



CookIt



COLÉGIO DE
S.MIGUEL
AMIZADE VERDADE EXIGÊNCIA

CookIt

Prova de Aptidão Tecnológica

Curso Científico-Tecnológico de Informática

Aluno: **Bernardo Ferreira**

Orientador de Projeto: **Paulo Figueira**

2017 / 2018





Índice

Introdução.....	5
Objetivos	6
Aplicação Web.....	7
Software Utilizado	7
WampServer.....	7
Sublime Text.....	7
Tecnologias Web Utilizadas	8
HTML	8
CSS.....	8
JavaScript.....	9
Bootstrap.....	9
jQuery.....	9
AJAX	10
PHP.....	10
mPDF	10
PHPMailer	11
Summernote.....	11
AngularJS.....	11
SQL	11
Desenvolvimento.....	12
Estrutura.....	12
Iniciar Sessão / Criar Conta.....	13
Formulário de Nova Receita	14
Formulário de Editar Receita	16
Página de Apresentação de Receitas	17
Página de Apresentação de uma Receita.....	19
Tabelas da Base de Dados.....	20
Servidor Web.....	23
Materiais Utilizados	23
Raspberry Pi 3.....	23
Cartão Micro SD.....	24
Transformador	24
Software Utilizado	25
Etcher.....	25
PuTTY	25
FileZilla.....	25
Raspbian	25
Apache2	26



CookIt

PHP	26
MySQL	26
phpMyAdmin	26
SSH	26
vsftpd.....	27
Configuração do Servidor.....	27
Aplicação Android.....	30
Software Utilizados	30
Android Studio.....	30
Desenvolvimento.....	31
Recursos Online	32
Conclusão.....	33



Índice de Ilustrações

Ilustração 1 - Logótipo do projeto.....	5
Ilustração 2 - Interface do Sublime Text	8
Ilustração 3 - Esquema do funcionamento de AJAX	10
Ilustração 4 - Esboço da estrutura da página.....	12
Ilustração 5 - Tag do formulário de criar conta	13
Ilustração 6 - Bloco do nome do formulário criar conta	13
Ilustração 7 - Esboço do formulário de nova receita	14
Ilustração 8 - Código de estilo da barra de navegação	15
Ilustração 9 - Bloco da imagem do formulário nova receita.....	15
Ilustração 10 - Função readURL	15
Ilustração 11 - Bloco para ingredientes do formulário nova receita	15
Ilustração 12 - Inicialização do Summernote	16
Ilustração 13 - Esboço da página de apresentação de receitas	17
Ilustração 14 - Código de estilo dos cartões	18
Ilustração 15 - Código para apresentar as várias receitas	18
Ilustração 16 - Função que vai buscar os dados para apresentar.....	19
Ilustração 17 - Esboço da página de apresentação de uma receita.....	19
Ilustração 18 - Vista final da aplicação web (durante a fase de testes)	20
Ilustração 19 - Etcher 1	27
Ilustração 20 - Etcher 2.....	27
Ilustração 21 - Ambiente de trabalho do Raspbian.....	27
Ilustração 22 - Configuração do phpMyAdmin	28
Ilustração 23 - Configuração do vsftpd	29
Ilustração 24 - Interface do Android Studio	30
Ilustração 25 - Emulador do Android Studio.....	31



Introdução

Neste relatório vou apresentar um projeto que consiste numa aplicação web onde os utilizadores inserem receitas culinárias, um servidor para alojar a mesma, e uma aplicação para dispositivo móvel com a finalidade de visualizar detalhadamente as receitas inseridas pelo utilizador.

Como a minha mãe gosta de cozinhar, eu era constantemente massacrado para imprimir receitas, por isso decidi, para o meu projeto, fazer algo que evitasse a necessidade de imprimir.



Ilustração 1 - Logótipo do projeto



Objetivos

“CookIt” tem como finalidade facilitar a visualização de receitas culinárias ao evitar confusão com folhas e pastas arquivadoras, sendo uma alternativa à impressão.

O projeto baseia-se em três partes:

- Uma aplicação web que permite ao utilizador:
 - Inserir e editar as suas receitas culinárias
 - Aceder às receitas publicadas por outros utilizadores
 - Descarregar as receitas
- Um servidor web que tem as funções de:
 - Alojamento da aplicação web
 - Base de dados
- Uma aplicação para dispositivo móvel com o objetivo de:
 - Visualizar as receitas de forma mais simples e ordenada



Aplicação Web

Uma aplicação web é um sistema de informática projetado para utilização através de um navegador, pode ser executado a partir de um servidor HTTP ou localmente, no dispositivo do utilizador.

Software Utilizado

WampServer



WampServer é um ambiente de desenvolvimento web para Windows, permite criar aplicações web com Apache2, PHP e base de dados MySQL.

Antes de carregar a aplicação para o servidor, fui utilizando o WampServer para testar a aplicação web no meu computador.

Sublime Text



Sublime Text é um software multiplataforma de edição de texto.

Escolhi este editor de texto porque já tinha usado o Sublime Text anteriormente, tem uma excelente performance e disponibiliza algumas ferramentas muito úteis como a seleção múltipla.

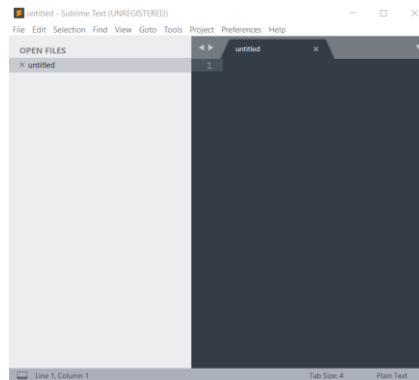


Ilustração 2 - Interface do Sublime Text

Tecnologias Web Utilizadas

HTML



HTML (“*Hyper Text Markup Language*”), é o formato padrão para criação de páginas online e aplicações web, os blocos de construção são chamados elementos, elementos são representados por tags, os Browsers não exibem as tags HTML mas usam-nas para renderizar o conteúdo da página.

CSS



CSS (“*Cascading Style Sheets*”), é um simples mecanismo para adicionar estilo a um documento web, descreve como elementos HTML são exibidos na tela.



JavaScript



JavaScript é uma linguagem de programação interpretada, foi originalmente implementada como parte dos navegadores web para que *scripts* pudessem ser executados do lado do cliente e interagissem com o utilizador sem a necessidade deste *script* passar pelo servidor.

Bootstrap



Bootstrap é um *framework* HTML, CSS e JavaScript de código aberto para desenvolvimento de páginas web responsivas e adaptáveis para dispositivos com tela mais pequena.

jQuery



jQuery é uma biblioteca JavaScript rápida e pequena, “*write less, do more*”, que torna o uso de JavaScript muito mais fácil.



AJAX



AJAX (“*Asynchronous JavaScript And XML*”), não é uma linguagem de programação, usa uma combinação de um objeto *XMLHttpRequest* do navegador (para pedir dados de um servidor web), JavaScript e HTML DOM (para exibir ou manipular os dados). AJAX permite que as páginas web sejam atualizadas sem ter de carregar toda a página, sendo possível atualizar apenas algumas partes.

Como funciona:

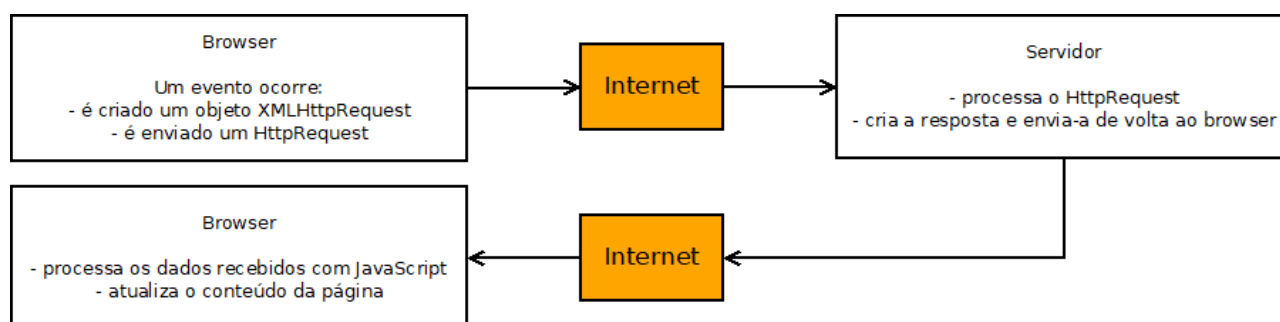


Ilustração 3 - Esquema do funcionamento de AJAX

PHP



PHP (“*PHP: HyperText Preprocessor*”, originalmente “*Personal Home Page*”), é uma linguagem de programação que corre do lado do servidor e uma ferramenta poderosa para criar páginas web dinâmicas.

mPDF

mPDF é uma biblioteca PHP que gera ficheiros PDF a partir de HTML.



PHPMailer



PHPMailer é uma biblioteca PHP para enviar e-mails facilmente e com segurança a partir de um servidor web.

Summernote



Summernote é um editor WYSIWYG (“*What You See Is What You Get*”), que permite ter na página web uma área de texto onde o utilizador pode formatar e decorar o texto colocado.

AngularJS



AngularJS é uma *framework* JavaScript de código aberto que estende o HTML com novos atributos, é perfeito para SPAs (“*Single Page Applications*”).

Foi a primeira vez que trabalhei com AngularJS, no início foi bastante difícil perceber esta *framework* pois achei um pouco complexa, mas valeu a pena porque torna mais simples a interatividade da página web com o utilizador.

SQL

SQL (“*Structured Query Language*”), é uma linguagem que permite o acesso e manipulação de bases de dados.



Desenvolvimento

Estrutura

A aplicação web é acedida a partir de um único ficheiro, para navegar por entre as páginas usa-se uma variável GET que indica qual página deverá mostrar, assim evita-se repetição de código. Tem uma interface simples, com uma barra de navegação e um rodapé que são exibidos em todas as páginas.

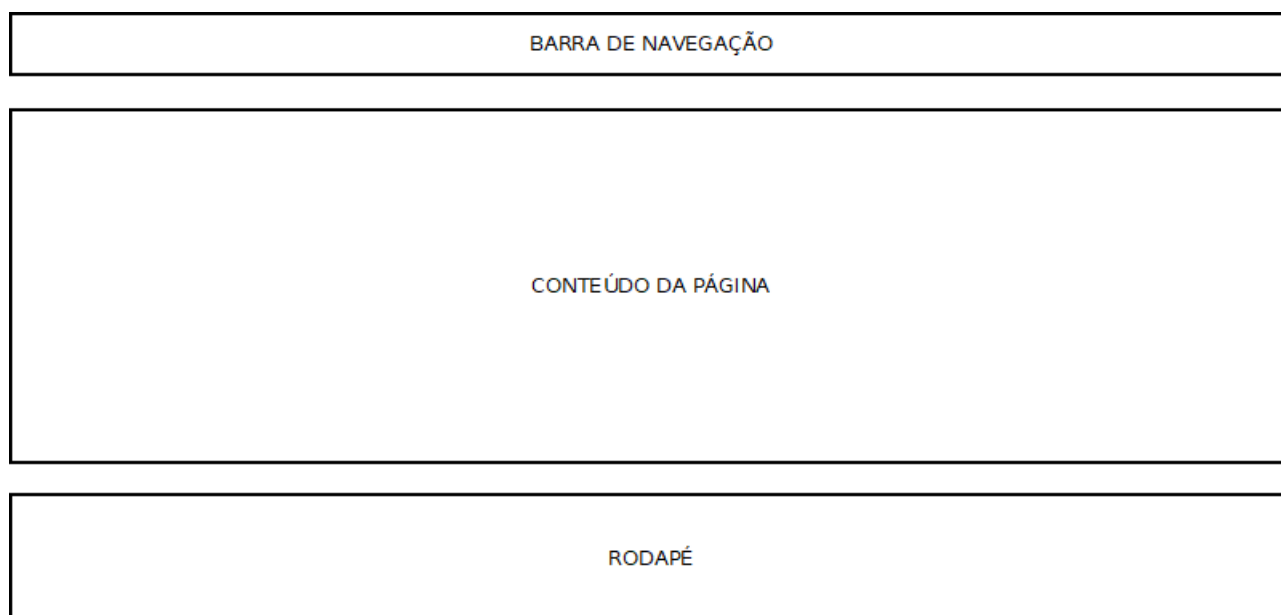


Ilustração 4 - Esboço da estrutura da página



Iniciar Sessão / Criar Conta

Ambos os formulários para iniciar sessão e criar conta são apresentados num *modal* (janela de diálogo).

Na janela de iniciar sessão existem três campos: e-mail, palavra-passe, e uma *checkbox* para permitir ao Browser guardar uma *cookie* com o e-mail, para que preencha o e-mail automaticamente da próxima vez que o utilizador pretender iniciar sessão.

Já na janela para criar conta são quatro campos: nome, e-mail, palavra-passe e confirmação da palavra-passe. Para fazer a validação deste formulário usei AngularJS o que permitiu fazer a validação sem que o utilizador tivesse de clicar em algum botão.

```
<div class="modal-body">  
<form role="form" id="form-signup" name="formSignup" novalidate>  
<div class="form-group">
```

Ilustração

5 - Tag do formulário de criar conta

```
<div class="form-group">  
  <label for="signup_name">  
    Nome  
  </label>  
  <input type="text" class="form-control" id="signup-name" name="name" ng-model="nameMdl" autocomplete="name" required>  
  <p class="help-block">  
    <span class="invalid-field" ng-show="formSignup.name.$error.required">Nome obrigatório.</span>  
    <span class="valid-field" ng-show="formSignup.name.$valid">Nome válido.</span>  
  </p>  
</div>
```

Ilustração 6 - Bloco do nome do formulário criar conta

O *input* para o nome tem a propriedade *required* para que o Browser saiba que é um campo obrigatório, mais a baixo temos dois bocados de texto que só são exibidos se a condição dentro do atributo *ng-show* for verdadeira. A primeira verifica se o campo *name* está vazio e apresenta “Nome obrigatório” caso esteja. A segunda verifica se o campo *name* está válido.

O botão para submeter o formulário tem um atributo *ng-disabled* que verifica se o nome e o email são inválidos, se a palavra-passe é inválida e se não combina com a confirmação de palavra-passe, se isto for verdadeiro o botão fica desativado.

Estes atributos (*ng-show* e *ng-disabled*) são AngularJS.



Os dados do formulário são enviados para o servidor com AJAX, aí a palavra-passe é codificada em SHA1 e são introduzidos os dados na base de dados. Após isso é enviado, com PHPMailer, um e-mail de confirmação de conta para o utilizador.

Formulário de Nova Receita

BARRA DE NAVEGAÇÃO DO FORMULÁRIO															
<table border="1" style="width: 100%; border-collapse: collapse;"><tr><td style="text-align: center; padding: 5px;">TÍTULO</td></tr><tr><td style="text-align: center; padding: 5px;">IMAGEM</td></tr><tr><td style="text-align: center; padding: 5px;">PREVISÃO DA IMAGEM</td></tr><tr><td style="text-align: center; padding: 5px;"><table border="1" style="width: 100%; border-collapse: collapse;"><tr><td style="text-align: center; padding: 5px;">DIFICULDADE</td><td style="text-align: center; padding: 5px;">TEMPO</td></tr><tr><td style="text-align: center; padding: 5px;">CATEGORIA</td><td style="text-align: center; padding: 5px;">ROBÔ</td></tr><tr><td colspan="2" style="text-align: center; padding: 5px;">INGREDIENTES</td></tr><tr><td colspan="2" style="text-align: center; padding: 5px;">PROCEDIMENTO</td></tr><tr><td colspan="2" style="text-align: center; padding: 5px;">PÚBLICO</td></tr></table></td></tr></table>		TÍTULO	IMAGEM	PREVISÃO DA IMAGEM	<table border="1" style="width: 100%; border-collapse: collapse;"><tr><td style="text-align: center; padding: 5px;">DIFICULDADE</td><td style="text-align: center; padding: 5px;">TEMPO</td></tr><tr><td style="text-align: center; padding: 5px;">CATEGORIA</td><td style="text-align: center; padding: 5px;">ROBÔ</td></tr><tr><td colspan="2" style="text-align: center; padding: 5px;">INGREDIENTES</td></tr><tr><td colspan="2" style="text-align: center; padding: 5px;">PROCEDIMENTO</td></tr><tr><td colspan="2" style="text-align: center; padding: 5px;">PÚBLICO</td></tr></table>	DIFICULDADE	TEMPO	CATEGORIA	ROBÔ	INGREDIENTES		PROCEDIMENTO		PÚBLICO	
TÍTULO															
IMAGEM															
PREVISÃO DA IMAGEM															
<table border="1" style="width: 100%; border-collapse: collapse;"><tr><td style="text-align: center; padding: 5px;">DIFICULDADE</td><td style="text-align: center; padding: 5px;">TEMPO</td></tr><tr><td style="text-align: center; padding: 5px;">CATEGORIA</td><td style="text-align: center; padding: 5px;">ROBÔ</td></tr><tr><td colspan="2" style="text-align: center; padding: 5px;">INGREDIENTES</td></tr><tr><td colspan="2" style="text-align: center; padding: 5px;">PROCEDIMENTO</td></tr><tr><td colspan="2" style="text-align: center; padding: 5px;">PÚBLICO</td></tr></table>	DIFICULDADE	TEMPO	CATEGORIA	ROBÔ	INGREDIENTES		PROCEDIMENTO		PÚBLICO						
DIFICULDADE	TEMPO														
CATEGORIA	ROBÔ														
INGREDIENTES															
PROCEDIMENTO															
PÚBLICO															

Ilustração 7 - Esboço do formulário de nova receita

O formulário está separado em quatro partes por isso criei uma barra de navegação distinta que leva o utilizador a cada parte com *smooth scroll*. Além disso a barra está “presa” no topo da página.



```
277 #form-create-nav {
278     position: -webkit-sticky;
279     position: sticky;
280     z-index: 1000;
281 }
```

Ilustração 8 - Código

de estilo da barra de navegação

Para criar a previsão da imagem carregada precisei de adicionar um atributo à tag do formulário, *runat="server"*, e um *onchange* à tag do *input*.

```
89 <div class="form-group">
90 <label for="form-create-image">
91     Imagem
92 </label>
93 <input type="file" accept="image/*" class="form-control" id="form-create-image" name="image" onchange="readURL(this);">
94 
95 </div>
```

Ilustração 9 - Bloco da imagem do formulário nova receita

Quando o utilizador altera a imagem é executada a função *readURL*:

```
15 function readURL (input) {
16
17     if (input.files && input.files[0]) {
18
19         var reader = new FileReader();
20
21         reader.onload = function(e) {
22
23             $("#image-preview").attr("src", e.target.result);
24         }
25
26         reader.readAsDataURL(input.files[0]);
27     }
28 }
```

Ilustração 10 - Função readURL

Para os ingredientes coloquei uma área de texto com Summernote:



```
155 <div id="form-create-ingredients">
156   <div class="subtitle">
157     <span>Ingredientes</span>
158   </div>
159
160   <div class="form-group">
161     <textarea class="summernote" name="ingredients"><?php if($edit==true) echo $ingredients; ?></textarea>
162   </div>
163 </div>
```

Ilustração 11 - Bloco para ingredientes do formulário nova receita

```
99  $(''.summernote').summernote({
100
101    toolbar: [
102      ['style', ['bold', 'italic', 'underline', 'clear']],
103      ['font', ['strikethrough', 'superscript', 'subscript']],
104      ['fontsize', ['fontsize']],
105      ['color', ['color']],
106      ['para', ['ul', 'ol', 'paragraph']],
107      ['height', ['height']]
108    ],
109    height: 200,
110    disableResizeEditor: true,
111    disableDragAndDrop: true
112  });
```

Ilustração 12 - Inicialização do Summernote

No procedimento coloquei um botão para que o utilizador adicione vários campos de texto fazendo assim um sistema de passos (passo 1, passo 2, ...).

Criei uma opção para que os utilizadores escolhessem se outros utilizadores poderiam ver a sua receita.

Formulário de Editar Receita

Para editar uma receita a aplicação utiliza o mesmo formulário de nova receita, apenas preenche os campos automaticamente com o que o utilizador colocou quando a criou.



Página de Apresentação de Receitas

BARRA DE PESQUISA			
CARTÃO	CARTÃO	CARTÃO	CARTÃO
CARTÃO	CARTÃO	CARTÃO	CARTÃO
BARRA DE PAGINAÇÃO			

Ilustração 13 - Esboço da página de apresentação de receitas

As receitas são apresentadas numa espécie de cartão, os cartões foram feitos de maneira que as imagens ficassem sem margens brancas e continuassem proporcionais ao seu tamanho original.



```
604 .recipe {
605     padding: 0px;
606     margin: 10px;
607     background-color: #FFFFFF;
608     box-shadow: 0px 0px 5px var(--dark-color);
609     z-index: 100;
610 }
611
612 .recipe:hover .img-container {
613     -webkit-transform: scale(1.1);
614     -moz-transform: scale(1.1);
615     transform: scale(1.1);
616     -webkit-filter: grayscale(0%);
617     filter: grayscale(0%);
618 }
619
620
621 .img-container {
622     -webkit-transition: 300ms;
623     -moz-transition: 300ms;
624     -o-transition: 300ms;
625     -webkit-filter: grayscale(50%);
626     filter: grayscale(50%);
627     height: 250px;
628     background-repeat: no-repeat;
629     background-size: cover;
630     background-position: center;
631 }
```

Ilustração 14 - Código de estilo dos cartões

Tanto o sistema de pesquisa como o de paginação funcionam com AngularJS, assim que algo é alterado na barra de pesquisa ou na barra de paginação é executada uma função que faz um pedido AJAX ao servidor e retorna os dados em JSON.

```
<div class="col-md-3" ng-repeat="recipe in recipes">
  <a href="?p=recipe_details&id={{recipe.id}}">
    <div class="recipe">
      <div style="overflow: hidden;"><div class="img-container" style="background-image: url('{{recipe.imgPath}}');"></div></div>
      <div class="recipe-caption">
        <p class="title" align="center" data-toggle="tooltip" title="{{recipe.title}}">{{recipe.title}}</p>
        <div class="recipe-prop">
          <div style="text-overflow: ellipsis; overflow: hidden; white-space: nowrap;">
            <span class="glyphicon glyphicon-cutlery"></span>
            {{recipe.category}}
          </div>
          <div style="text-overflow: ellipsis; overflow: hidden; white-space: nowrap;">
            <span class="glyphicon glyphicon-time"></span>
            {{recipe.timeDrt}}
          </div>
          <div style="text-overflow: ellipsis; overflow: hidden; white-space: nowrap;">
            <span class="glyphicon glyphicon-fire" ng-repeat="i in nFireIcon(recipe.difficulty) track by $index"></span>
            {{recipe.difficulty}}
          </div>
          <div style="text-overflow: ellipsis; overflow: hidden; white-space: nowrap;">
            <span class="glyphicon glyphicon-flash"></span>
            {{recipe.robot}}
          </div>
        </div>
      </div>
    </div>
  </a>
</div>
```

Ilustração 15 - Código para apresentar as várias receitas



```
$scope.updateRecipes = function () {  
  
    $http.get(  
        "ajax/listRecipes.php?in=" + $scope.searchInMdl + "&cat=" + $scope.searchBarCategoriesMdl + "&title=" + $scope.searchBarTitleMdl + "&robot=" + $scope.  
        .searchBarRobotMdl + "&order=" + $scope.searchBarOrderByMdl + "&current_page=" + $scope.pagination.currentPage  
    ).then(function (response) {  
        $scope.recipes = response.data.recipes;  
        $scope.pagination = response.data.pagination;  
        $scope.setPaginationButtons();  
        $('[data-toggle="tooltip"]').tooltip({trigger: "hover"});  
    });  
}
```

Ilustração 16 - Função que vai buscar os dados para apresentar

Página de Apresentação de uma Receita

TÍTULO		
	IMAGEM	
DETALHES E AÇÕES		
	INGREDIENTES	
	PROCEDIMENTO	

Ilustração 17 - Esboço da página de apresentação de uma receita

Nesta página qualquer um pode descarregar a receita ou criar um PDF (podendo depois imprimir a partir daí) sem que tenham sessão iniciada. A aplicação verifica se o utilizador tem a sessão aberta, se tiver, é criado um botão para adicionar aos favoritos, se o utilizador tiver a sessão aberta e for o autor da receita é criado um botão para editar.

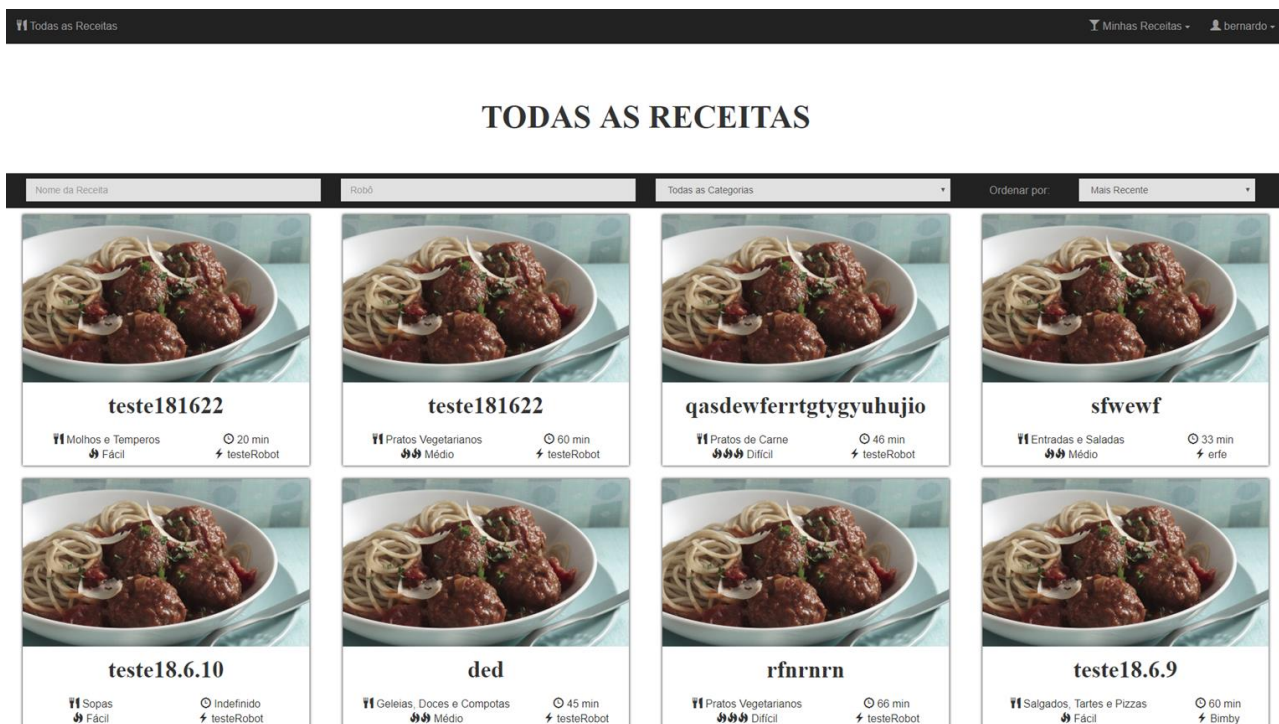


Ilustração 18 - Vista final da aplicação web (durante a fase de testes)

Tabelas da Base de Dados

Tabela de utilizadores:

users
name
email
password
valid
validation_code
id

O campo *valid* define se a conta do utilizador foi confirmada e o campo *validation_code* guarda o código de confirmação do utilizador, uma vez confirmado este campo fica em branco.



Tabela de receitas:

recipes
title
difficulty
time_drt
category
robot
ingredients
img_path
public
user_id
lst_chg_date
valid
id

O campo *title* guarda o título da receita, *difficulty* a dificuldade, *time_drt* a duração estimada, *category* a categoria, *robot* o robô, *ingredients* os ingredientes, *img_path* o nome da imagem, *public* define se a receita é pública ou não, *user_id* indica o id do autor, *lst_chg_date* a data da última alteração, *valid* indica se a receita é válida (possui título, ingredientes e pelo menos um passo no procedimento).

Tabela dos passos:

steps
step_n
value
recipe_id

O campo *step_n* define a ordem do passo (1º, 2º, 3º, ...), o *value* contém o texto e o *recipe_id* é o que faz referência à receita.



Tabela dos favoritos:

favorites
<i>user_id</i>
<i>recipe_id</i>

O campo *user_id* faz referência ao utilizador e o *recipe_id* à receita.



Servidor Web

Um servidor web é um software responsável por aceitar pedidos HTTP de clientes e servi-los com respostas.

Materiais Utilizados

Raspberry Pi 3



Processador	Cortex A53
Velocidade de processador	1,2 GHz
Nº de núcleos	Quad Core
RAM	1 GB
Conectividade	Wi-Fi e Bluetooth
Ligações	RJ-45, 4xUSB2.0, HDMI, Micro USB, Micro SD, Jack 3.5'
Preço	49,99€



Cartão Micro SD



Espaço	64 GB
Preço	???

Transformador



Tensão	5 V
Corrente	2500 mA
Preço	4,22€

Item	Preço
Raspberry Pi	49,99€
Transformador	4,22€
Preço Total	54,21€



Software Utilizado

Etcher



Etcher é um software para gravar imagens de sistemas operativos em cartões SD e pens USB.

PuTTY



PuTTY é um software de emulação de terminal que suporta SSH.

FileZilla



FileZilla é um cliente FTP, SFTP e FTPS.

Raspbian



Raspbian é um sistema operativo variante do Debian otimizado para o hardware do Raspberry Pi.



Apache2



Apache é um servidor do tipo HTTPD, compatível com o protocolo HTTP versão 1.1. As suas funcionalidades são mantidas através de uma estrutura de módulos, permitindo que o utilizador escreva seus próprios módulos.

PHP



PHP (*"PHP: HyperText Preprocessor"*, originalmente *"Personal Home Page"*), é uma linguagem de programação que corre do lado do servidor e uma ferramenta poderosa para criar páginas web dinâmicas.

MySQL



MySQL é um sistema de manipulação de base de dados que utiliza a linguagem SQL como interface.

phpMyAdmin



phpMyAdmin é uma aplicação web para administração do MySQL pela internet.

SSH

SSH (*"Secure Shell"*), é um protocolo de rede criptográfico para operação de serviços de rede de forma segura.



vsftpd

vsftpd (“*very secure FTP daemon*”), é um servidor FTP para sistemas Unix.

Configuração do Servidor

Para gravar a imagem do sistema operativo no micro SD utilizei o Etcher, depois de concluído, como o Raspbian já vem pré-instalado, é só colocar o micro SD no Raspberry.

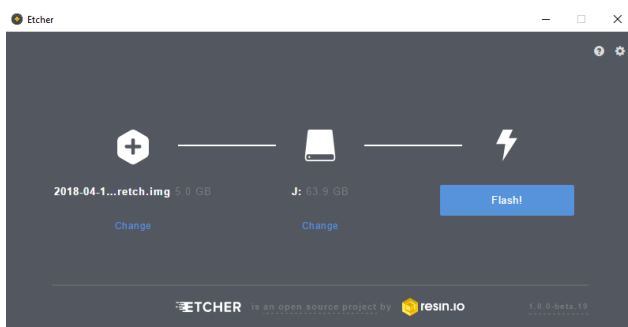


Ilustração 19 - Etcher 1

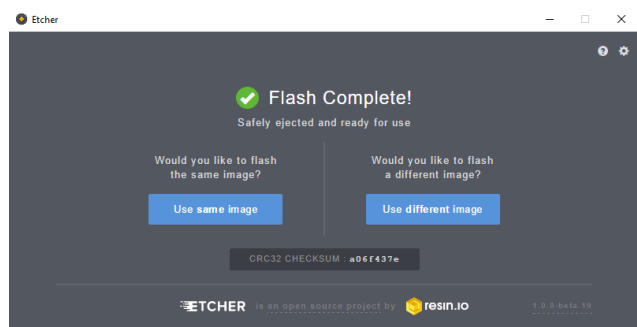


Ilustração 20 - Etcher 2

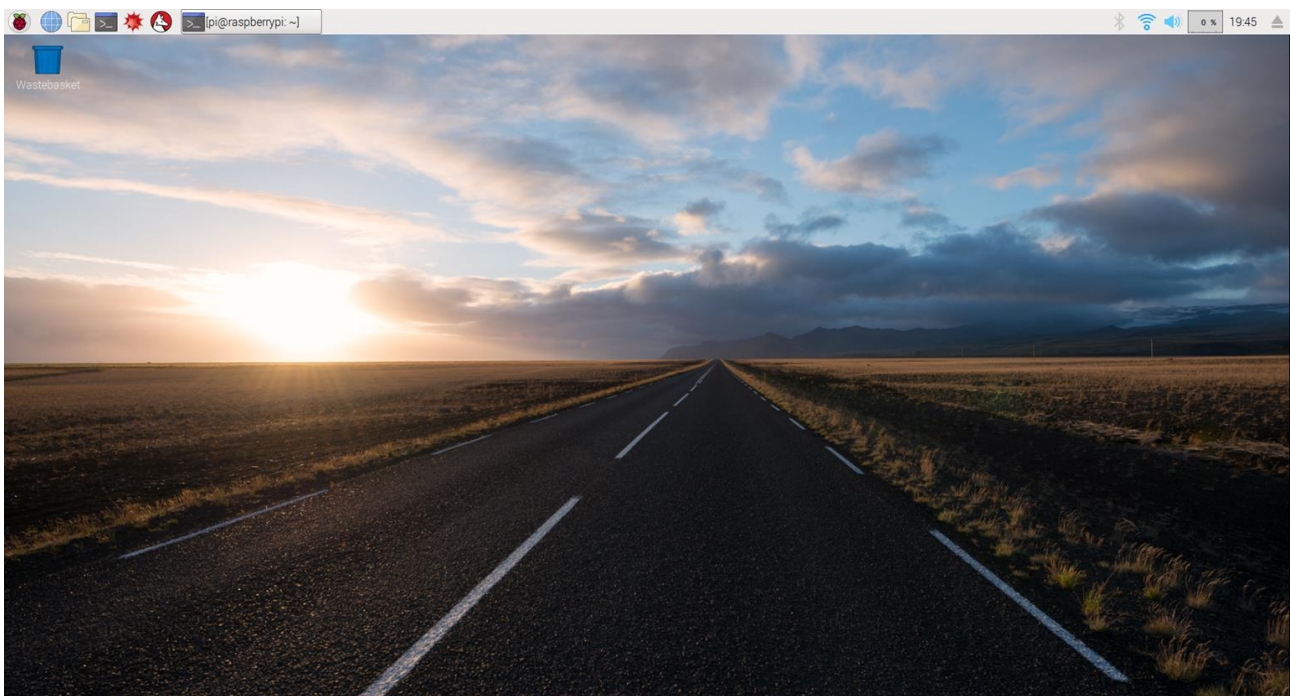


Ilustração 21 - Ambiente de trabalho do Raspbian



Os primeiros comandos que executei foram para atualizar o sistema operativo:

```
>sudo apt-get update
>sudo apt-get upgrade
```

Após isso pus em execução o serviço SSH:

```
>sudo service ssh enable
>sudo service ssh start
```

Instalei o Apache2:

```
>sudo apt-get install apache2
```

O PHP, o MySQL e o phpMyAdmin:

```
>sudo apt-get install php libapache2-mod-php
>sudo apt-get install mysql-server php-mysql
>sudo apt-get install phpmyadmin
```

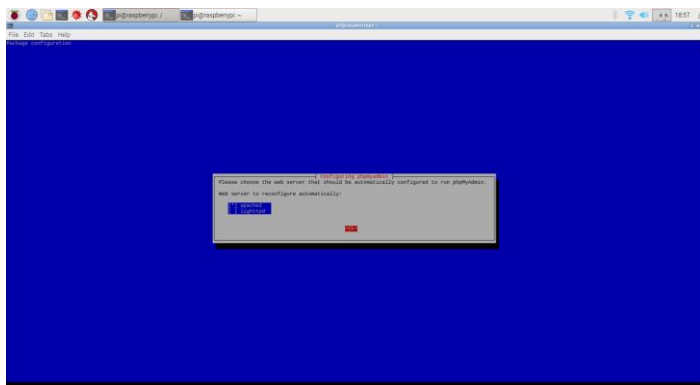


Ilustração 22 - Configuração do

phpMyAdmin

Criei um utilizador na base de dados:

```
>sudo mysql -u root
>CREATE USER 'cookit'@'localhost' IDENTIFIED BY 'password';
>CREATE DATABASE cookit;
>GRANT ALL PRIVILEGES ON cookit.* TO 'cookit'@'localhost';
>FLUSH PRIVILEGES;
```

E instalei o vsftpd:

```
>sudo apt-get install vsftpd
```

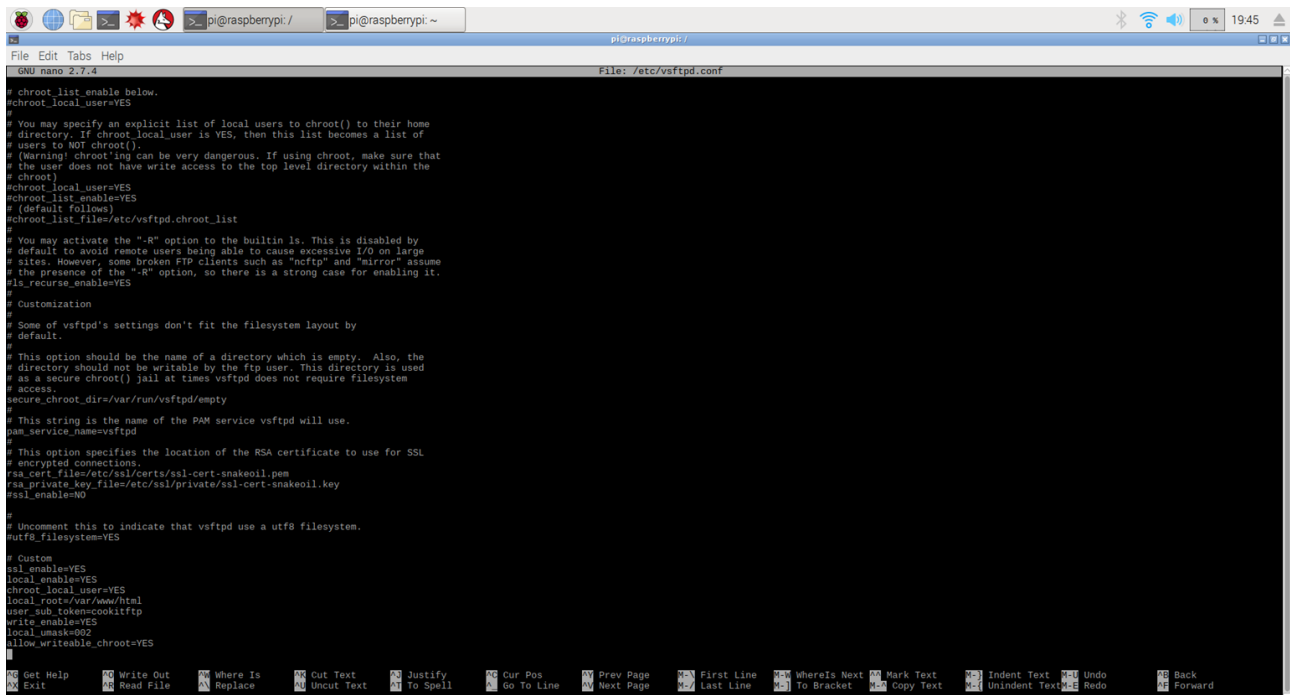
Por fim configurei o vsftpd:

```
>sudo nano /etc/vsftpd.conf
```

Adicionei no final do ficheiro:



```
ssl_enable=YES
local_enable=YES
chroot_local_user=YES
local_root=/var/www/html
user_sub_token=cookitftp
write_enable=YES
local_umask=002
allow_writeable_chroot=YES
```



```
File Edit Tabs Help
GNU nano 2.7.4 File: /etc/vsftpd.conf
# chroot_list_enable below.
#chroot_list_enable=YES
# You may specify an explicit list of local users to chroot() to their home
# directory. If chroot_local_user is YES, then this list becomes a list of
# users to NOT chroot().
# (Warning! chrooting can be very dangerous. If using chroot, make sure that
# the user does not have write access to the top level directory within the
# chroot)
#chroot_list_enable=NO
#chroot_list_enable=YES
# (default follows)
#chroot_list_file=/etc/vsftpd.chroot_list
# You may activate the "-R" option to the builtin ls. This is disabled by
# default to avoid remote users being able to cause excessive I/O on large
# sites. However, some broken FTP clients such as "ncftp" and "mirror" assume
# the presence of the "-R" option, so there is a strong case for enabling it.
#ls_recurse_enable=YES
#
# Customization
#
# Some of vsftpd's settings don't fit the filesystem layout by
# default.
#
# This option should be the name of a directory which is empty. Also, the
# directory should not be writable by the ftp user. This directory is used
# as a secure chroot() jail at times vsftpd does not require filesystem
# access.
#secure_chroot_dir=/var/run/vsftpd/empty
#
# This string is the name of the PAM service vsftpd will use.
#pam_service_name=vsftpd
#
# This option specifies the location of the RSA certificate to use for SSL
# encrypted connections.
#rsa_cert_file=/etc/ssl/certs/ssl-cert-snakeoil.pem
#rsa_private_key_file=/etc/ssl/private/ssl-cert-snakeoil.key
#ssl_enable=NO
#
# Uncomment this to indicate that vsftpd use a utf8 filesystem.
#utf8_filesystem=YES
#
# Custom
ssl_enable=YES
local_enable=YES
chroot_local_user=YES
local_root=/var/www/html
user_sub_token=cookitftp
write_enable=YES
local_umask=002
allow_writeable_chroot=YES
#
# Get Help
# Exit
# Write Out
# Read File
# Where Is
# Replace
# Cut Text
# Uncut Text
# Justify
# To Spell
# Cur Pos
# Go To Line
# Prev Page
# Next Page
# First Line
# Last Line
# WhereIs Next
# To Bracket
# Mark Text
# Copy Text
# Indent Text
# Undo
# Unindent Text
# Redo
# Back
# Forward
```

Ilustração 23 - Configuração do vsftpd

E reiniciei o serviço vsftpd:

```
>sudo service vsftpd restart
```

Após ter configurado o servidor utilizei o FileZilla para carregar toda a aplicação web para o servidor, usei o PuTTY para aceder ao terminal do Raspberry adicionei os devidos privilégios aos ficheiros lá colocados:

```
>sudo chown -R www-data:www-data /var/www/html/*
>sudo chmod -R 755 /var/www/html/*
```



Aplicação Android

Software Utilizados

Android Studio



Android Studio é um ambiente de desenvolvimento integrado para desenvolver para a plataforma Android.

Foi a primeira vez que utilizei este software, mas não tive muita dificuldade a adaptar-me ele.

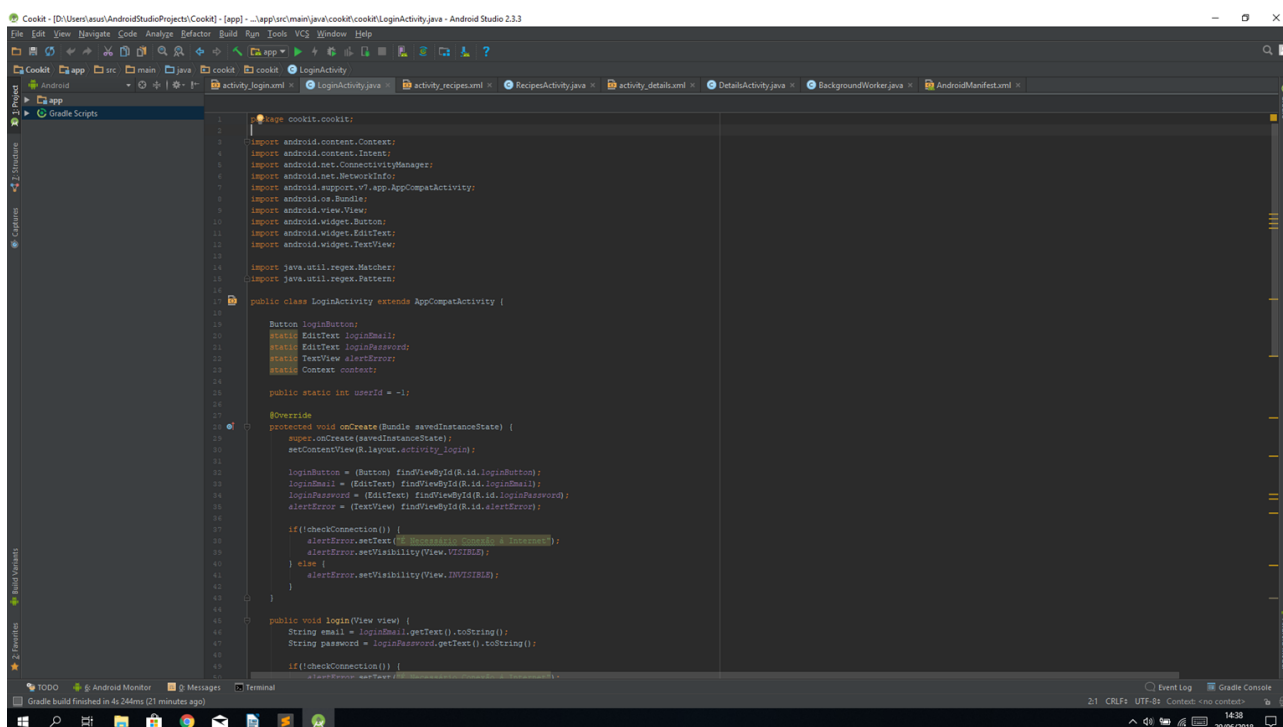


Ilustração 24 - Interface do Android Studio



Desenvolvimento

Esta aplicação foi concebida para apenas visualizar as receitas, no Android Studio as “páginas” são chamadas de atividades, existem apenas três atividades:

- A atividade para iniciar sessão, a aplicação vai verificar os dados que o utilizador colocou ao servidor com pedidos HTTP, todos os pedidos deste tipo são feitos através de uma class que criei chamada BackgroundWorker.
- Depois de iniciar sessão é exibida a segunda atividade, que mostra todas as receitas que o utilizador adicionou aos favoritos.
- O utilizador escolhe aquela que pretende visualizar e aparece a terceira atividade, que é dividida em secções: a capa onde tem o título e os detalhes da receita, os ingredientes, e o procedimento onde cada passo é uma secção; para o utilizador navegar por entre estas secções utiliza dois botões.



Ilustração 25 - Emulador do Android Studio



Recursos Online

Durante o desenvolvimento do projeto serviram-me de apoio os seguintes websites e fóruns:

- <https://developer.android.com/>
- <https://www.learn2crack.com/>
- <https://stackoverflow.com/>
- <https://askubuntu.com/>
- <https://www.codecademy.com/learn>
- <https://www.w3schools.com/>
- <https://www.raspberrypi.org/>
- <https://angular.io/>
- <https://jquery.com/>
- <http://php.net/>
- <https://getbootstrap.com/>
- <https://github.com/mpdf/mpdf>
- <https://github.com/PHPMailer/PHPMailer>
- <https://summernote.org/>



Conclusão

Gostei de fazer este projeto pois fiquei a conhecer e a saber usar alguns softwares com que nunca tinha trabalhado antes, tais como o Android Studio e o AngularJS.

Ao longo do seu desenvolvimento encontrei diversas dificuldades, que por vezes me faziam perder a cabeça, mas consegui superar a maioria delas, aquelas que não consegui, encontrei outra alternativa.