não vão ser introduzidos de qualquer maneira
235     localStorage.clear();
236     console.warn("Local storage limpa!")
237 }
238
239 // Renderizar o elemento React no DOM
240 root.render(form);
241
```

Figura 20 - Código que preenche a página com o Formulário

5. Conclusões

A realização deste projeto foi considerada um sucesso. Durante o seu desenvolvimento, graças a uma ideia sólida sobre o que e como implementar, surgiram poucos problemas. A boa comunicação entre os elementos do grupo contribuiu para aumentar a qualidade do projeto, tanto ao nível das funcionalidades, como das escolhas de design e da experiência do utilizador.

A conclusão do trabalho revelou-se muito prática, permitindo aprofundar os conhecimentos em HTML e CSS, que, combinados com JavaScript e React, tornaram o projeto mais dinâmico e melhoraram significativamente a experiência do utilizador.

6. Bibliografia

Todas as informações teóricas presentes neste relatório foram pesquisadas nos diapositivos disponibilizados para a Unidade Curricular de Aplicações para a Internet I, no Moodle.