IMPLEMENTAÇÃO DO FRONTEND NO PROJECTO WEB DA COMUNIDADE DEV´S AO

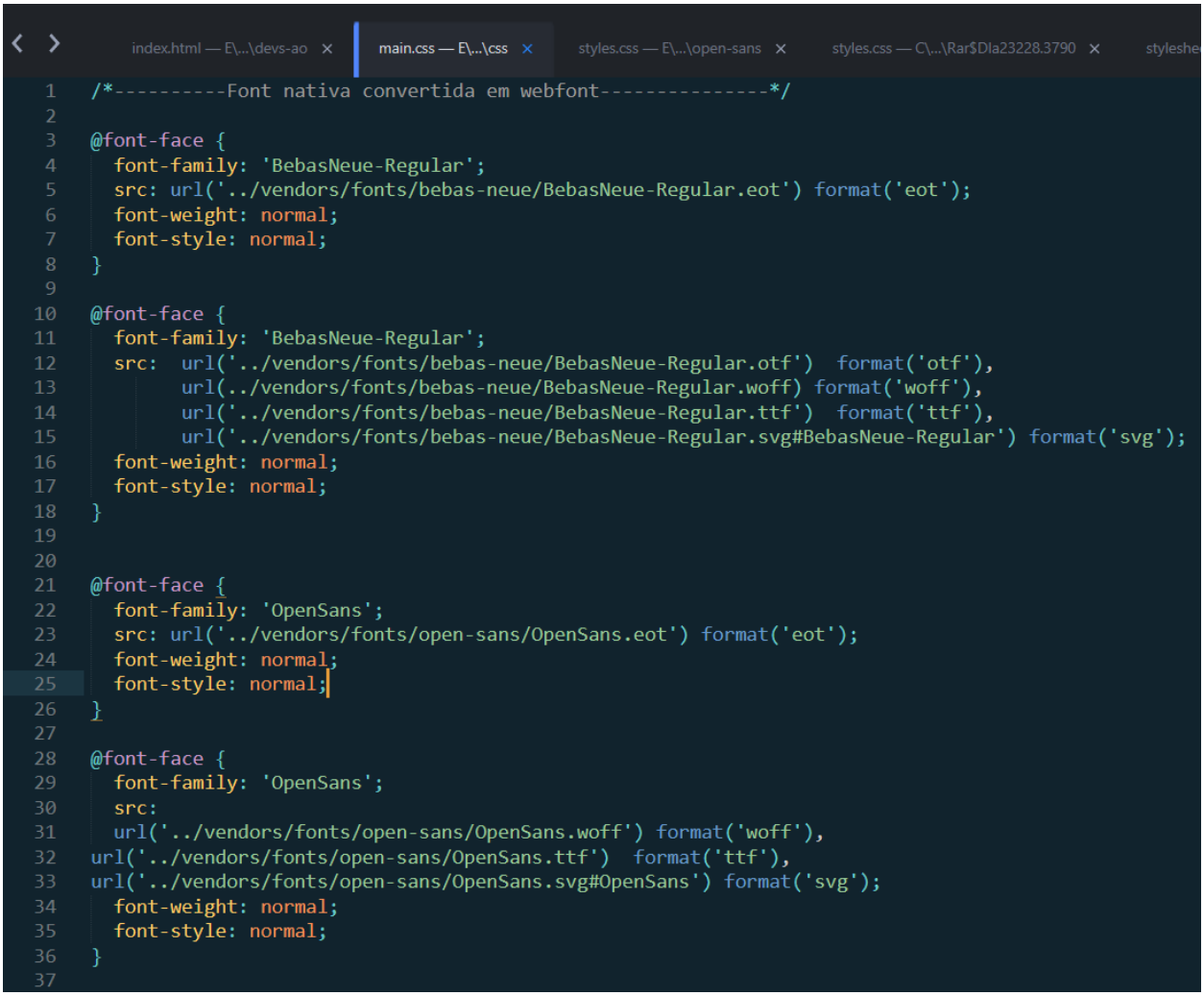
LINGUAGENS USADAS: HTML3 E CSS3(por enquanto).

RECURSOS EXTERNOS OBTIDO PELA INTERNET USADOS NA SITE

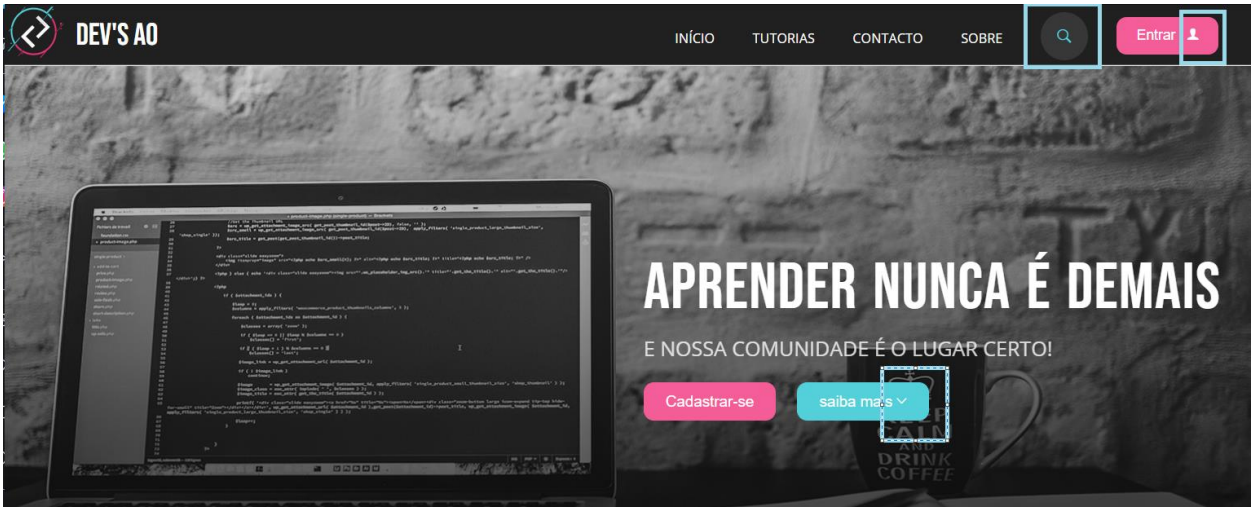
1 – Web Fonts: Bebas neue, OpensSans – Estas fontes não foram usadas no site via link do google font. As fontes foram baixadas, instaladas em meu pc e foram convertidas em webfonts através de gerador de webfonts(webfont generator - https://www.fontsquirrel.com/tools/webfont-generator). Os ficheiros de webfont obtido pelo gerador de webfont encontram-se no seguinte directório:

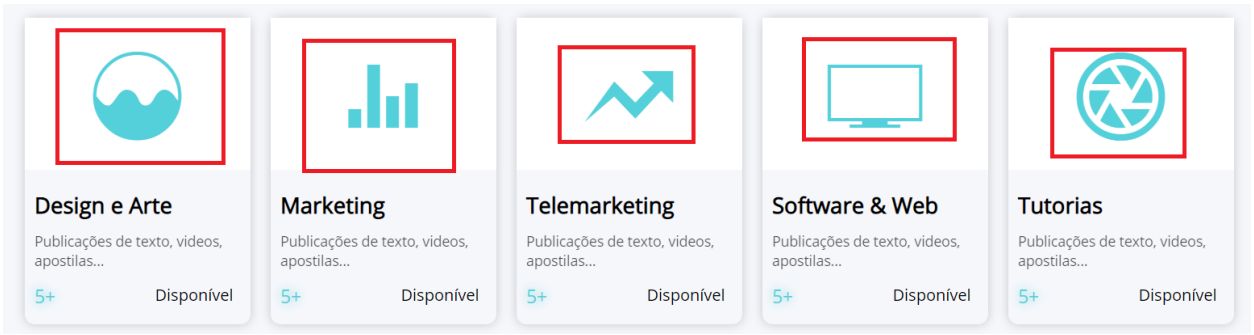
Font	Directório
Bebas nue	vendors/fonts/bebas-neue/
Opens sans	vendors/fonts/open-sans/

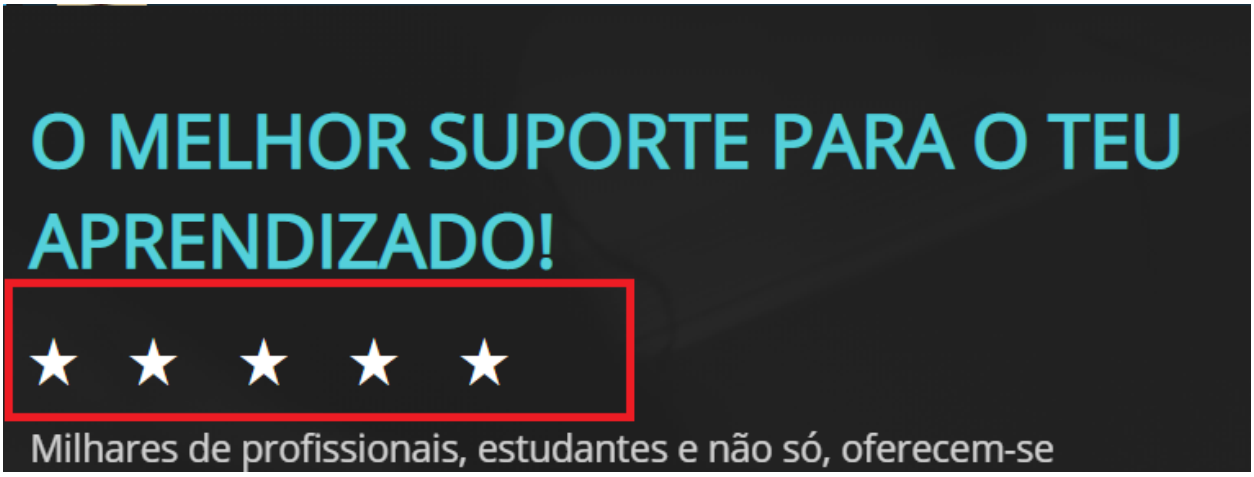
O ficheiro main.css (que é responsavel pelo css usado na index.html) connecta-se ao ficheiros de webfont através do webfonts listados assim tornando possivel o uso nas fontes no site.



2 – IconFonts (ícones): são ícones desenhado e baseado em material design e o nome “font” é porq eles são acompanhados de seus ficheiros em que na qual contêm os ícones já desenhados e estes tais ficheiros estão em formato de webfont, tanto que estes ícones são chamados na tag do html e podemos formata-la tal como formatamos qualquer tag de texto (como <p>,.<h1>,<h2>. etc) no css. Veja as IconFonts usadas no site







Existem várias ícons fonts, uma melhor que a outra, para este projecto, usou-se o ionicons. Baixando o ionicon no seu site oficial (https://ionicons.com/v2/) ele tráz um ficheiro com o nome ionicon.min.css. Este é o ficheiro que é conectado aoq index.html ou em qualquer uma outra página que se deseja usar o ícon font.

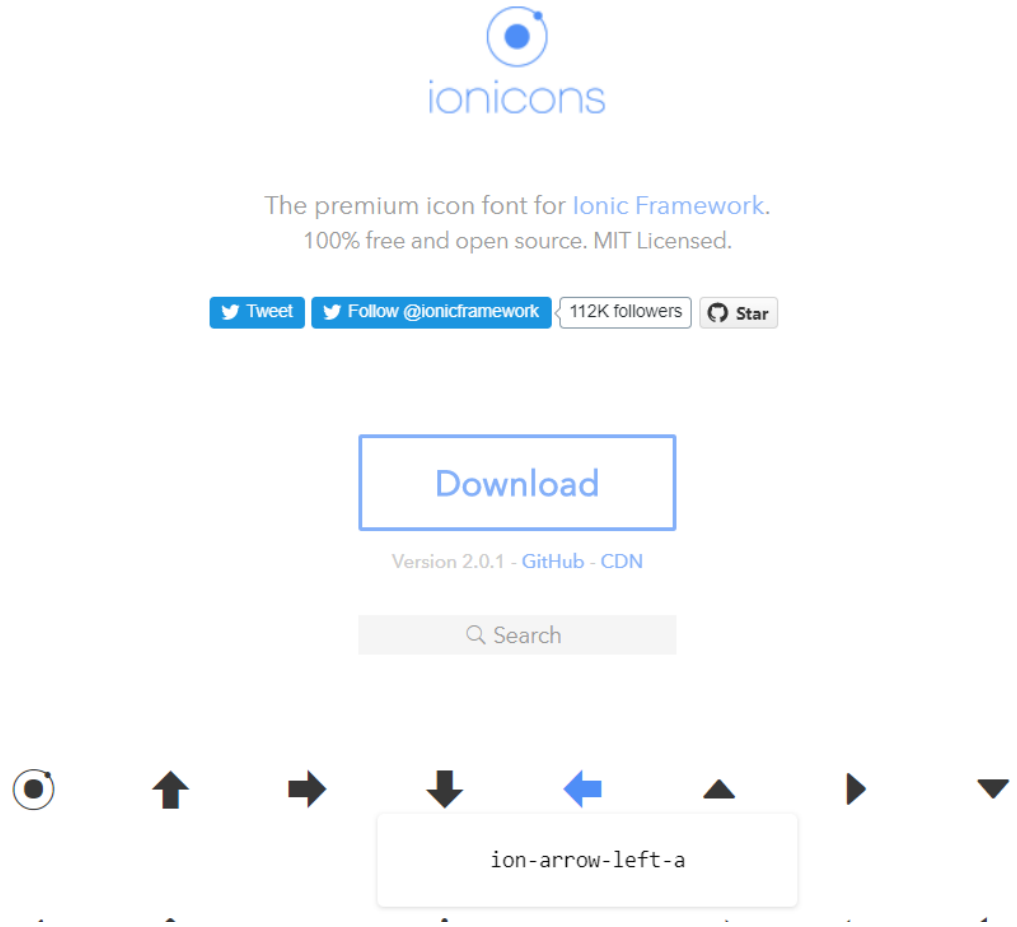
```
6    <link rel="stylesheet" href="vendors/css/ionicons.min.css">
```

Os ficheiros webfont do icon font estão em seguinte directório: **vendors/fonts/**

Para chamar a um icon font basta dentro da tag <i> chamar o nome do ícone dentro de uma class.

```
<i class="ion-ios-search-strong"></i>
```

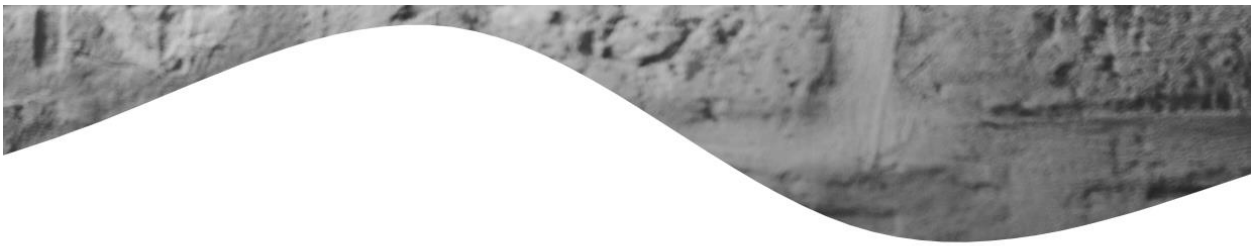
No site do ionicons.com/v2/ é lá aonde você obtem os nomes do ícones e poder chama-las em class da tag <i> no html.

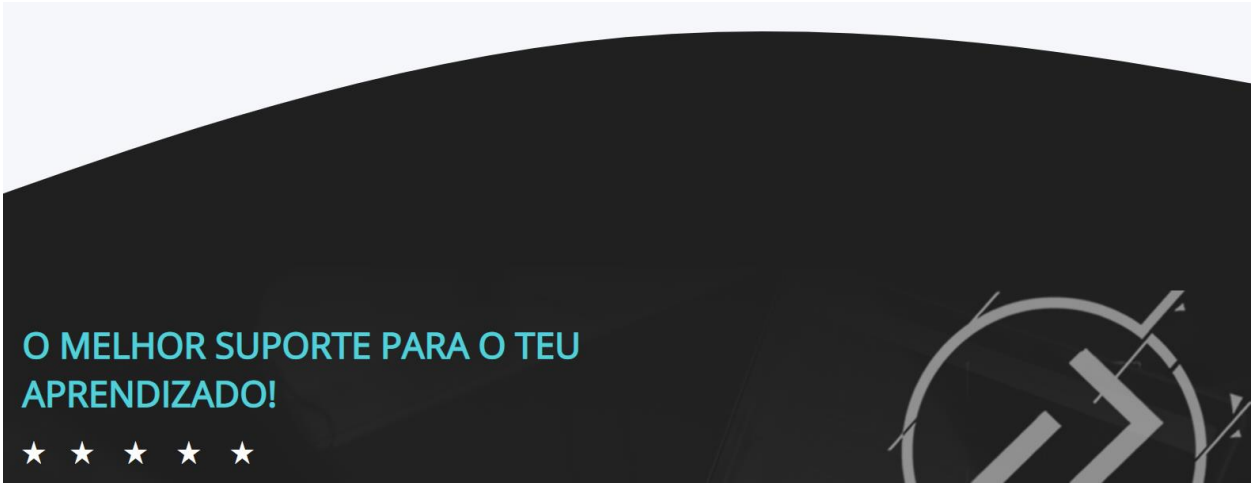


Obs: também pode dar uma olhada no “fontawesome”. É o iconfont mais usado hojeem dia

3 – SVG: Trata-se de uma linguagem XML para descrever de forma vetorial desenhos e gráficos bidimensionais. Por ser uma linguagem de XML o SVG nos permite alterar as suas propreidades, como cores, tamanho etc e pode ser formatado no css também.

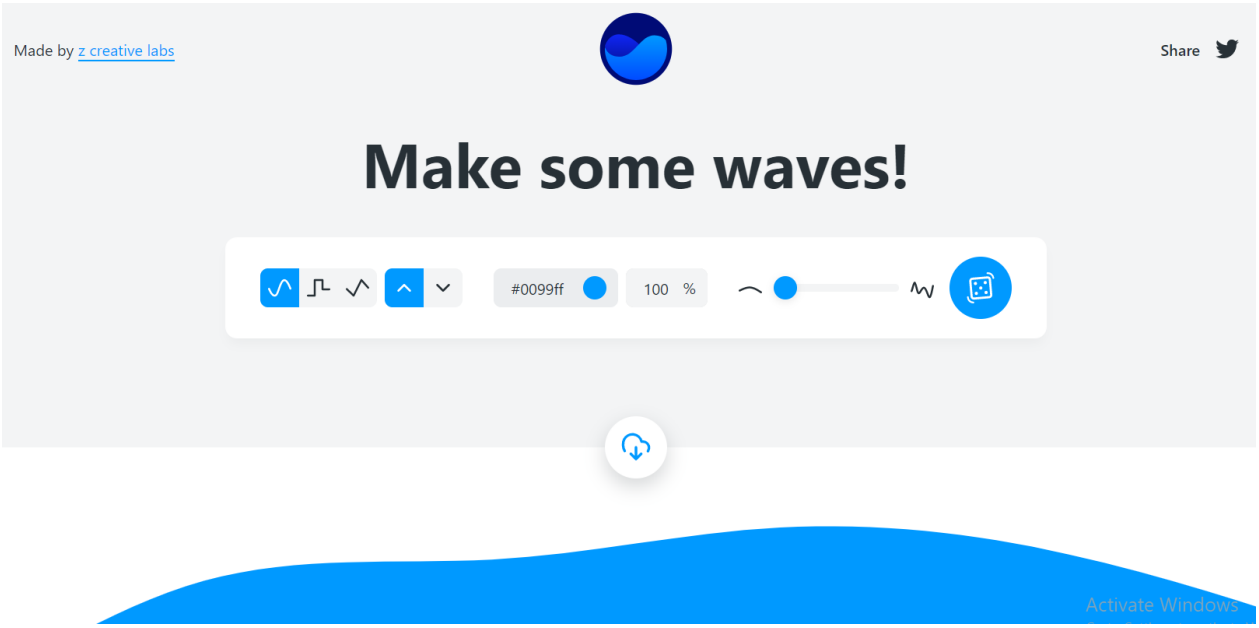
Usou-se o SVG para dar aquele efeito de onda branco no banner e também outras secções que contém o mesmo efeito.





Para obter o SVG usado no site usou-se um gerador de svg https://getwaves.io. No gerador, é possível personalisar as definições vectórias gráficas do SVG. O gerador gera um código SVG que basta copiar e colar no ficheiro html e usar-lo.

Este é o gerador de SVG https://getwaves.io



Cá está o código SVG obtido pelo gerador e usado na página

```
</nav>
    <svg class="svg1" xmlns="http://www.w3.org/2000/svg" viewBox="0 0 1440 320"><path fill="white" fill-opacity="1"
    d="M0,224L48,213.3C96,203,192,181,288,144C384,107,480,53,576,74.7C672,96,768,192,864,224C960,256,1056,224,1152,1
    92C1248,160,1344,128,1392,112L1440,96L1440,320L1392,320C1344,320,1248,320,1152,320C1056,320,960,320,864,320C768,
    320,672,320,576,320C480,320,384,320,288,320C192,320,96,320,48,320L0,320Z"></path></svg>
<div class="row">
```

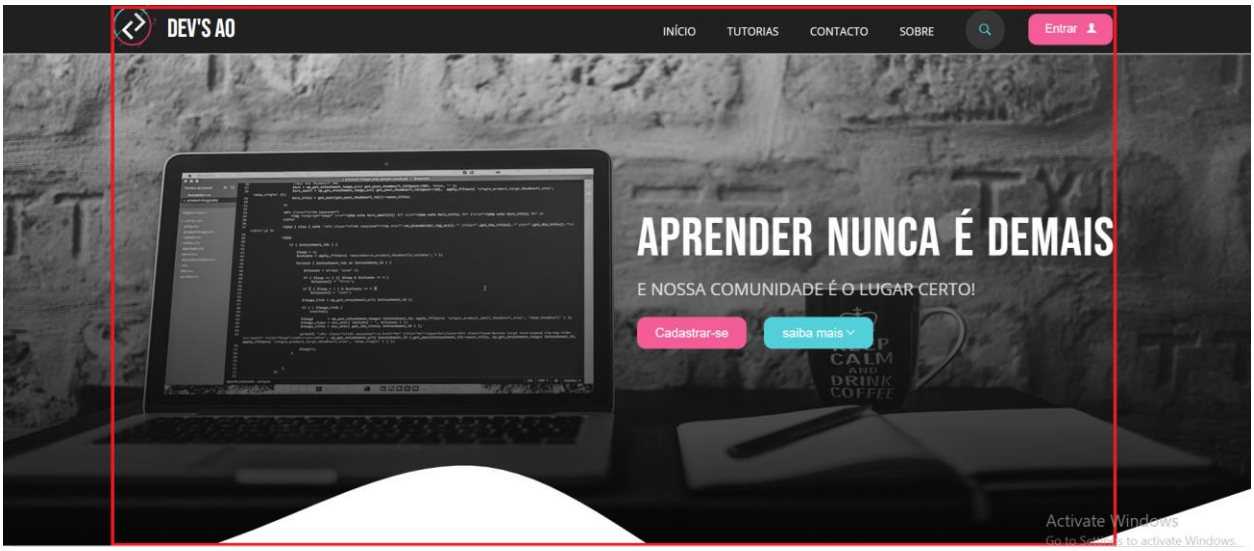
CONFIGURAÇÕES DE ALGUNS ELEMENTOS E CLASSES ÚTEIS NO CSS

1 - .row

.row: essa class serve para delimitar todo conteúdo do página e posicionar-la ao centro. Como alguns desenvolvedores que usam bootstrap já têm idea de uma class row. Essa class tem a mesma função a única diferença é que para esta página, foi feito em css puro mas não muda nada e faz a mesma coisa.

```css
.row {
    max-width: 1220px;
    margin: 0 auto;
    border: 0px solid white;
}
```

Obs. A propriedade “margin: 0 auto;” é que faz com que a div se posiciona ao centro. Isso dá 0 às margin top e bottom e automaticamente afasta-se dos espaços das laterais restante consoante o tamanho da div. Uma vés que o width da div não oucupa todo tamanho da sua tag mãe então ela se posiciona ao centro.



O .row evita com que estejamos a dar padding em divs diferentes e tentar alinnhar fazendo com que as divs estejam delimitadas à mesma linha do padding. Pois não seria muito eficáz.

Como é necessário que apenas o conteúdo da página fique centralizado e não própriamente a página por completo então a página não é suportada por apenas uma .row como se estivesse a ser usado um único container. Neste caso, toda vés que se inicniar uma div mãe para qualquer conteúdo, deve-se a chamar a class .row

```
index.html — E\...\devs-ao   main.css — E\...\css   styles.css — E\...\open-sans   styles.css — C\...\Rar$DIa23228.3790   stylesheet.css

12      <nav>
13          <div class="row">
14              <img class="logo" src="imagens/so-logo-branco.png">
15              <span class="logo-span">DEV's AO</span>
16              <div class="nav-direita">
17                  <ul class="main-nav">
18                      <li><a class="activa" href="#">Início</a></li>
19                      <li><a href="#">Tutorias</a></li>
20                      <li><a href="#">Contacto</a></li>
21                      <li><a href="#">sobre</a></li>
22                      <li><a href="#"><i class="ion-ios-search-strong"></i></a></li>
23
24                  </ul>
25                  <button class="btn-entrar">Entrar <i class="ion-person"></i> </button>
26
27              </div>
28          </div>
29      </nav>
30          <svg class="svg1" xmlns="http://www.w3.org/2000/svg" viewBox="0 0 1440 320"><path fill="whi
            d="M0,224L48,213.3C96,203,192,181,288,144C384,107,480,53,576,74.7C672,96,768,192,864,224C96
            92C1248,160,1344,128,1392,112L1440,96L1440,320L1392,320C1344,320,1248,320,1152,320C1056,320
            320,672,320,576,320C480,320,384,320,288,320C192,320,96,320,48,320L0,320Z"></path></svg>
31      <div class="row">
32
33          <div class="hero">
34
35              <h1>Aprender nunca é demais</h1>
36              <h3>e nossa comunidade é o lugar certo!</h3>
37              <button class="btn-primario">Cadastrar-se</button>
38              <button class="btn-secundario">saiba mais <i class="ion-ios-arrow-down"></i></button>
39          </div>
40      </div>
```

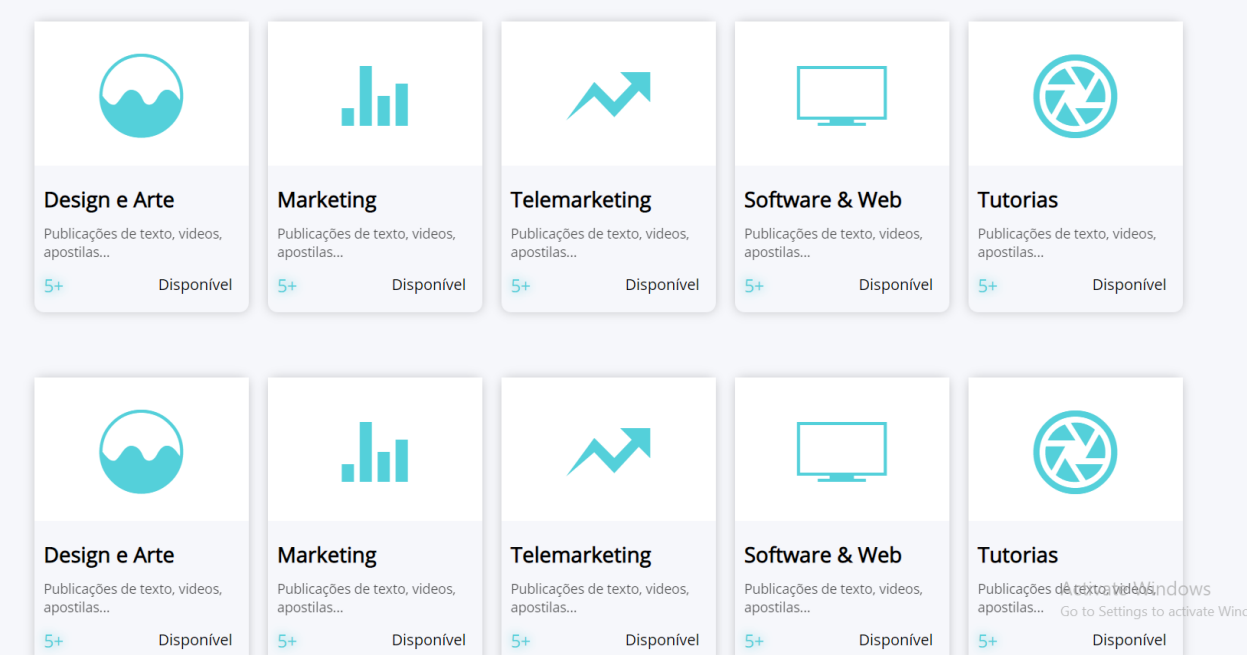
2 - <section>

usou-se a tag <section></section> para diferenciar cada tipo de conteúdo em nossa página, então, com o css, aplicou-se um padding de 80px para o Top e Bottom de formas que em todas secções tenham o mesmo whitespace antes e depois que termina o conteúdo.

```
section {
    padding: 80px 0;
}
```

OUTRAS DICAS BOAS USADAS PARA ORGANIZAÇÃO DO LAYOUT

CSS GRIDS



Usou-se Css Grids para dispor estes modern Cards conforme mostra a imagem acima. Praticamente para as cartas usou-se a tag <ul> que normalmente é usada para listar uma informação. E porque estas cartas também formam um conjunto de informação listadas mas expostas graficamente como se fosse uma grelha.

Neste caso a tag <ul> é a tag mãe que contém o “display: grid” e serve como o container e cada carta é uma tag <li> que contém os conteúdos de cada carta.

```css
.categoria-popular ul {
    width: 100%;
    display: grid;
    grid-template-columns: repeat(5, auto);
    list-style: none;
}
.categoria-popular ul  li{
    border: 0px solid black;
    margin-right: 1em;
    margin-bottom: 3.4em;
    border-radius: 10px;
    box-shadow: 0 0 10px rgba(0,0,0,.2);

}
```

A propriedade “grid-template-columns” é para definir que todas as (que nesse caso é a tag filha) serão expostas em colunas. O valor repeat(5, auto), diz que serão expostas 5 colunas e para cada coluna o tamanho será automaticamente ajustado de formas a caber 5 colunas em cada linha. Isso é, Nem que tivermos mais de cinco <li> em nossa lista, automaticamente uma nova linha será criada e as

continuarão a ser listadas na mesma ordem(5 colunas por cada linha). Se na nova linha tiver mais de 5 <li> novamente, será criada uma nova linha e acontecerá a mesma coisa e assim vai se sucedendo.

Uma outra forma de definir o números de colunas na Grid é definir o tamanho(valor em px,%,re, etc) que cada <li> irá oucupar e repetir o mesmo valor de acordo o número de colunas que deseja-se obter.

```
.categoria-popular ul {
    width: 100%;
    display: grid;
    grid-template-columns: 20% 20% 20% 20% 20%;
    list-style: none;
}
```

Neste caso repete-se o valor “20%” cinco vezes e conclua-se que o número repitido do valor define o número de colunas do css grid. Este mêtodo(20% 20% 20% 20% 20%) foi usado na página mas o seu problema é que este mêtodo compromete a responsividade neste caso. por isso optou-se por usar a primeira opção. Talvéz em uma outra qualquer altura poderá se usar o segundo mêtodo.

FLEX-BOX

Come vê-se, na imagem abaixo, o conteúdo está dividido em duas partes, uma que contém o texto e outra que contém a logotipo da comunidade em cinza. Ou seja, temos uma div mãe que serve como container e esta mesma div contém o “display: box”. E nesta div mãe temos duas divs expostas pelo flex box. Oque faz com que automaticamente estejam expostos um ao lado do outro sem ter que fazer muitas configurções css nas mesmas divs.


```
453  .flex-box-container {
454      display: flex;
455
456  }
```

```
457  .flex-box-1-of-2{
458      flex: 50%;
459  }
```

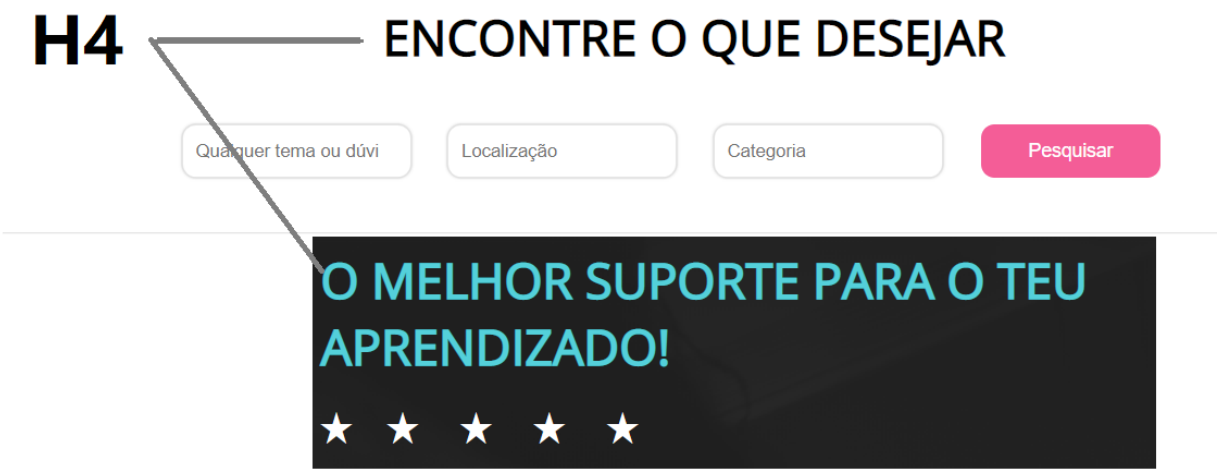
TYPOGRAPHY (TÍTULOS/HEADINGS)

Alguns elementos de Heading(títulos) foram configurado em css e repetido no html para exercer as mesmas funções. Exemplificando, para todo título de uma secção usou-se a heading H4. Pois faz com quem a página têm um padrão e matenha a consistência em termos de textualização(typography).

H1 – Usado e somente no banner para a título principal dado como intro (feacture em inglês em termos de UX/Design) que é texto em que se expressa uma mensagem cativante que de imediato chama a atenção do usuário.

H3 – Usado e somente no baner como o subtítulo fazendo parte do intro.

H4 – Usado e somente nos títulos de cada <section>. Para mostrar assim em destaque que a seguinte secção aborda um conteúdo relevante ao título, então é nesseçário que o título tenha um tamanho de fonte ou cor diferente de modos a dar a êmfase e diferenciar o título do texto.



H5 – Usado e somente para títulos do conteúdo dentro das cartas

P – usado como o conteúdo principal ou aquele texto longo de qualquer secção (note que algumas secções têm configuração do css diferente mais similar. As vezes até mesmo só font-size é que não é o mesmo. Como no caso do texto usado nas cartas que tente a ser menor com relação aos outros textos).

Span – usou-se para diferenciar um texto do outro em qualquer secção. Até ao momento usou-se o span para os dizeres na logo e para os números nas cartas.

Cá temos o exemplo de alguns lugares da página aonde foram usados os seguintes elementos citados a cima

OBRIGADO