

RELATÓRIO

Trabalho Prático

Introdução

O nosso *website*, dedicado a amantes da culinária, faz referência, não só a várias receitas populares de Portugal, Brasil e Itália, como também a chefes de renome.

Para a sua construção foi utilizado **HTML (HyperText Markup Language)**, sendo este o esqueleto da **World Wide Web**, permitindo aos programadores de *web* estruturar e apresentar conteúdo na *internet*. O **HTML** usa uma série de marcas, que permitem definir a estrutura e conteúdo da página *web*. Estas marcas descrevem os diferentes elementos constituintes da página. É também possível acrescentar atributos a estas marcas, que fornecem informação adicional aos elementos. Por exemplo, o atributo *alt* pode ser usado para descrever uma imagem e *href* para definir um *link*. É uma tecnologia em evolução, adicionando ao longo dos anos novas funcionalidades, ultimamente com o **HTML5**.

Por sua vez, é usado **CSS (Cascading Style Sheets)**, um conjunto de estilos com o propósito de alterar a apresentação e formato das páginas *web*. Permite aos programadores *web*, separar a estrutura e conteúdo da página *web* do seu aspecto visual, facilitando a manutenção de páginas *web* complexas. **CSS** funciona com definição de estilos em diferentes elementos de uma página *web*, tais como, títulos, parágrafos, hiperligações e imagens. Estes estilos podem incluir propriedades como, por exemplo, a cor, o tamanho, o espaçamento e a posição, e pode ser aplicado a um só elemento ou então a um grupo. Uma funcionalidade a realçar, é o facto de permitir criar estilos consistentes para todas as páginas *web*, podendo esses estilos, serem responsivos, ou seja, adaptar o seu formato de acordo com o tamanho e características do ecrã. Com a evolução do **CSS** e a introdução do **CSS3**, foram introduzidas novas funcionalidades, uma delas, a *grid*, utilizada nas nossas páginas *web*.

Adicionalmente, para tornar algum conteúdo dinâmico e interativo, recorreu-se a uma linguagem de programação, o **Javascript**. Esta linguagem é bastante utilizada em conjunto com **HTML** e **CSS**, de forma a aumentar a interatividade, funcionalidades e atualização de conteúdo em tempo real. O **Javascript** tem a vantagem de poder manipular o **DOM (Document Object Model)** de uma página *web*. O **DOM** é uma estrutura que representa uma página *web*, deste modo, é possível modificar, adicionar e remover elementos **HTML** de forma dinâmica. Esta ferramenta também pode ser utilizada para interagir com **API's**, que são interfaces que permitem a aplicações *web* comunicar com outros serviços. Outro aspecto importante do **Javascript** é a possibilidade de poder gerar eventos, implementados usando “**event listeners**”, e desta maneira, detectar e responder a ações por parte dos utilizadores é exequível, transformando a experiência do utilizador ao tornar o conteúdo interativo.

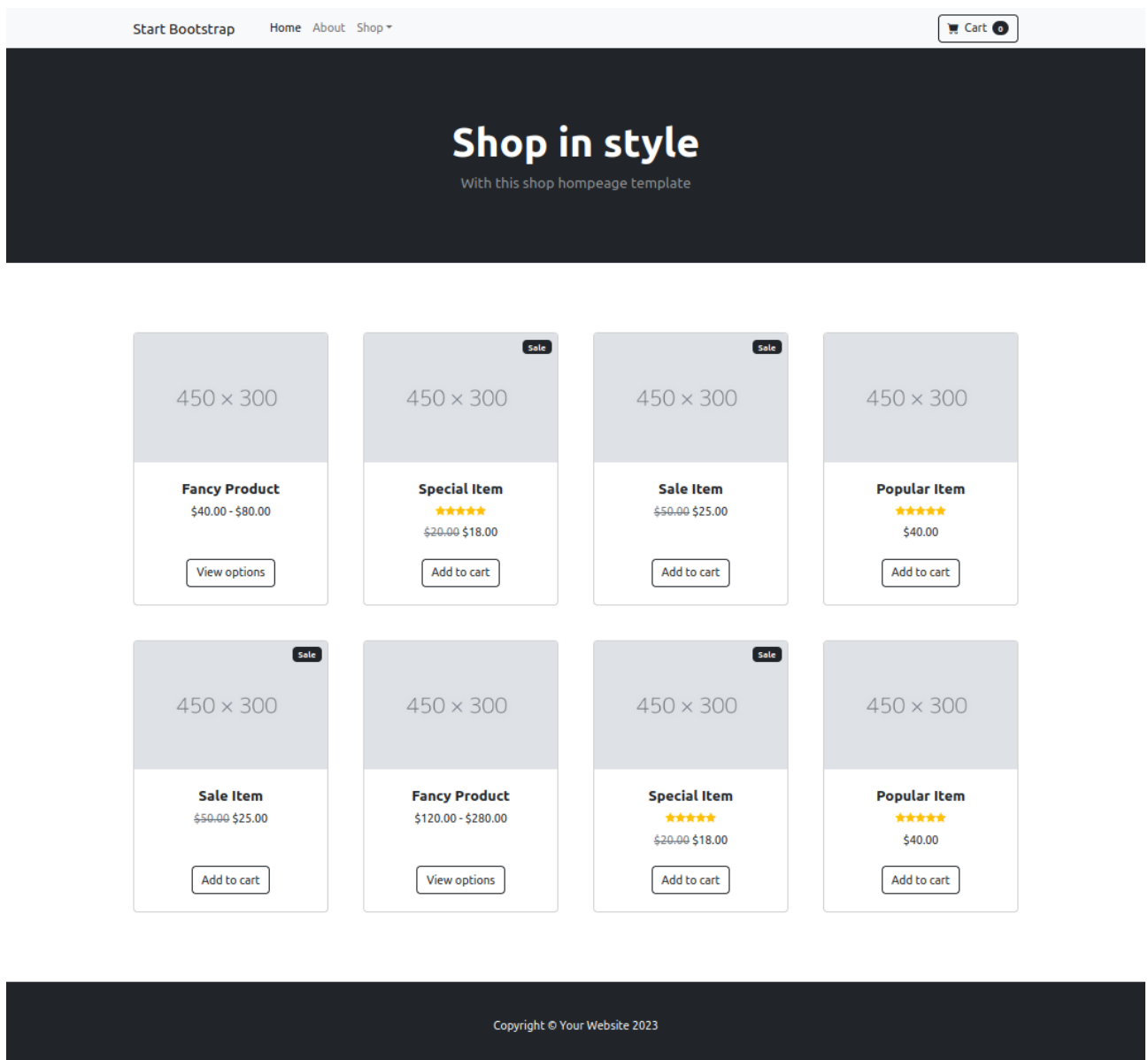
O conteúdo desenvolvido, apresentado nas próximas páginas *web*, é a prova do poder e flexibilidade do **HTML**, **CSS** e **Javascript**.

Link de acesso: <https://laurarccarreira.github.io/>

Templates

Após ter sido feita uma pesquisa na página <https://startbootstrap.com/>, concluiu-se que a *template* “Shop Homepage” seria a mais adequada tendo em conta o tema escolhido para o *website*, visto que os cartões seriam a melhor forma de expor receitas numa página. Uma vez que a *template* é extremamente simples, foram apenas aproveitados os cartões, a *navbar* e o *footer*, sendo o restante código implementado de raiz ou com o auxílio da biblioteca **Bootstrap5**.

Template utilizada: <https://startbootstrap.com/template/shop-homepage>

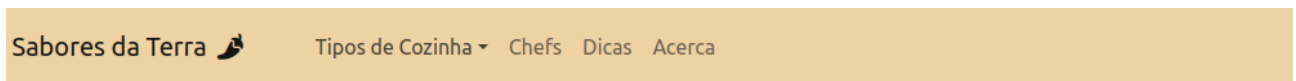


Trabalho Realizado sobre o Site

Conteúdo homólogo

Aqui será referido o conteúdo pertencente à estrutura de todas as páginas *web*, que por sua vez é inteiramente igual.

- **Navbar**



A *navbar* implementada teve como modelo base a *navbar* fornecida na *template* supramencionada.

A cor de fundo foi alterada para “#edd2a4”, uma das cores definida para a paleta a usar no *website* e foram adicionados o nome do website (Sabores da Terra), seguido de um ícone de uma malagueta (retirado da biblioteca de ícones do **Font Awesome**).

De forma a mostrar na *navbar* quais os tipos de cozinha mencionados no *website*, foi utilizado o *dropdown* da *template* e, no **CSS** correspondente aos itens da mesma, o código foi alterado de forma a que, quando ativo, a *background-color* do item seja “#edd2a4”.

```
.dropdown-item:active{
  background-color: #edd2a4 !important;
  color: #000000E6 !important;
  box-shadow: inset 1px 1px 4px 0px;
}
```

Foram também adicionados à *navbar* as restantes páginas **HTML** e ligações para as mesmas.

```
<body style="background-color: #f2e0bf ">
  <!-- Navigation-->
  <nav class="navbar navbar-expand-lg navbar-light" style="background-color: #edd2a4;">
    <div class="container px-4 px-lg-5">
      <a class="navbar-brand" href="index.html">Sabores da Terra<i class='fas fa-pepper-hot m-2'></i></a>
      <button class="navbar-toggler" type="button" data-bs-toggle="collapse"
        data-bs-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false"
        aria-label="Toggle navigation"><span class="navbar-toggler-icon"></span></button>
      <div class="collapse navbar-collapse" id="navbarSupportedContent">
        <ul class="navbar-nav me-auto mb-2 mb-lg-0 ms-lg-4">
          <li class="nav-item dropdown">
            <a class="nav-link dropdown-toggle" id="navbarDropdown" href="#" role="button"
              data-bs-toggle="dropdown" aria-expanded="false">Tipos de Cozinha</a>
            <ul class="dropdown-menu" aria-labelledby="navbarDropdown">
              <li><a class="dropdown-item" href="/src/portuguesa.html">Portuguesa</a></li>
              <li><a class="dropdown-item" href="/src/brasileira.html">Brasileira</a></li>
              <li><a class="dropdown-item" href="/src/italiana.html">Italiana</a></li>
            </ul>
          </li>
          <li class="nav-item"><a class="nav-link" href="/src/chefs.html">Chefs</a></li>
          <li class="nav-item"><a class="nav-link" href="/src/dicas.html">Dicas</a></li>
          <li class="nav-item"><a class="nav-link" href="/src/about.html">Acerca</a></li>
        </ul>
      </div>
    </div>
  </nav>
```

Dependendo da página em que estamos, será adicionado, a um dos elementos interativos, a classe **active** (gerada pelo **Bootstrap5**) que irá dar ênfase ao conteúdo desse elemento (que corresponde ao nome da página em que estamos). Apesar desta classe já ter **CSS** vindo do **Bootstrap5**, adicionamos um peso à sua fonte para dar destaque à palavra e tornar a sua distinção mais fácil para o utilizador.

```
<li class="nav-item"><a class="nav-link active" href="./dicas.html">Dicas</a></li>
```

```
.navbar * .active{  
  font-weight: 500 !important;  
}
```

- **Footer**



Em relação ao *footer*, este também foi reutilizado a partir da *template* escolhida, tendo sido adicionados 4 ícones, de redes sociais, da biblioteca de ícones do **Bootstrap5** e, a cada um destes ícones, foi associada a hiperligação correspondente à rede social do famoso chef Gordon Ramsay. O espaçamento entre cada ícone foi feito utilizando a classe **ms**.

```
<!-- Footer -->  
<footer class="py-5 text-center">  
  <div class="container">  
    <a href="https://www.instagram.com/gordongram/"><i class="bi bi-instagram footer-icons"></i></a>  
    <a href="https://twitter.com/gordonramsay" class="ms-1"><i class="bi bi-twitter footer-icons"></i></a>  
    <a href="https://www.facebook.com/gordonramsay" class="ms-1"><i class="bi bi-facebook footer-icons"></i></a>  
    <a href="https://www.youtube.com/channel/UCIEv3lZ tNXHzL3ox- uUGQ" class="ms-1"><i  
      class="bi bi-youtube footer-icons"></i></a>  
    <p class="m-0">Copyright &copy; Sabores da Terra 2023</p>  
  </div>  
</footer>
```

Para alterar a cor dos ícones e do texto adicionado, foi criada uma classe **CSS**.

```
136 /* Footer Icons */  
137 .footer-icons {  
138   color: #563c24;  
139 }
```

- **Outras modificações**

A todos os elementos *button* foi atribuído um efeito por **CSS** e auxílio da pseudo classe **button:active**, que dá a entender que o botão está a ser pressionado através da redução do seu tamanho e geração de uma sombra à volta do elemento.

```
button:active{  
  transform: scale(0.95);  
  box-shadow: grey 0px 0px 15px 3px;  
}
```

Página “index.html”

Esta é a página principal, com o propósito de apresentar, de uma forma ligeira, o conteúdo das restantes páginas *web*.

Inicialmente, foi modificada a divisão introdutória do *template*, adicionando classes do **Bootstrap5**, com o intuito de formatar o texto e estilo da divisão e cujo propósito seria saudar visitantes do *site*. Estas classes foram, **px-lg-5**, que define margens no eixo do x e **my-2**, no eixo do y.

```
<div class="container px-lg-5 my-2 div-index" >
  <div class="text-center">
    <h1 class="display-4 fw-bolder">Bem-vindo!</h1>
    <p class="lead fw-normal mb-0">Aqui irás encontrar as receitas mais deliciosas
      de várias
      cozinhas mundiais</p>
  </div>
</div>
```

Também foi adicionada a classe **.div-index** que altera a cor de fundo e texto.

```
.div-index {
  background-color: #edd2a4;
  color: #563c24;
}
```

De seguida, de forma a apresentar mesas típicas de alguns países, tais como, Portugal, Brasil e Itália. Foi introduzida a classe **carousel slide** do **Bootstrap5**, assim como outras classes para definir margens, **mx-auto** e **my-5**, definindo margens no eixo do x e y, respectivamente.

Foram ainda adicionadas outras classes, de forma a esterilizar e formatar os elementos contidos na **<div>** referente ao carrossel, nomeadamente, **d-block**, **w-50**, **mx-auto** e **rounded-2**, com a função de, alinhar, definir comprimento, delimitar margens e atribuir um estilo arredondado no canto das imagens, respectivamente. Com recurso à classe **carousel-control-prev/next**, são implementados botões, possibilitando a visualização de outra imagem. No entanto, é expectável que não seja necessário interagir com o carrossel para ver todas as imagens possíveis, e portanto, a classe já inclui este intervalo para fazer uma transição. Segundo a documentação do **Bootstrap5**, é possível alterar este intervalo através de **Javascript**.

```
const myCarouselElement = document.querySelector('#myCarousel')

const carousel = new bootstrap.Carousel(myCarouselElement, {
  interval: 2000,
  touch: false
})
```

```

<div id="carouselExampleAutoplaying" class="carousel slide mx-auto my-5" data-bs-ride="carousel">
  <div class="carousel-inner">
    <div class="carousel-item active">
      
    </div>
    <div class="carousel-item">
      
    </div>
    <div class="carousel-item">
      
    </div>
  </div>
  <button class="carousel-control-prev" type="button" data-bs-target="#carouselExampleAutoplaying"
    data-bs-slide="prev">
    <i class="fa fa-angle-left fa-3x" aria-hidden="true"></i>
    <span class="visually-hidden">Previous</span>
  </button>
  <button class="carousel-control-next" type="button" data-bs-target="#carouselExampleAutoplaying"
    data-bs-slide="next">
    <i class="fa fa-angle-right fa-3x" aria-hidden="true"></i>
    <span class="visually-hidden">Next</span>
  </button>
</div>

```

De maneira a dividir cada secção, apresentando um título e descrição para cada uma, foram usadas as classes **py-3**, **text-center** na **<div>** mais abrangente e, na mais interior, **px-lg-5** e **my-4**, tudo classes do **Bootstrap5**, usadas para detalhes relacionados com a posição, margens e preenchimento do texto.

```

<div class="py-3 text-center div-index">
  <div class="container px-lg-5 my-4">
    <h1 class="display-6 fw-bolder">Encontra um restaurante</h1>
    <p class="lead fw-normal mb-0">Melhor do que seguir receitas, é ter a papinha toda feita num
      restaurante à tua escolha</p>
  </div>
</div>

```

Por último, nesta página, foi adicionado uma mapa para mostrar alguns restaurantes de uma determinada zona. Para este mapa recorre-se a uma biblioteca de **Javascript**, a **Leaflet** que implementa a **API** do **OpenStreetmap**. Porém, este conteúdo também precisa de ter uma certa disposição na página *web*, para estar em conformidade com os restantes elementos e, por isso, foi acrescentada uma **<section>** com uma classe **py-5**, do **Bootstrap5**, e uma classe criada, **.map-section**.

```

<section class="py-5 map-section">
  <div id="map"></div>
</section>

```

```

.map-section {
  display: flex;
  justify-content: center;
  align-items: center;
}

```

Para mudar o tamanho e margem do mapa apresentado, foi feita em CSS a alteração do estilo através da seleção por “id” (neste caso, id=”map”) da **<div>** em que este se encontra.

```
#map {
  width: 700px;
  height: 600px;
  margin: auto
}
```

Portanto, para instanciar o mapa e definir a janela de visualização, recorre-se à seguinte instrução:

```
// Create a Leaflet map object
var map = L.map('map').setView([41.14278390310927, -8.612213735423238], 17);
```

Encontrar a forma mais intuitiva e perceptível para mostrar todos os restaurantes foi um objetivo e, portanto, personalizou-se um *icon*, recorrendo a uma biblioteca, a **Flaticon**.

```
var myIcon = L.icon({
  iconUrl: './assets/icons/restaurant.png',
  iconSize: [30, 30],
  iconAnchor: [15, 35],
  popupAnchor: [0, -30]
});
```

De modo a que este mapa funcione é necessário comunicar com a **API** do **OpenStreetMap**.

```
L.tileLayer('https://tile.openstreetmap.org/{z}/{x}/{y}.png', {}).addTo(map);
```

Assim, para ter informação para mostrar recorreu-se a um serviço exterior, o **JSOMBIN.io** onde foi alojado um ficheiro **Json**, com as coordenadas de todos os restaurantes.

```
$.getJSON("https://api.jsonbin.io/v3/b/64442b4b8e4aa6225e8ec932", function (data) {
  var restaurants = L.geoJSON(data.record.features, {
    pointToLayer: function (feature, latlng) {
      return L.marker(latlng, {
        icon: myIcon
      });
    },
  }).addTo(map);
});
```

É importante referir que houve mais conteúdo introduzido nesta página web, no entanto, é aproveitado o facto de ser o conteúdo principal e bastante essencial para o propósito da página “Tipos de cozinha e, portanto, será referido com mais detalhe nessa mesma secção.

Páginas na secção “Tipos de cozinha”

O objetivo deste conjunto de páginas é expor receitas típicas de três países diferentes e, para tal, consideraram-se os cartões disponíveis na *template* referida anteriormente como a melhor solução a implementar.

Além da *navbar* e do *footer* (comuns a todas as páginas), os principais elementos das páginas **portuguesa.html**, **brasileira.html** e **italiana.html** são um *header* e uma *section*.

O *header* foi implementado de raiz e é composto por um **<h1>** que indica o tipo de cozinha ao qual a página se refere, uma pequena descrição e introdução (**<p>**) e um ícone com a bandeira do país ao qual pertence o tipo de cozinha.

```
<!-- Header-->
<header class="py-4" style="background-color: #edd2a4;">
  <div class="container px-4 px-lg-5 my-2 div-cozinha">
    <div class="text-center">
      <h1 class="display-4 fw-bolder">
        Cozinha Portuguesa
      </h1>
      <p class="lead fw-normal mb-0">Com influências atlânticas e mediterrâneas, estas receitas prometem ser
        <br> do melhor que a Europa tem para oferecer
      </p>
      
    </div>
  </div>
</header>
```

O código relativo à *section* foi parcialmente escrito a partir do código da *template* e cada cartão foi modificado de forma a mostrar uma imagem da receita em questão, o nome da mesma, o tempo de confecção e de preparação, a dificuldade de execução e o número de doses para a qual está destinada. Todas estas informações foram complementadas visualmente com recurso aos ícones disponibilizados pela biblioteca de ícones do **Bootstrap5**.

```
<!-- Section-->
<section class="py-5">
  <div class="container px-4 px-lg-5 mt-5">
    <div class="row gx-4 gx-lg-5 row-cols-2 row-cols-md-3 row-cols-xl-4 justify-content-center">
      <div class="col mb-5">
        <div class="card h-100 item">
          <!-- Product image-->
          
          <!-- Product details-->
          <div class="card-body p-4">
            <div class="text-center">
              <!-- Product name-->
              <h5 class="fw-bolder">Arroz de Pato</h5>
              <!-- Product details-->
              <ul class="item-details">
                <li>01:00 <i class="bi bi-clock-fill"></i></li>
                <li>
                  
                  
                  
                </li>
                <li>6 <i class="bi bi-person-fill"></i></li>
              </ul>
            </div>
          </div>
        </div>
      </div>
```


Importa salientar que todas as alterações feitas que estejam relacionadas com a posição relativa do elemento foram feitas, de raiz, utilizando classes de **Bootstrap5**, tais como, **px**, **mb**, **mx** e **my**.

Ainda em relação ao cartão, este possui um botão “Ver Receita” que, quando pressionado, permite abrir um *modal* e, com recurso à classe **.card.item: hover**, implementada de raiz, quando existe um evento *mouseover*, o cartão sobressai em relação aos restantes.

```
.card.item: hover {
  animation: card_zoom 300ms forwards;
}
```

O *modal* foi implementado a partir do código exemplo presente na documentação do **Bootstrap5** e o seu objetivo é mostrar os ingredientes necessários e o modo de preparo da receita.

Para a divisão relativa aos ingredientes, foram criadas as classes **CSS** **.ingredientes-receita**, cujo objetivo é criar uma grelha de 12 colunas e **.is-6**, que formata a lista de ingredientes de forma a que cada elemento ocupe 6 colunas. Para a divisão do modo de preparo, foi criada a classe **.preparo-receita**, que formata o texto, justificando-o.

```
.preparo-receita li{
  text-align: justify;
}

/* Cozinha Modals */
.ingredientes-receita {
  display: grid;
  grid-template-columns: repeat(12, 1fr);
}

.is-6 {
  grid-column: auto/span 6;
}
```

Página “chefs.html”

Esta página foi criada com o objetivo de introduzir informação sobre 3 *chefs* aclamados, tendo sido escolhido um de cada nacionalidade, de acordo com os tipos de cozinha apresentados.

Quanto à introdução de conteúdo dentro do *body* da página, foram criadas as classes **.chef-intro** para ajuste do *display* do conteúdo, justificação do texto e alinhamento do conteúdo, **.chef-picture: hover**, **.chef-intro img** e **.chef-container p**, explicadas mais à frente.

A imagem seguinte mostra unicamente o código **HTML** referente à introdução da imagem e respectiva bibliografia do chef Henrique Sá Pessoa. Contudo, foi usado o mesmo código para os restantes chefes, mudando apenas a imagem introduzida e o texto escrito.

```

<div class="chef-intro">
  <div class="chef-picture">
    <a href="https://www.almalisboa.pt/pt" target="_blank">
      
    </a>
  </div>
  <div class="chef-info">
    <h3>Henrique Sá Pessoa</h3>
    <p>Henrique Sá Pessoa nascido em Oeiras (Portugal) em 1976 é um premiado chef de cozinha português laureado com duas estrelas Michelin reconhecido pelo seu trabalho no restaurante Alma. Sá Pessoa faz parte da chamada revolução da alta gastronomia portuguesa, sendo o terceiro chef português na história a ser reconhecido com duas estrelas Michelin.</p>
  </div>

```

Analisando o código HTML mostrado pela figura acima, é evidente que foi criada uma hiperligação direta através da imagem para o site de um dos restaurantes de cada um dos chefs. No caso, pretendia-se que o site fosse aberto numa nova janela e, por isso, recorreu-se ao uso do atributo `target="_blank"`.

Classe **.chef-picture: hover**, utilizada para reduzir a opacidade da imagem do chef aquando um evento de *mouseover*, conferindo uma sensação de interação mais dinâmica:

```

.chef-picture: hover {
  opacity: 0.60;
}

```

Classe **.chef-container p**, cujos objetivos principais são a formatação, ajuste e alinhamento do conteúdo dentro do `<p>` referente à apresentação de algumas informações acerca do chef:

```

/* Chefs */
.chef-container p {
  display: flex;
  flex-direction: column;
  align-items: center;
  justify-content: center;
  gap: 20px;
  text-align: justify;
}

```

Em relação à alteração do aspeto visual da imagem, definiu-se um bordo circular através da implementação da classe **.chef-intro img**:

```

.chef-intro img {
  border-radius: 100%;
}

```

Além disso, foram criados 3 cartões idênticos aos utilizados na página “Tipos de Cozinha.html”, com o objetivo de introduzir 3 das receitas mais famosas de cada um dos chefs. Como referido, estes cartões são iguais aos utilizados anteriormente e, por isso, as únicas mudanças a apontar são a mudança das imagens usadas, o tempo de preparação das receitas, a sua dificuldade (indicada pelo número de chapéus de cozinheiro), o número de doses e o nome da receita.

Foi também usado um *modal* (aberto ao clicar no botão “Ver Receita”), para a descrição da receita e dos ingredientes necessários. Assim, tanto o código HTML como o CSS usados foram iguais, alterando-se apenas o seu conteúdo em texto (em HTML).

Página “dicas.html”

Esta página possui conteúdo completamente novo, com exceção da *navbar*, header, e footer que utilizam os mesmos elementos e classes de outras páginas. Assim sendo, comecemos por introduzir a **<div>**.

```
<div class="search-dicas text-center mt-4">
  <input class="form-control-sm search-input" type="text" id="dicas-filter" list="datalist-dicas"
    placeholder="Palavra-Chave">
  <button class="search-btn" id="search-dicas">Procurar <i class="bi bi-search mx-1"></i></button>
  <datalist id="datalist-dicas">
  </datalist>
</div>
```

Com a classe **.search-dicas** e as classes de **Bootstrap 5** “**text-center mt-4**”, os elementos no seu interior ficam centrados na página, denominadamente *input* e *button*, e é atribuída uma pequena distância entre os mesmos.

```
/* Dicas e Truques */
.search-dicas {
  display: flex;
  justify-content: center;
  align-items: center;
  gap: 10px;
  margin-bottom: 30px;
}
```

Quanto aos elementos no interior da **<div>**, o *input* contém a classe **.form-control** (Bootstrap 5) e o *button* contém a classe **.search-btn**, que visa atribuir-lhe uma estrutura apelativa, com a combinação de cores da paleta utilizada e, atribuir mudanças, através da pseudo classe **:active**, como a cor do background-color, que ocorrem com a ativação do elemento.

```
.search-btn {
  border: 1px solid #563c24;
  border-radius: 8px;
  padding: 5px 10px;
  background-color: #f2e0bf;
}

.search-btn:hover {
  background-color: #563c24;
  color: white;
  transition-duration: 400ms;
}
```

A *datalist* não contém nenhuma classe, pois o seu único propósito é armazenar o conteúdo que será disponibilizado no *autocomplete* do elemento *input*. Essa pesquisa no *datalist* é feita através do atributo **list='datalist-dicas'**, presente no *input*, que vai apontar para a *datalist* que contém esse *id*, ou seja **id='datalist-dicas'**. Para terminar, ao aspecto visual desta primeira divisão, foi adicionado um ícone de lupa do **Bootstrap 5**, que dá o significado de procura ao *button* existente.

A seguir ao elemento **<div>**, encontramos o **<article>**.

```
<article>
  <dl>
    <dt id="dt1">1</dt>
    <dd class="dicas-list" value="dt1">Para evitar que os ovos rachem durante a cozedura, adicione uma colher de chá de vinagre à água antes de ferver.</dd>
```

No interior de **<article>**, podemos encontrar um único elemento, **<dl>**, e vários elementos **<dt>** e **<dd>** no seu interior. **<dl>** representa o nome *description-list*, pois o seu interior constitui o conteúdo que representa uma lista de descrições acerca de algo sendo, neste caso, uma descrição das várias dicas que disponibilizamos. **<dd>** terá o conteúdo da dica e **<dl>** irá representar o número da sua ordem na lista. Já a classe **.dicas-list**, onde se encontram os elementos **<dd>**, tem o objetivo de auxiliar o javascript utilizado para manipular os dados.

Para a disponibilização do conteúdo, foi atribuído, diretamente aos elementos e, através de **CSS**, o *display* da **<dl>** como *grelha*, com um pequeno espaçamento entre o conteúdo que nele está inserido. Nos restantes elementos, foi atribuído **CSS** apenas para auxiliar a estética e visualização do conteúdo (*padding*, *background*, etc).

```
dl {
  display: grid;
  grid-template-columns: auto 1fr;
  gap: 0.5rem 0;
}

dl > * {
  background: #antiquewhite;
  padding: 1rem;
}

dl > :nth-of-type(2n) {
  background: #rgb(255, 250, 244);
}

dt {
  font-weight: 700 ;
  font-size: 1.8rem ;
}

dd {
  margin: 0 !important;
  padding: 1rem;
  display: flex;
  align-items: center;
}
```

Como últimas adições nesta página, apresentaremos uma breve síntese do funcionamento do JavaScript utilizado:

```
// Autocomplete Options
input_field.addEventListener('keyup', function(){
  const datalist = document.getElementById('datalist-dicas');
  const inputValue = input_field.value.toLowerCase();
  const options = datalist.getElementsByTagName('option');
  const array_options = Array.from(options);
  array_options.forEach(element => {
    element.remove();
  });
  ingredientes.forEach(word => {
    if(word.toLowerCase().startsWith(inputValue) && inputValue !== ""){
      let childElement = document.createElement('option');
      childElement.value = word;
      datalist_dicas.appendChild(childElement);
    }
  });
});
```

Este código permite que os utilizadores de um site possam pesquisar dicas de culinária, usando um recurso de preenchimento automático. É feita a comparação da palavra inserida pelo utilizador com uma lista de ingredientes e são criadas opções de preenchimento automático com base nos ingredientes correspondentes. Quando o utilizador seleciona uma das opções, o valor correspondente é inserido no campo de entrada. Isso ajuda o utilizador a pesquisar as dicas de culinária com mais facilidade e rapidez, sem precisar de escrever o nome completo do ingrediente.

Foram também adicionados eventos ao *input* e *button* que, através da pesquisa, disponibilizam as dicas que contêm o ingrediente escrito pelo utilizador. Por último, foi utilizada a função **addDicasEvent()**, que atribui às dicas um evento, cujo funcionamento ocorre no seu clique e disponibiliza todas as dicas que contêm ingredientes relacionados.

```
function addDicasEvent(){
  let list_dicas = document.getElementsByClassName('dicas-list');
  let array_dicas = Array.from(list_dicas);
  array_dicas.forEach(element => {
    element.addEventListener('click', function(){
      let ingredienteToSearch = [];
      ingredientes.forEach(ingrediente => {
        if(element.textContent.includes(ingrediente)){
          ingredienteToSearch.push(ingrediente);
        }
      });
      let i = 0;
      let search_result = '';
      ddElements.forEach((dd) => {
        let display = 0;
        ingredienteToSearch.forEach(ingrediente => {
          if (dd.textContent.includes(ingrediente) && display == 0) {
            i++;
            search_result += `<dt>${i}</dt> <dd class="dicas-list">${dd.textContent}</dd> `;
            display = 1;
          }
        })
      });
      dlElement.innerHTML = search_result;
      addDicasEvent();
    })
  });
}
```

Para auxiliar o utilizador e dar a percepção da possibilidade de pesquisa por clique na dica, adicionamos **CSS** às dicas, com ajuda das pseudo classes **:hover** e **:active**. Dentro da pseudo classe **active**, fizemos a adição de uma animação cuja função (através da **@keyframes**) será provocar uma translação do elemento, bem como uma sombra à sua volta.

```
dl > dd:hover:active{
  transform: scale(0.99) !important;
}
dl > dd:hover {
  animation: go_right 300ms forwards;
  cursor: pointer;
}
@keyframes go_right {
  100%{
    transform: translate(10px,0%);
    box-shadow: 0px 0px 5px 0px;
  }
}
```

Página “acerca.html”

O conteúdo desta página, pode ser separado em três secções principais, cada uma delas precedida de uma **<div>** com o seu tópico.

```
<div class="my-3 align-items-center" style="width: 700px;">
  <ul class="list-group" style="--bs-list-group-bg: #f2e0bf; --bs-list-group-color: #563c24;">
    <li class="list-group-item text-center" sty>
      <h5>Afonso Baixo</h5>
    </li>
    <li class="list-group-item text-center">
      <h5>Bárbara Moreira</h5>
    </li>
    <li class="list-group-item text-center">
      <h5>Laura Carreira</h5>
    </li>
    <li class="list-group-item text-center">
      <h5>Luís Leal</h5>
    </li>
    <li class="list-group-item text-center">
      <h5>Paulo Macedo</h5>
    </li>
  </ul>
</div>
```

A **<div>** apresentada é formatada com o recurso a duas classes do **Bootstrap5**, **my-3** e **align-items-center**, permitindo estabelecer uma margem e alinhar conteúdo. O comprimento da divisão também é alterado através do atributo de **CSS** “width”. No interior, é definida uma lista com recurso à classe **list-group**, também do **Bootstrap5**, sendo alterada nesta lista, a cor de texto e também cor de fundo, com o auxílio dos atributos de **CSS**, “**--bs-list-group-color**” e “**--bs-list-group-gb**”, respectivamente. Os nomes dos membros do grupo são centrados e definidos com um heading **<h5>**.

A listagem dos cursos efetuou-se de igual forma.

```
<div class="my-3 align-items-center" style="width: 700px;">
  <ul class="list-group" style="--bs-list-group-bg: #f2e0bf; --bs-list-group-color: #563c24;">
    <li class="list-group-item text-center" style="text-align: center;">
      <h5>Engenharia Biomédica</h5>
    </li>
    <li class="list-group-item text-center" style="text-align: center;">
      <h5>Engenharia de Computadores e Informática</h5>
    </li>
  </ul>
</div>
```

As alterações a nível de **CSS** incidem principalmente no próprio elemento ****. Para além das propriedades já fornecidas pela **Bootstrap5**, acrescentamos aquelas que considerámos mais relevantes para o embelezamento do conteúdo.

Além disso, destacamos, no final desta página, um mapa marcado com as coordenadas geográficas dos respectivos departamentos a que cada curso pertence. Para este efeito, recorremos à biblioteca **Leaflet.js**, para a criação do próprio mapa e interação com o mesmo.

No ficheiro **about.js**, começámos por inicializar o próprio mapa, e de seguida, com a especificação da região que pretendemos apresentar através da função **.setView()**.

```
var map = L.map('map').setView([40.63189212093659, -8.65822760177182], 17);
```

Tendo em conta que se podia representar cada departamento de uma forma mais direta, foi introduzido um *icon* da biblioteca **FlatIcon**, da seguinte forma:

```
var myIcon = L.icon({
  iconUrl: '../assets/icons/university.png',
  iconSize: [40, 40],
  iconAnchor: [28, 35],
  popupAnchor: [0, -40]
});
```

Para a criação dos marcadores, fizemos-nos valer da função **L.marker()** e, para adição dos mesmos às coordenadas pretendidas, da função **addTo()**.

```
L.marker([40.63327715251739, -8.659529757486075],
  { icon: myIcon }
).addTo(map)
L.marker([40.630261824695495, -8.65669602425622],
  { icon: myIcon }
).addTo(map)
```

Conclusões

Sabendo que este projeto tinha como objetivo a consolidação dos conteúdos lecionados nas aulas de PW, nomeadamente no que toca à utilização de **HTML**, **CSS** e **Bootstrap5**, consideramos que este foi atingido com sucesso através da criação do nosso site “Sabores da Terra”. Isto porque, como evidenciado ao longo deste relatório, foram inseridos, removidos e alterados diversos elementos na estrutura do template escolhido, através de **HTML**, e procedeu-se à sua formatação através de **CSS**.

Apesar de não ter sido lecionado em aula, temos noção da importância que o **JavaScript** tem na dinamização de uma página *web* e, por isso, decidimos implementar também esta linguagem de programação nas nossas páginas.

Além disso, sabemos que estas linguagens de programação e de estilo permitem o desenvolvimento de páginas *web* bastante mais complexas do que aquela que foi criada no âmbito deste módulo. Desta forma, consideramos que haveria sempre espaço para melhorar o nosso trabalho.

Referências

Html & CSS:

- <https://colorhunt.co/palettes/food>
- <https://getbootstrap.com/docs/5.3/getting-started/introduction/>
- <https://freefrontend.com/bootstrap-lists/>

Javascript:

- <https://jquery.com/>
- <https://leafletjs.com/>

Icons:

- <https://www.flaticon.com/>
- <https://fontawesome.com/>
- <https://icons.getbootstrap.com/>

Receitas & Chefs:

- <https://www.pingodoce.pt/receitas/>
- <https://pt.petitchef.com/>
- <https://www.24kitchen.pt/receitas>
- <https://www.tudogostoso.com.br>
- <https://www.24kitchen.pt/chefs/henrique-sa-pessoa>
- <https://www.almalisboa.pt/pt>
- <https://www.alexatala.com.br/en/home/>
- <https://tanamesa.com/receita/>
- <http://domrestaurante.com.br/pt-br/home.html>
- <https://www.greatitalianchefs.com/chefs/nadia-santini>
- <https://www.four-magazine.com/chefs/nadia-santini/>

- <https://www.ktchnrebel.com/legend-nadia-santini/>
- <https://www.falstaff.com/en/recipes/cooking/risotto-alla-milanese-with-artichokes>
- <https://chefsimon.com/gourmets/mike-errecart/recettes/la-pasta-al-pomodoro-de-nadia-santini>
- <https://www.dalpescatore.com/it>

Outros:

- <https://www.openstreetmap.org/>
- <https://jsonbin.io/>