Relatório do Projeto Final – *NoteFlex*



Autor: Catarina Nabais, 21478, Turma A

Docente: Professor João Monge

Março 2021

ìndice

[Introdução 2](#_Toc66720283)

[1. Responsividade 2](#_Toc66720284)

[2. Código universal 2](#_Toc66720285)

[a. HTML 2](#_Toc66720286)

[b. CSS 2](#_Toc66720287)

[c. JQuery/Javascript 3](#_Toc66720288)

[3. Menu de navegação 5](#_Toc66720289)

[*a.* Versão *Mobile* 5](#_Toc66720290)

[*b.* Versão *Desktop* 6](#_Toc66720291)

[4. Página “*Index*” 6](#_Toc66720292)

[a. Versão *Mobile* 7](#_Toc66720293)

[*b.* Versão *Desktop* 8](#_Toc66720294)

[5. Página “*Submit & Request*” 9](#_Toc66720295)

[a. Versão *Mobile* 10](#_Toc66720296)

[*b.* Versão *Desktop* 10](#_Toc66720297)

[6. Página “*Faq & Contact*” 11](#_Toc66720298)

[5. Conclusões 11](#_Toc66720299)

# Introdução

Este trabalho consiste numa demonstração crítica da criação de um *website* responsivo, analisando o código que o compõe. Através de uma compreensão dos códigos HTML, CSS, e JavaScript explorados em aula, o grupo pôs em prática conceitos que demonstram a necessidade de um código responsivo e simples, e que, a partir de um conceito complexo, nos deu oportunidade de criar um website simples mas eficaz.

**Conceito**: NoteFlex é uma plataforma dedicada à partilha de apontamentos de estudo. No site, os alunos podem ler e requisitar apontamentos uns dos outros, assim como submeter os seus próprios. NoteFlex tem como principal objetivo o aumento do desempenho escolar através da acessibilidade, da comunicação, e da interajuda.

**Palavras-chave:** Responsivo. Responsividade. HTML. CSS. JQuery. Java Script. Código.

# Responsividade

Todo o código que compõe o site foi construído considerando um telemóvel ou tablet como dispositivo principal usado pelo utilizador. Utilizando uma *media query,* os componentes adaptam-se a tamanhos de ecrã maiores. Consoante o tamanho do dispositivo, o *layout* do *website* pode mudar, mas os seus componentes mantêm-se (sendo que em certas circunstâncias podem também mudar, como num caso específico na página *“Submit & Request*”, falado mais à frente).

# Código universal

## HTML

Para que o site atingisse a mencionada adaptabilidade, tendo em conta tanto a linguagem em utilização como o tamanho do dispositivo, a primeira, e mais básica, medida tomada foi assegurar que o código HTML continha as seguintes *tags* *meta*:

**<meta *charset*="UTF-8">**

**<meta *name*="viewport" *content*="width=device-width, initial-scale=1.0">**

*Charset* assegura que o browser saberá ler caracteres especiais (como, por exemplo, o ‘ç’ da língua portuguesa); “width=device-width” controla o tamanho do *viewport* que aqui está assinalado como igual ao comprimento do dispositivo (adaptando-se, portanto, ao tamanho do ecrã em utilização); a propriedade “initial-scale=1.0” controla o zoom inicial da página.

## CSS

O *header* foi um dos primeiros elementos a ser definido, pois é onde se encontra a nossa barra de navegação e a imagem principal da página *Index*. Foi decidido que o *header* apenas teria apenas a barra de navegação. Logo, as suas propriedades são:

**header {**

**position:sticky;**

**…**

**display:flex;**

**justify-content:flex-end;**

**}**

A posição *sticky* assegura que o menu de navegação no header acompanha o ecrã no *scroll*. *Justify-content:flex-end* fixa os elementos (links) no final da flexbox que é o menu. *Overflow:hidden*

Para código geral, retirou-se a “decoração” dos links em ambas as barras de navegação (independentemente do ecrã), com as propriedades *text-decoration* e *list-style*, pois os *links* foram criados em lista de modo a facilitar a acessibilidade e compreensão. *Text-decoration:none* retira o sublinhado *default* dos links, enquanto *list-style:none* retira o estilo *bulleted* da lista.

**.navlinks li,.navlinks a {**

**text-decoration:none;**

**list-style:none;**

**}**

O *footer* recebe as propriedades seguintes, para que fique estático no final da página.

**footer {**

**position: static;**

**bottom: 0;**

**}**

Em *mobile,* dispostos em bloco para melhorar estética*:*

**.links a{**

**padding:5px;**

**display:block;**

**}**

Em *desktop,* dispostos em flexbox e em “2 partes” devido á propriedade *justify-content:space around*, que dispõe os 2 divs diferentes em que se encontra cada grupo de links, em limites opostos do div *parent*:

**.links {**

**display: flex;**

**justify-content: space-around;**

**}**

## JQuery/Javascript

O código JavaScript do site tem como objetivo duas funcionalidades interativas. A primeira destina-se a tornar visível/invisível o menu de navegação exclusivo para *mobile* (figs. 2 e 3).

**$(".menu\_icon").on("click", function () {**

**var estado = $(".navlinks\_tele").css("display");**

**if (estado == "none") {**

**$(".navlinks\_tele").css("display", "block")**

**} else {**

**$(".navlinks\_tele").css("display", "none")**

**}**

**});**

O estado inicial do menu é invisível (como observaremos no código CSS mais à frente, no ponto 3.1.), pelo que foi estabelecido no código JS que se este se encontrar invisível (none), o clique no botão “menu\_icon” (icon de barras visível nas figs. 2 e 3) o torna visível, e vice-versa.

A “segunda parte do código” despoleta eventos que atuam sobre a mencionada lista de *Frequently Asked Questions*, o acordião presente na página FAQ (fig. 1) e explicado mais à frente no ponto 6.

**$("li.question").on("click", function () {**

**$(this).next().slideToggle("500")**

**.siblings("li.answer").slideUp()**

**$(this).find("i").toggleClass("fas fa-plus fas fa-minus")**

**$("li.question").not(this).find("i").removeClass("fas**

**fa-minus").addClass("fas fa-plus");**

**});**

Graphical user interface, text

Description automatically generatedPrimeiramente, para distinguir a ação, foram utilizados dois ícones, um de “+” (que demonstra uma questão “fechada”) e um de “-“ (que demonstra uma “aberta”). O código funciona da seguinte maneira:

* Primeiro define o evento. Quando o utilizador clica num elemento “li” que seja de classe “question”:
* É selecionado o próximo elemento (.answer) e a resposta correspondente é aberta através do método “slideToggle”. O método no código “.siblings(“li-answer”).slideUp()” serve para que, ao mesmo tempo que se torna visível uma destas respostas, todas as outras são escondidas através do método “slideUp”;
* Foram criados ícones “+”/”-“. Alterando a classe dos elementos “i”, estes aparecerão de forma alternada. Esta funcionalidade foi adicionada pois, caso contrário, o ícone só mudaria de um para outro se o utilizador clicasse na mesma questão (mesmo estando esta já fechada ao clicar noutra questão).

Foram explicitadas as ações nas perguntas “siblings/irmãs” para que quando se abre uma questão, apenas essa se tornará visível e todas as outras continuem escondidas. Ou seja, o utilizador só abre uma questão de cada vez, e quando abre uma as outras fecham “automaticamente” para minimizar esforço e melhorar experiência.

Figura 1, acordião na página "FAQ & Contact", versão mobile

# Menu de navegação

## Versão *Mobile*

Foi criado um menu exclusivo para mobile, que se esconde (*display:none*) a partir dos 768px, quando se trata de ecrãs maiores. Encontra-se fora do *header* pois aparecerá abaixo deste, como um novo menu.

**<div class="navlinks\_tele">**

**<ul>**

**<li><a href="index.html" alt="Home">Home</a></li>**

**<li><a href="subreq.html" alt="Submit and Request">Submit & Request</a></li>**

**<li><a href="faqcontact.html" alt="FAQ and Contact">FAQ & Contact</a></li>**

**<li>**

**<form>**

**<input type="text" id="search\_bar" name="search">**

Graphical user interface, application

Description automatically generated **</form>**

**</li>**

**</ul>**

**</div>**

**nav {**

**display:none;**

**}**

**.navlinks\_tele {**

**background-color:#eee8d8;**

**display: none;**

**}**

Figura 2, menu de navegação mobile escondido

Graphical user interface, application

Description automatically generated

Figura 3, menu de navegação mobile visível

Os links são dispostos em bloco, para criar o menu vertical, exemplificado na imagem:

**.links a{**

**padding:5px;**

**display:block;**

**}**

## Versão *Desktop*

Começamos por uma media query para que estes atributos só sejam visíveis em ecrãs que, como indica o CSS, tenham largura mínima de 768px.

**@media screen and (min-width: 768px){**

Os links estão dispostos com a propriedade *display:inline* para que a lista fique em linha na barra de navegação. *Align-items:center* assegura os links se encontram sempre centrados verticalmente e horizontalmente na barra (menu evidenciado na fig. 5).

**.navlinks{**

**display:inline;**

**}**

**.navlinks ul{**

**display: flex;**

**align-items: center;**

**}**

Evidenciamos também o icon de barras, já mencionado no ponto 2.3., que começa por estar invisível:

**.menu\_icon{**

**display:none;**

**}**

# Página “*Index*”

O *site* consiste em 3 páginas com componentes responsivos.

A **página principal (*homepage*)** é onde acontece o primeiro contacto do utilizador com a plataforma e, portanto, onde se encontram algumas informações introdutórias como.

Foram criados containers para agrupar elementos que fazem parte da mesma secção. São também estilizados como *flexbox* para permitirem o redimensionamento dos componentes de forma responsiva e sem causar *overflow*.

**<main>**

**<div class="container">**

**<div class="img\_wrapper">**

**<img src="assets/home\_img.jpg" id="home\_img" alt="Study Sesh">**

**</div>**

**…..**

* Imagem *header*, sob o qual está um botão interativo para utilizadores recorrentes que queiram imediatamente e facilmente direcionados à página de submissão e pedidos (submit/request). Com os seguintes atributos:

**.img\_wrapper img{**

**opacity:0.8;**

**width:100%;**

**height:378px;**

**object-fit:cover;**

**}**

Utilizaram-se as propriedades *width:100%* e *object-fit:cover* para que a imagem ocupe a largura do *div* “parente” por inteiro, mas que a altura “caiba” consoante as dimensões do div.

* Um pequeno parágrafo explicativo que introduz o objetivo do *site* “*What is Noteflex?*”;
* *Tops* divididos por informações mais relevantes. Estes tops privilegiam as necessidades dos utilizadores do *site*;
* Uma newsletter, onde o utilizador se pode registar para receber notícias e notificações;
* O *footer*, com links adicionais que disponibilizam mais informação pertinente.

## Versão *Mobile*

A disposição das colunas em *mobile* (ou seja, ecrãs menores de 768px, segundo o especificado acima) é *block*, para facilitar leitura e *layout*. Para isso, o *display* original é *block*, exemplificado na imagem ao lado.

**.home\_columns {**

**display: block;**

**margin: 50px;**

**}**

Text

Description automatically generated

Figura 4, colunas destaque da homepage, dispostas em bloco

## Versão *Desktop*

Graphical user interface, text

Description automatically generated

Figura 5, header principal, versão desktop

Já em *desktop*, o display muda para *flex*, e como já demonstrado no ponto 2.2., o flex-direction está disposto em linhas/*row*, logo as colunas passarão a ser visíveis em linha, como na imagem abaixo.

**.home\_columns {**

**display: flex;**

**justify-content: space-between;**

**column-gap: 20px;**

**margin: 50px;**

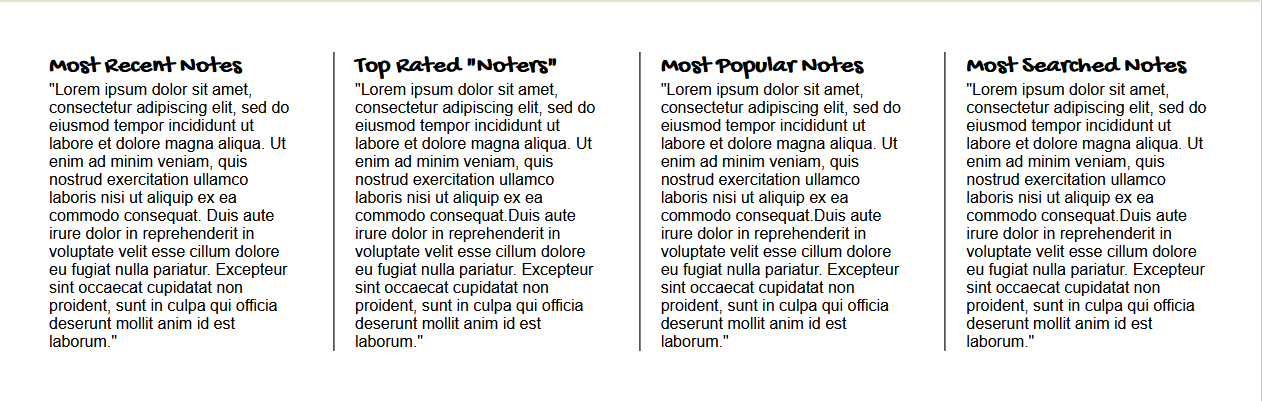
 **}**

Figura 6, colunas destaque da homepage, dispostas em linha

# Página “*Submit & Request*”

Na **página “Submit & Request”**, o utilizador pode ou submeter os seus apontamentos, ou requisitar apontamentos de outros utilizadores, consoante as suas necessidades.

Para tal, a página a página tem:

• Um formulário para submissão, onde o utilizador tem à disposição uma variedade de opções desde escolher a disciplina dos apontamentos que submete, até explicitar o seu nome, email (se assim o desejar), e submeter um documento;

• Outro formulário para pedidos, onde o utilizador pode requisitar apontamentos para determinada(s) disciplina(s).

A página contém ainda imagens (ou uma imagem, no caso mobile) responsivas, para melhorar a experiência visual dos utilizadores.

O *layout* da página inclui 2 *containers*, ocupando metade da página cada um. Um é para o *form* e imagens referentes à submissão, e o outro para os ligados à requisição. Criaram-se 2 linhas com 2 colunas cada, de forma a alinhar os componentes entre si dentro de cada *container* e permitir o seu redimensionamento adequado consoante o tamanho do ecrã.

**<div class="container" id="container\_request">**

**<div class="row">**

**<div class="column">**

**<form id="requestform">**

As imagens em si têm atributos flex para serem adaptáveis ao tipo de ecrã, como *align-self*, que alinha a imagem dentro do seu *container* ao meio, e *width: 80%*, que redimensiona a imagem a 80% da largura do seu *container*.

**#request\_img {**

**width: 80%;**

**display:flex;**

**align-self:center;**

**border-radius:10px;**

**}**

Quanto aos *forms*, para ser possível requisitar apontamentos de várias disciplinas em simultâneo, escolheu-se o *type ‘checkbox’*:

**<li><input type="checkbox" id="afc" name="afc">**

**<label for="afc">Arquitetura e Funcionamento de Computadores</label>**

**</li>**

**<li><input type="checkbox" id="mqi" name="mqi">**

**<label for="mqi">Métodos Quantitativos I</label>**

**</li>**

No *input*, escolheu-se o *type* 'radio’ com o objetivo que o utilizador submeta notas de uma disciplina de cada vez.

**<li>**

**<input type="radio" id="afcs" name="subject" value="afcs">**

**<label for="afcs">Arquitetura e Funcionamento de Computadores</label>**

**</li>**

Existem ainda várias escolhas como tipo de documento, nome, email, e um botão de submissão final.

## Versão *Mobile*

Graphical user interface, application

Description automatically generatedNa versão mobile, os containers onde se encontram os *forms* são dispostos em bloco, como exemplificado na imagem ao lado. Foram dados *IDs* específicos para diferenciar dos outros *containers* utilizados na página *Index*.

Figura 7, form de submissão, disposto em bloco

**#container\_submit, #container\_request {**

**margin: 70px;**

**display: block;**

**}**

Por uma questão de estética, foi decidido que a imagem que, no *desktop*, se encontra do lado direito na última coluna, seria escondida em *mobile* através do código:

**#request\_img {**

**display:none;**

**}**

## Versão *Desktop*

A disposição em colunas na versão *desktop* dá-se lado a lado, como na imagem:

Graphical user interface

Description automatically generated with medium confidence

Figura 8, form de requisição, disposto em linha

e através do código:

**.row {**

**display: flex;**

**flex-direction: row;**

**flex-wrap: wrap;**

**width: 100%;**

**}**

**.column {**

**display: flex;**

**flex-direction: column; }**

Criam-se assim 2 linhas e 2 colunas com o método *flexbox*. No elemento de classe *row,* a nossa linha, dispomos os elementos em direção da linha. A propriedade *flex-wrap* foi utilizado para que estes elementos fossem empilhados dentro do seu container, em vez de se disporem fora deste em linha. Já nas colunas, a propriedade f*lex-direction* define a direção dos conteúdos da *flexbox* em coluna, neste caso a imagem e o *form*.

# Página “*Faq & Contact*”

Finalmente, foi criada a **página “FAQ & Contact”**, que neste momento contém um *accordion* (acordião), com várias possíveis questões que o utilizador tenha, e que este pode ler antes de contactar a administração. O acordião é construído como uma lista, que depois é estilizada usando o

Cada resposta do nosso acordião começa com a propriedade *display:none* para que as respostas não estejam visíveis a não ser quando se clica nas perguntas.

**.faq li.answer {**

**background: #776564;**

**display: none;**

**color:#fff;**

**}**

O aparecimento/desaparecimento controla-se no código JavaScript, como explicado no ponto 2.3.

# Conclusões

A sumarização da totalidade do projeto de *web design* apresentado apresentou-se como um desafio. No entanto acredito que este relatório atingiu os objetivos estabelecidos e que representou devidamente o desenvolvimento das capacidades adquiridas ao longo da disciplina de Web e Multimédia I.

Este trabalho cimentou conhecimentos de HTML, CSS, JQuery/Java Script desenvolvidos ao longo do semestre, e representou a maneira como todos estes se interligam para criar um *website*. O ponto principal do relatório estabeleceu a importância e os mecanismos necessários para criar estruturas responsivas que se adaptem a qualquer dispositivo preferido pelo utilizador.