

Curso Técnico Superior Profissional em Tecnologias e Programação de Sistemas de Informação

Unidade Curricular: Design Hipermedia

1.º Ano/1.º Semestre

Docente: Marco Miguel Olival Olim

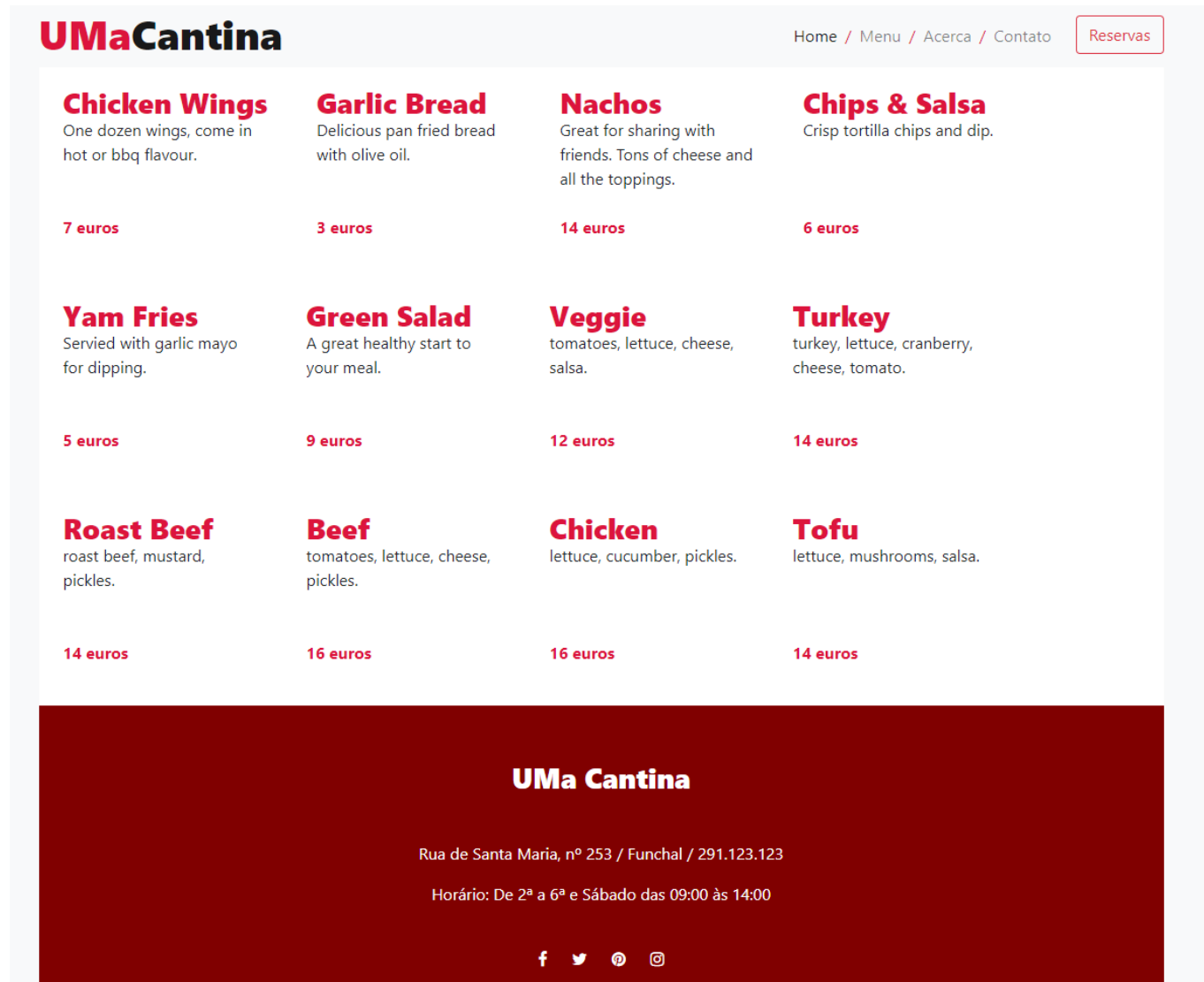
Data 04/12/2017

ESTE EXERCÍCIO PRETENDE EXPLORAR AS DIFERENTES ABORDAGENS À UTILIZAÇÃO DO AJAX COM O JQUERY

Como foi já referido no exercício anterior, o Ajax (Asynchronous JavaScript and XML) permite obter dados do servidor sem que seja necessário carregar novamente a página de html. Estes dados podem ser representados em XML (eXtensible Markup Language) ou JSON (JavaScript Object Notation) que, para o nosso exemplo da página do menu da Cantina da UMa, seria representado da seguinte forma:

```
[
  {
    "obs": "One dozen wings, come in hot or bbq flavour.",
    "prato": "Chicken Wings",
    "tipo": "appetizer",
    "total": 7
  },
  {
    "obs": "Delicious pan fried bread with olive oil.",
    "prato": "Garlic Bread",
    "tipo": "appetizer",
    "total": 3
  },
  {
    "obs": "Great for sharing with friends. Tons of cheese and all the toppings.",
    "prato": "Nachos",
    "tipo": "appetizer",
    "total": 14
  },
  {
    "obs": "Crisp tortilla chips and dip.",
    "prato": "Chips & Salsa",
    "tipo": "appetizer",
    "total": 6
  }
]
```

A página do menu deste site deverá ter o seguinte aspeto:



Esta página deverá usar os mesmos estilos da página principal mas recomenda-se adicionar os seguinte novos estilos:

```
...
.cor-principal {
  color: crimson;
}

/*novos estilos para o menu*/
.preco {
  font-weight: 700;
}
.obs {
  width: 12em;
  word-wrap: break-word;
  height: 5em;
  overflow: hidden;
}
```

Cofinanciado por:



O html mantém-se essencialmente o mesmo, removendo-se todo conteúdo da tag <section>, acrescentando-se depois o id “cardápio” onde será inserido o conteúdo dinamicamente.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-
beta.2/css/bootstrap.min.css" >
  <link href="https://maxcdn.bootstrapcdn.com/font-awesome/4.7.0/css/font-
awesome.min.css" rel="stylesheet" >
  <link rel="stylesheet" href="UmaCantina.css">
  <title>UMa Cantina - Menu</title>
</head>
<body>
  <header>
    <div class="container">
      <nav class="navbar navbar-expand-md navbar-light bg-light px-0">
        <a class="navbar-brand" href="#">
          <h1><span class="cor-principal">UMa</span>Cantina</h1>
        </a>
        <button class="navbar-toggler" type="button" data-toggle="collapse"
data-target="#navbarNavAltMarkup" aria-controls="navbarNavAltMarkup" aria-
expanded="false" aria-label="Toggle navigation">
          <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarNavAltMarkup">
          <ul class="navbar-nav ml-auto">
            <li class="nav-item active">
              <a href="index.html" class="nav-link">Home</a>
            </li>
            <li class="divider py-2 d-none d-md-block cor-principal"></li>
            <li class="nav-item">
              <a href="menu.html" class="nav-link">Menu</a>
            </li>
            <li class="divider py-2 d-none d-md-block cor-principal"></li>
            <li class="nav-item">
              <a href="about.html" class="nav-link">Acerca</a>
            </li>
            <li class="divider py-2 d-none d-md-block cor-principal"></li>
            <li class="nav-item pr-3">
              <a href="contact.html" class="nav-link">Contato</a>
            </li>
            <li class="nav-item">
              <button type="button" class="btn btn-outline-danger" data-
toggle="modal" data-target="#reservationmodal">Reservas</button>
            </li>
          </ul>
        </div>
      </nav>
    </div>
  </header>
```

Cofinanciado por:



```

<section>
  <div class="container">
    <div class="col-lg-12">
      <div class="row d-flex flex-wrap" id="cardapio">
      </div>
    </div>
  </div>
</section>
<footer>
  <div class="container">
    <div class="col-lg-12 text-center">
      <h3>UMa Cantina</h3>
      <br>
      <br>
      <p>Rua de Santa Maria, nº 253 / Funchal / 291.123.123</p>
      <p>Horário: De 2ª a 6ª e Sábado das 09:00 às 14:00</p>
      <br>
      <div class="social-media">
        <a href="#">
          <i class="fa fa-facebook"></i>
        </a>
        <a href="#">
          <i class="fa fa-twitter"></i>
        </a>
        <a href="#">
          <i class="fa fa-pinterest"></i>
        </a>
        <a href="#">
          <i class="fa fa-instagram"></i>
        </a>
      </div>
      <br>
      <p>&copy; 2017 Universidade da Madeira</p>
    </div>
  </div>
</footer>
<script src="https://code.jquery.com/jquery-3.2.1.min.js"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.3/umd/popper.min.js"></script>
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-
beta.2/js/bootstrap.min.js" ></script>

</body>

</html>

```

Para carregar os dados usamos uma chamada de AJAX usando o método jQuery .getJSON():

```

<script>
  $(document).ready(function() {
    $.getJSON('https://final-project-recording-817f4.firebaseio.com/cardapio/.json');
  });
</script>

```

Cofinanciado por:



Aparentemente este código não produz nenhum efeito. Apesar de serem carregados os dados, falta agora indicar ao JavaScript o que fazer com o resultado. Para isso é necessário usar uma função de callback

```
<script>
    $(document).ready(function() {
        $.getJSON('https://final-project-recording-817f4.firebaseio.com/cardapio/.json', function(data) {
        });
    });
</script>
```

Nesta função seria possível percorrer a lista de elementos com um ciclo for mas o jQuery dispõe de um método de interação. Para cada elemento, inserimos então o respetivo elemento html no id “cardápio”, substituindo tudo o que lá exista:

```
<script>
    $(document).ready(function() {
        $.getJSON('https://final-project-recording-817f4.firebaseio.com/cardapio/.json', function(data) {
            var html = '';
            $.each(data, function(entryIndex, entry) {
                html += '<div class="entry p-4">';
                html += '<h3 class="cor-principal">' + entry.prato + '</h3>';
                html += '<p class="obs">' + entry.obs + '</p>';
                html += '<p class="cor-principal preco">';
                html += entry.total;
                html += ' euros</p>';
                html += '</div>';
            });
            $('#cardapio').html(html);
        });
    });
</script>
```

O jQuery implementa o AJAX para ser consumido unicamente a partir de webserver, não sendo permitida a utilização de ficheiros locais, por uma questão de segurança. Se criasse o ficheiro dados.json com dados json válidos e tentasse aceder a estes (com \$.getJSON('dados.json', function(data) {})) obteria o seguinte resultado na consola do browser:

jquery-3.2.1.min.js:4 Failed to load file:///C:/Users/olimm/Desktop/uma_testes/dados.json: Cross origin requests are only supported for protocol schemes: http, data, chrome, chrome-extension, https.

Cofinanciado por:



Para criar um webserver local podemos utilizar o nodeJS para o efeito, iniciando um novo ficheiro chamado webserver.js

```
const http = require('http');
const url = require('url');
const fs = require('fs');
const path = require('path');
const port = process.argv[2] || 9000;
http.createServer(function(req, res) {
  console.log(`${req.method} ${req.url}`);
  const parsedUrl = url.parse(req.url);
  let pathname = `.${parsedUrl.pathname}`;
  const mimeType = {
    '.html': 'text/html',
    '.json': 'application/json',
    '.css': 'text/css',
    '.png': 'image/png',
    '.jpg': 'image/jpeg',
    '.eot': 'application/vnd.ms-fontobject',
    '.ttf': 'application/font-sfnt'
  };
  fs.exists(pathname, function(exist) {
    if (!exist) {
      res.statusCode = 404;
      res.end(`File ${pathname} not found!`);
      return;
    }
    if (fs.statSync(pathname).isDirectory()) {
      pathname += '/index.html';
    }
    fs.readFile(pathname, function(err, data) {
      if (err) {
        res.statusCode = 500;
        res.end(`Error getting the file: ${err}.`);
      } else {
        const ext = path.parse(pathname).ext;
        res.setHeader('Content-type', mimeType[ext] || 'text/plain');
        res.end(data);
      }
    });
  });
}).listen(parseInt(port));
console.log(`Server listening on port ${port}`);
```

Para executar este script usamos a consola de Sistema:

```
C:\>node webserver.js
```

Cofinanciado por:



No browser já pode aceder ao webserver usando <http://localhost:9000/dados.json> . Pode também agora aceder a estes dados a partir de <http://localhost:9000/menu.html> sem os erros de CORS.

Propõe-se agora a realização dos seguintes exercícios:

1. Da mesma forma que fez para o menu, exclua agora o conteúdo da tag <section> do ficheiro html principal e coloque-o num novo ficheiro partial.html . Carregue agora este ficheiro por AJAX usando o método .load()
2. Modifique os botões da barra de navegação para carregar todo o conteúdo na mesma página (correspondente ao menu, acerca e contato) em vez de multiplicar o mesmo código ineficientemente por diferentes páginas de html

```
<script>
    $(document).ready(function() {
        $('#home').click(function(event) {
            event.preventDefault();
            $('#conteudo').load('partial.html');
        });
    });
</script>
```

3. Separe o menu por tipo, identificando-o com um título
4. Acrescente aos dados.json o link de uma imagem ilustrativa do prato e crie um modal no menu.html para apresentar essa imagem dinamicamente

Cofinanciado por:

