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

Unidade Curricular: Design Hipermédia — **1 º Ano/1º Semestre**

Docente: Marco Miguel Olival Olim — **Data** 27/11/2019

ESTE EXERCÍCIO PRETENDE EXPLORAR AS DIFERENTES TÉCNICAS DE MICROINTERAÇÃO COM CSS

Microinterações são animações subtis, desencadeadas quando o utilizador interage com uma parte específica do UI. A grande mais valia das microinterações é proporcionar um melhor feedback visual ao utilizador e, consequentemente, uma melhor experiência de utilização da aplicação web e usabilidade em geral.

1. Comece por criar um **index.html** com o habitual markup e um botão com a classe "botao":

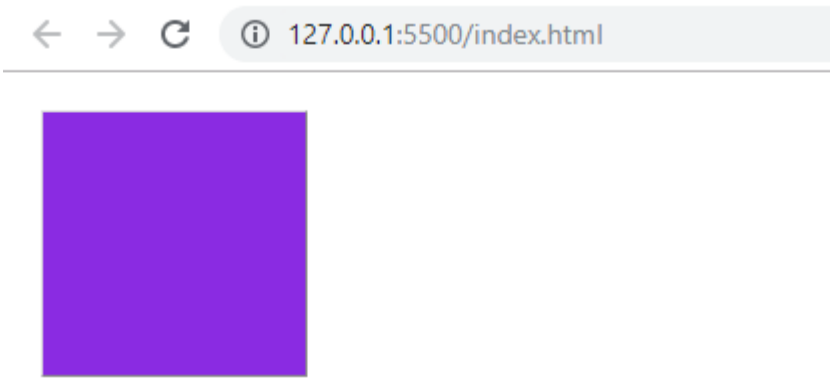
index.html > html > body > button.botao

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initia
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Document</title>
</head>
<body>
    <button class="botao"></button>
</body>
```

2. Pode associar um ficheiro CSS ou criar uma tag style para formatar este elemento:

```
<body>
    <button class="botao"></button>
    <style>
        .botao{
            width:200px;
            height: 200px;
            margin:20px;
            background-color: blueviolet;
        }
    </style>
</body>
```

3. Obtemos:



Cofinanciado por:

MADEIRA 14-20 — REGIÃO AUTÓNOMA DA MADEIRA — PORTUGAL 2020 — UNIÃO EUROPEIA Fundo Social Europeu

4. Queremos alterar o elemento ao passar o rato por cima deste. Para o efeito usamos o seletor :hover

```
index.html > html > body > style > .botao:hover
<style>
    .botao{
        width:200px;
        height: 200px;
        margin:20px;
        background-color: blueviolet;
    }
    .botao:hover{
        width: 400px;
    }
</style>
```

5. Neste momento o efeito é abrupto ao passar sobre o elemento, redimensionando logo para 400px

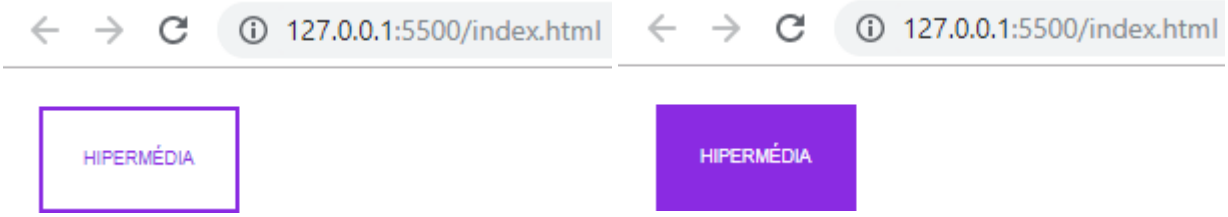
6. Basta adicionar as propriedades da transição para que a animação seja ativada

```
index.html > html > body > style > .botao
<style>
    .botao{
        width:200px;
        height: 200px;
        margin:20px;
        background-color: blueviolet;

        transition-duration: 200ms;
    }
    .botao:hover{
        width: 400px;
    }
</style>
```

7. Vamos alterar as propriedades para que se assemelhe mais a um botão

```
<body>
    <button class="botao">HIPERMÉDIA</button>
    <style>
        .botao{
            width:150px;
            height: 80px;
            margin:20px;
            background-color: transparent;
            border:none;
            color: blueviolet;
            cursor: pointer;

            transition-duration: 200ms;
        }
        .botao:hover{
            width: 400px;
        }
</style>
```

8. Vamos também alterar a animação para em vez de mudar as dimensões alterar as cores:



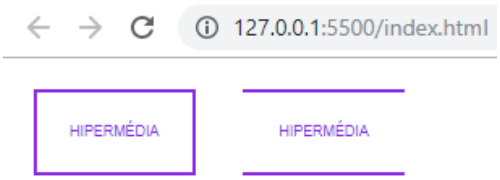
Cofinanciado por:

MADEIRA 14-20 · REGIÃO AUTÓNOMA DA MADEIRA · PORTUGAL 2020 · UNIÃO EUROPEIA Fundo Social Europeu

9. Criamos então um id próprio de cada botão:

```
<body>
    <button class="botao" id="anim1">HIPERMÉDIA</button>
    <style>
        .botao{
            width:150px;
            height: 80px;
            margin:20px;
            background-color: transparent;
            border:none;
            color: blueviolet;
            cursor: pointer;
        }
        #anim1{
            border: 3px solid blueviolet;
            transition-duration: 200ms;
        }
        #anim1:hover{
            background-color: blueviolet;
            color:white;
        }
    </style>
</body>
```

10. Duplicamos o botão para criarmos outra animação só com duas linhas em cima e em baixo no botão:



11. Usamos o pseudo seletores ::before e ::after para gerar estas linhas

index.html > html > body > button#anim2.botao

```
<body>
    <button class="botao" id="anim1">HIPERMÉDIA</button>
    <button class="botao" id="anim2">HIPERMÉDIA</button>
    <style>
        .botao{
            width:150px;
            height: 80px;
            margin:20px;
            background-color: transparent;
            border:none;
            color: blueviolet;
            cursor: pointer;
        }
        #anim2{
            position: relative;
        }
        #anim2::before{
            position:absolute;
            content:'';
            top:0;
            left:0;
            width:100%;
            height: 3px;
            background-color: blueviolet;
        }
        #anim2::after{
            position:absolute;
            content:'';
            bottom:0;
            right:0;
            width:100%;
            height: 3px;
            background-color: blueviolet;
        }
        #anim1{
```

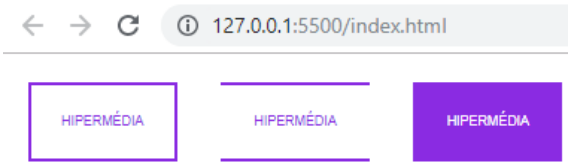
Cofinanciado por:

MADEIRA 14-20 REGIÃO AUTÓNOMA DA MADEIRA PORTUGAL 2020 UNIÃO EUROPEIA Fundo Social Europeu

12. Colocamos estas linhas a zero (width:0) no início da transição e a 100% aquando do hover

```
index.html > html > body > style > #anim2:hover::before
22      #anim2{
23          position: relative;
24      }
25      #anim2::before{
26          position:absolute;
27          content:'';
28          top:0;
29          left:0;
30          width:0;
31          height: 3px;
32          background-color: blueviolet;
33
34          transition-duration: 200ms;
35      }
36      #anim2::after{
37          position:absolute;
38          content:'';
39          bottom:0;
40          right:0;
41          width:0;
42          height: 3px;
43          background-color: blueviolet;
44
45          transition-duration: 200ms;
46      }
47      #anim2:hover::before,
48      #anim2:hover::after{
49          width:100%;
50      }
51      #anim1{
```

13. O próximo botão será semelhante ao primeiro, mas em vez de uma transição em fade usamos em slide



14. Note que temos de colocar o z-index antes dos outros para que o texto não desapareça por baixo do fundo:

```
12      <button class="botao" id="anim3">HIPERMÉDIA</button>
13      <style>
14          .botao{
15              width:150px;
16              height: 80px;
17              margin:20px;
18              background-color: transparent;
19              border:none;
20              color: blueviolet;
21              cursor: pointer;
22          }
23          #anim3{
24              position: relative;
25              color:white;
26          }
27          #anim3::after{
28              position:absolute;
29              content:'';
30              bottom:0;
31              left:0;
32              width:100%;
33              height: 100%;
34              background-color: blueviolet;
35              z-index: -1;
36          }
37          #anim2{
```

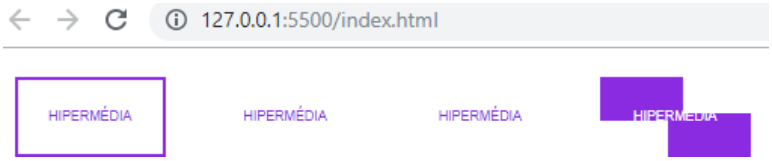
Cofinanciado por:

MADEIRA 14-20 · REGIÃO AUTÓNOMA DA MADEIRA · PORTUGAL 2020 · UNIÃO EUROPEIA Fundo Social Europeu

15. Colocamos a cor das letras no seletor :hover e definimos a altura a zero no estado inicial da transição:

```
}
#anim3{
    position: relative;

    transition-duration: 200ms;
}
#anim3::after{
    position:absolute;
    content:'';
    bottom:0;
    left:0;
    width:100%;
    height: 0;
    background-color: blueviolet;
    z-index: -1;

    transition-duration: 200ms;
}
#anim3:hover{
    color: white;
}
#anim3:hover::after{
    height: 100%;
}
```

16. Vamos combinar o efeito de slide deste último botão com as duas linhas do segundo botão



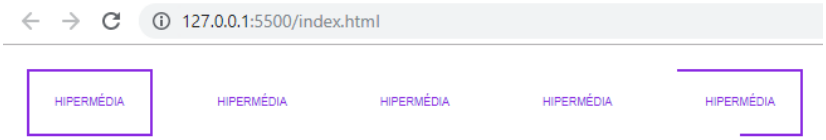
17. Comece por criar mais um botão com o id #anim4:

```
}
#anim4{
    position: relative;
    transition-duration: 200ms;
}
#anim4::before{
    position:absolute;
    content:'';
    top:0;
    left:0;
    width:0;
    height: 0;
    background-color: blueviolet;
    z-index: -1;
    transition: 200ms;
}
#anim4::after{
    position:absolute;
    content:'';
    bottom:0;
    right:0;
    width:0;
    height: 0;
    background-color: blueviolet;
    z-index: -1;
    transition: 200ms;
}
#anim4:hover{
    color: white;
}
#anim4:hover::before,
#anim4:hover::after{
    width:100%;
    height:100%;
}
#anim3{
```

Cofinanciado por:

MADEIRA 14-20 REGIÃO AUTÓNOMA DA MADEIRA PORTUGAL 2020 UNIÃO EUROPEIA Fundo Social Europeu

18. Voltando ao segundo exemplo, propõe-se agora que a linha dê a volta ao botão numa animação contínua:



19. Inicialmente duplicamos os estilos do #anim2 e acrescentamos apenas um transition-delay no ::after

```
<button class="botao" id="anim5">HIPERMÉDIA</button>
<style>
    .botao{ ···
    }
    #anim5{
        position: relative;
    }
    #anim5::before{
        position:absolute;
        content:'';
        top:0;
        left:0;
        width:0;
        height: 3px;
        background-color: blueviolet;

        transition-duration: 200ms;
    }
    #anim5::after{
        position:absolute;
        content:'';
        bottom:0;
        right:0;
        width:0;
        height: 3px;
        background-color: blueviolet;

        transition-delay: 200ms;
        transition-duration: 200ms;
    }
    #anim5:hover::before,
    #anim5:hover::after{
        width:100%;
    }
    #anim4{
```

20. Para desenhar as outras duas linhas precisamos de usar novamente os pseudo-seletores ::before e ::after mas estes já foram usados. De forma a dispormos de mais dois pseudo-seletores ::before e ::after incluímos então um novo elemento no botão:

```
<button class="botao" id="anim5"><span id="anim5-span">HIPERMÉDIA</span></button>
<style>
```

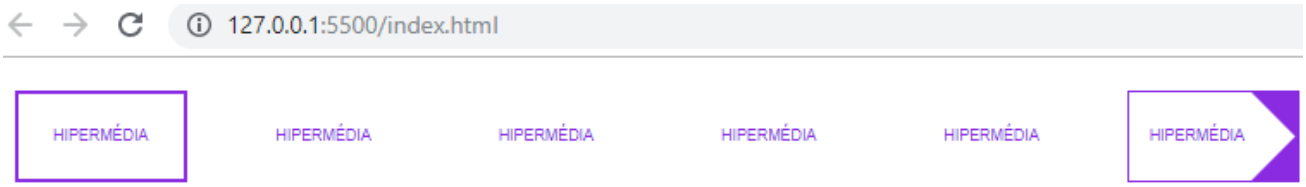
21. Atente que neste elemento alteramos é a altura. Sincronize agora os delays para completar a animação.

```
        transition-delay: 400ms;
        transition-duration: 200ms;
    }
    #anim5-span::after{
        position:absolute;
        content:'';
        top:0;
        right:0;
        width:3px;
        height: 0;
        background-color: blueviolet;
        transition-delay: 200ms;
        transition-duration: 200ms;
    }
    #anim5-span:hover::before,
    #anim5-span:hover::after{
        height:100%;
    }
    #anim5:hover::before,
    #anim5:hover::after{
        width:100%;
    }
    #anim4{
```

Cofinanciado por:

MADEIRA 14-20 · REGIÃO AUTÓNOMA DA MADEIRA · PORTUGAL 2020 · UNIÃO EUROPEIA Fundo Social Europeu

22. Por último, é proposta uma animação com uma forma mais complexa:



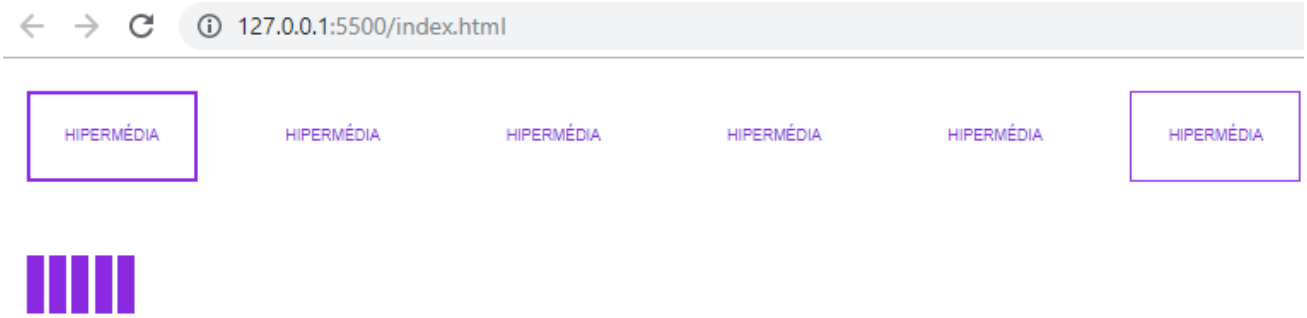
23. Para obter o triângulo usamos agora o border:

```
<button class="botao" id="anim6"><span id="text">HIPERMÉDIA</span></button>
<style>
    .botao{…
    }
    #anim6{
        position: relative;
        border: 2px solid blueviolet;
        transition-duration: 0.4s;
    }
    #anim6::after{
        content: '';
        position: absolute;
        top: 0;
        right: 0;
        width: 0;
        height: 0;
        border-style: solid;
        border-width: 0 0 0 0;
        border-color: transparent blueviolet transparent transparent;
        transition-duration: 0.4s;
    }
    #anim6::before{
        content: '';
        position: absolute;
        bottom: 0;
        right: 0;
        width: 0;
        height: 0;
        border-style: solid;
        border-width: 0 0 0 0;
        border-color: transparent transparent blueviolet transparent;
        transition-duration: 0.4s;
    }
    #text{
        transition-duration: 0.4s;
    }

    #anim6:hover > #text{
        padding-right: 30px;
    }

    #anim6:hover::after{
        border-width: 0 40px 40px 0;
    }

    #anim6:hover::before{
        border-width: 0 0 40px 40px;
    }
```

Cofinanciado por:

MADEIRA 14-20 · REGIÃO AUTÓNOMA DA MADEIRA · PORTUGAL 2020 · UNIÃO EUROPEIA Fundo Social Europeu

24. Além do Transition dispomos também do animation para criar animações com CSS, sendo que neste último é também possível definir múltiplos passos na animação com @keyframes. Vamos então criar um Loader:



25. Estas cinco barras serão geradas no markup por 5 divs dentro de um div com classe ".loader"

```
index.html > html > body > div.loader
16      <div class="loader">
17          <div class="barra1"></div>
18          <div class="barra2"></div>
19          <div class="barra3"></div>
20          <div class="barra4"></div>
21          <div class="barra5"></div>
22      </div>
23      <style>
```

26. Definimos então a formatação inicial do .loader e de cada uma das barras

```
index.html > html > body > style > .loader
22      </div>
23      <style>
24          .loader{
25              height: 100px;
26              width: 95px;
27              margin: 20px;
28              display: flex;
29              justify-content: space-between;
30              align-items: center;
31          }
32          .barra1,
33          .barra2,
34          .barra3,
35          .barra4,
36          .barra5{
37              height: 50px;
38              width: 15px;
39              background-color: blueviolet;
40          }
41          .botao{ ...
```

27. De seguida definimos então os @keyframes da nossa animação com o nome "animação":

```
36          .barra5{
37              height: 50px;
38              width: 15px;
39              background-color: blueviolet;
40
41              animation: animação 1.5s infinite;
42          }
43          @keyframes animação{
44              0%{
45                  transform: scaleY(1);
46              }
47              50%{
48                  transform: scaleY(2);
49              }
50              100%{
51                  transform: scaleY(1);
52              }
53          }
54          .botao{ ...
```

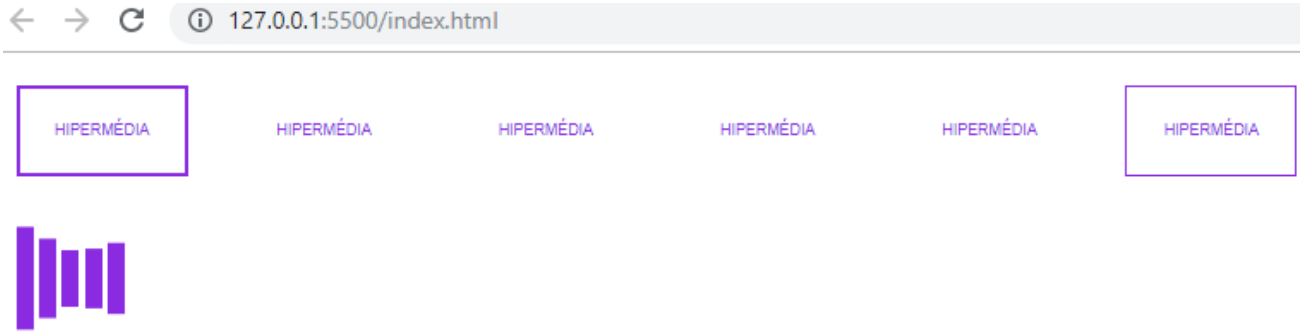
Cofinanciado por:

MADEIRA 14-20 · REGIÃO AUTÓNOMA DA MADEIRA · PORTUGAL 2020 · UNIÃO EUROPEIA Fundo Social Europeu

28. Neste momento todas as barras expandem-se ao mesmo tempo. Recorremos então ao animation-delay:

```
        100%{
            transform: scaleY(1);
        }
    }
    .barra2{
        animation-delay: 0.2s;
    }
    .barra3{
        animation-delay: 0.4s;
    }
    .barra4{
        animation-delay: 0.6s;
    }
    .barra5{
        animation-delay: 0.8s;
    }
    .botao{ ...
```

29. Ficamos então com uma animação intercalada, repetindo-se num ciclo infinito…



30. Vamos agora criar outros loaders mas, como estamos a utilizar elementos em bloco como o div, cada um destes ficariam numa nova linha. Para corrigir isto, envolvemos todos os loaders num flex container.

```
index.html > html > body > div.loaders
    <button class="botao" id="animo">HIPERMÉDIA</button>
    <div class="loaders">
        <div class="loader">
            <div class="barra1"></div>
            <div class="barra2"></div>
            <div class="barra3"></div>
            <div class="barra4"></div>
            <div class="barra5"></div>
        </div>
    </div>
    <style>
        .loaders{
            display:flex;
        }
```

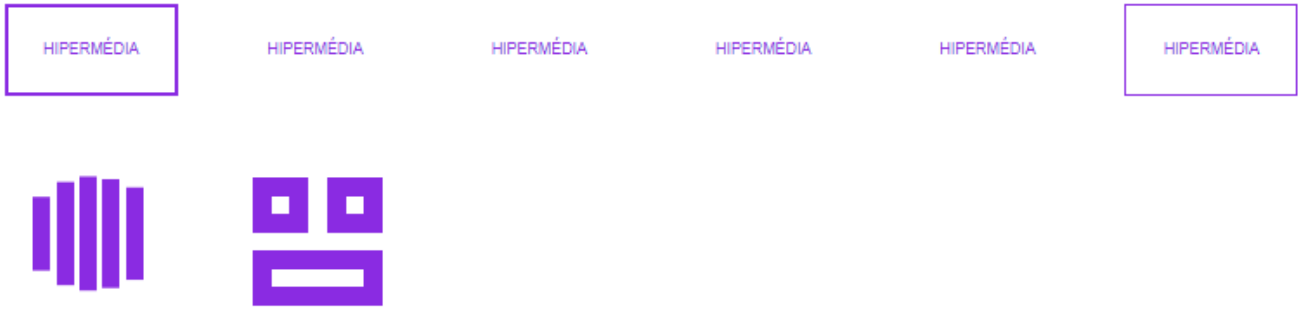
31. Este próximo loader irá necessitar apenas de três divs

```
index.html > html > body > div.loaders > div.loader2
    <div class="loaders">
        <div class="loader">
            <div class="barra1"></div>
            <div class="barra2"></div>
            <div class="barra3"></div>
            <div class="barra4"></div>
            <div class="barra5"></div>
        </div>
        <div class="loader2">
            <div class="box1"></div>
            <div class="box2"></div>
            <div class="box3"></div>
        </div>
    </div>
    <style>
```

Cofinanciado por:

MADEIRA 14-20 · REGIÃO AUTÓNOMA DA MADEIRA · PORTUGAL 2020 · UNIÃO EUROPEIA Fundo Social Europeu

32. Iremos animar cada uma destas divs, individualmente, de forma a obter o seguinte resultado:



33. Para obtermos esta configuração inicial destas boxes usamos a seguinte formatação:

```
index.html > html > body > style > .loader2
30      <style>
31          .loader2{
32              height: 112px;
33              width:112px;
34              margin: 50px;
35              position:relative;
36          }
37          .loader2 .box1,
38          .loader2 .box2,
39          .loader2 .box3{
40              display: block;
41              position:absolute;
42              box-sizing:border-box;
43              border: 16px solid blueviolet;
44          }
45          .loader2 .box1{
46              height: 48px;
47              width: 112px;
48              margin-top: 64px;
49              margin-left: 0;
50          }
51          .loader2 .box2{
52              height: 48px;
53              width: 48px;
54              margin-top: 0;
55              margin-left: 0;
56          }
57          .loader2 .box3{
58              height: 48px;
59              width: 48px;
60              margin-top: 0;
61              margin-left: 64px;
62          }
63          .loaders{
```

34. A animação de cada “box” terá a duração de 4s e, ao usar “ease”, esta deixa de ser linear e acelera no início:

```
45          .loader2 .box1{
46              height: 48px;
47              width: 112px;
48              margin-top: 64px;
49              margin-left: 0;
50              animation: animate1 4s forwards ease-in-out infinite;
51          }
52          .loader2 .box2{
53              height: 48px;
54              width: 48px;
55              margin-top: 0;
56              margin-left: 0;
57              animation: animate2 4s forwards ease-in-out infinite;
58          }
59          .loader2 .box3{
60              height: 48px;
61              width: 48px;
62              margin-top: 0;
63              margin-left: 64px;
64              animation: animate3 4s forwards ease-in-out infinite;
65          }
```

Cofinanciado por:

MADEIRA 14-20 REGIÃO AUTÓNOMA DA MADEIRA PORTUGAL 2020 UNIÃO EUROPEIA Fundo Social Europeu

35. Cada uma destas animações faz variar a largura e altura do elemento, compensando com a margem:

```
index.html > html > body > style > @keyframes animate1
        animation: animate3 4s forwards ease-in-out infinite;
    }
    @keyframes animate1{
        0%{
            width: 112px;
            height: 48px;
            margin-top: 64px;
            margin-left: 0px;
        }

        12.5%,25%, 37.5%, 50%, 62.5%{
            width: 48px;
            height: 48px;
            margin-top: 64px;
            margin-left: 0px;
        }
        75%{
            width: 48px;
            height: 112px;
            margin-top: 0px;
            margin-left: 0px;
        }
        87.5%,100%{
            width: 48px;
            height: 48px;
            margin-top: 0px;
            margin-left: 0px;
        }
    }
    @keyframes animate2{
```

36. A próxima animação terá de ser sincronizada com o obtido na primeira

```
    }
    @keyframes animate2{
        0%, 12.5%, 25%, 37.5%{
            width: 48px;
            height: 48px;
            margin-top: 0px;
            margin-left: 0px;
        }
        50%{
            width: 112px;
            height: 48px;
            margin-top: 0px;
            margin-left: 0px;
        }
        62.5%, 75%, 87.5%, 100%{
            width: 48px;
            height: 48px;
            margin-top: 0px;
            margin-left: 64px;
        }
    }
    @keyframes animate3{
```

37. Tente agora criar o animate3, sincronizado com os keyframes anteriores, de forma a obter este resultado:

Cofinanciado por:

MADEIRA 14-20
REGIÃO AUTÓNOMA DA MADEIRA
PORTUGAL 2020
UNIÃO EUROPEIA
Fundo Social Europeu

38. Verifique então se obteve os mesmos keyframes nesta última animação

```
@keyframes animate3{
    0%, 12.5%{
        width: 48px;
        height: 48px;
        margin-top: 0px;
        margin-left: 64px;
    }
    25%{
        width: 48px;
        height: 112px;
        margin-top: 0px;
        margin-left: 64px;
    }
    37.5%, 50%, 62.5%, 75%, 87.5%{
        width: 48px;
        height: 48px;
        margin-top: 64px;
        margin-left: 64px;
    }
    100%{
        width: 112px;
        height: 48px;
        margin-top: 64px;
        margin-left: 0px;
    }
}
```

39. Os próximos três loaders usam SVG (Scalable Vector Graphics) para desenhar as formas, sendo que estas também podem ser animadas por CSS (apesar do SVG também ter um processo de animação – o SMIL)

40. Começamos por definir o markup do loader3, loader4 e loader5, colocando os elementos vetoriais:

```
index.html > html > body > div.loaders
<div class="loaders">
    <div class="loader">…
    </div>
    <div class="loader2">…
    </div>
    <div class="loader3">
        <svg viewBox="0 0 80 80">
            <circle cx="40" cy="40" r="32"></circle>
        </svg>
    </div>
    <div class="loader4 triangle">
        <svg viewBox="0 0 86 80">
            <polygon points="43 8 79 72 7 72"></polygon>
        </svg>
    </div>
    <div class="loader5">
        <svg viewBox="0 0 80 80">
            <rect x="8" y="8" width="64" height="64"></rect>
        </svg>
    </div>
</div>
<style>
```

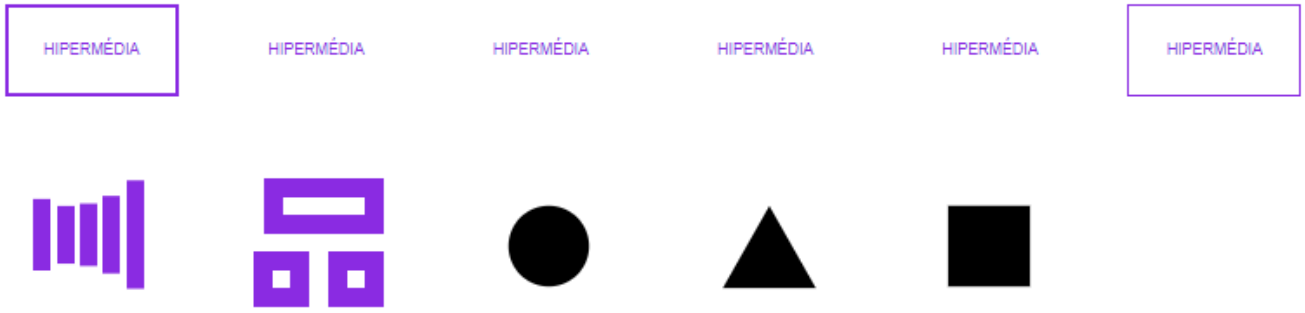
Cofinanciado por:

MADEIRA 14-20 · REGIÃO AUTÓNOMA DA MADEIRA · PORTUGAL 2020 · UNIÃO EUROPEIA Fundo Social Europeu

41. Neste momento os elementos não aparecem porque as dimensões são 0. Começamos por formatar isso:

```
index.html > html > body > style
                </svg>
            </div>
        </div>
        <style>
            .loader3,
            .loader4,
            .loader5{
                width: 88px;
                height: 88px;
                position: relative;
                margin: 65px 49px;
            }
            .triangle{
                width: 96px;
            }
```

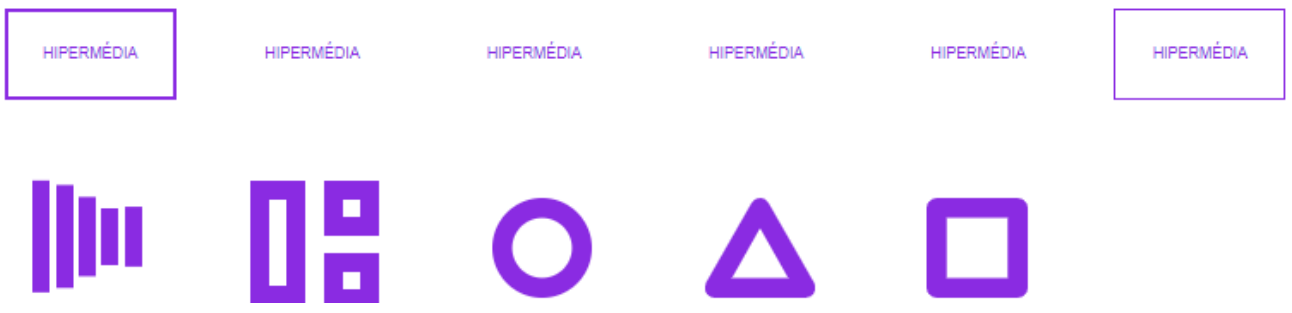
42. Deverá obter:



43. Só nos interessa a linha e não o preenchimento interior dos SVG, pelo formatamos as formas:

```
index.html > html > body > style > svg
            .triangle{
                width: 96px;
            }
            svg{
                display: block;
                width: 100%;
                height: 100%;
                stroke-linecap: round;
                stroke-linejoin: round;
            }
            rect,
            polygon,
            circle{
                fill: none;
                stroke: blueviolet;
                stroke-width: 16px;
            }
```

44. Resultando em:



Cofinanciado por:

MADEIRA 14-20 · REGIÃO AUTÓNOMA DA MADEIRA · PORTUGAL 2020 · UNIÃO EUROPEIA Fundo Social Europeu

45. Queremos agora a abertura de um espaço nestas formas. Uma forma de obter isto é alterar a linha da forma como tracejada e definir o tamanho dos traços e espaçamentos com stroke-dasharray:

```
index.html > html > body > style
65      polygon,
66      circle{
67          fill: none;
68          stroke: blueviolet;
69          stroke-width: 16px;
70      }
71      polygon{
72          stroke-dasharray: 145 76 145 76;
73          stroke-dashoffset: 0;
74      }
75      rect{
76          stroke-dasharray: 192 64 192 64;
77          stroke-dashoffset: 0;
78      }
79      circle{
80          stroke-dasharray: 150 50 150 50;
81          stroke-dashoffset: 75;
82      }
```

46. Definindo o tamanho do stroke e da gap obtem:

47. Colocamos agora o ponto usando o pseudo seletor ::before

```
index.html > html > body > style > .triangle::before
84      }
85      .loader3::before,
86      .loader4::before,
87      .loader5::before{
88          content: '';
89          width: 16px;
90          height: 16px;
91          border-radius: 50%;
92          position: absolute;
93          top: 74px;
94          left: 38px;
95          background: limegreen;
96          display: block;
97          transform: translate(-36px, -36px);
98      }
99
100     .triangle::before{
101         left: 42px;
102         transform: translate(-20px, -36px);
103     }
```

Cofinanciado por:

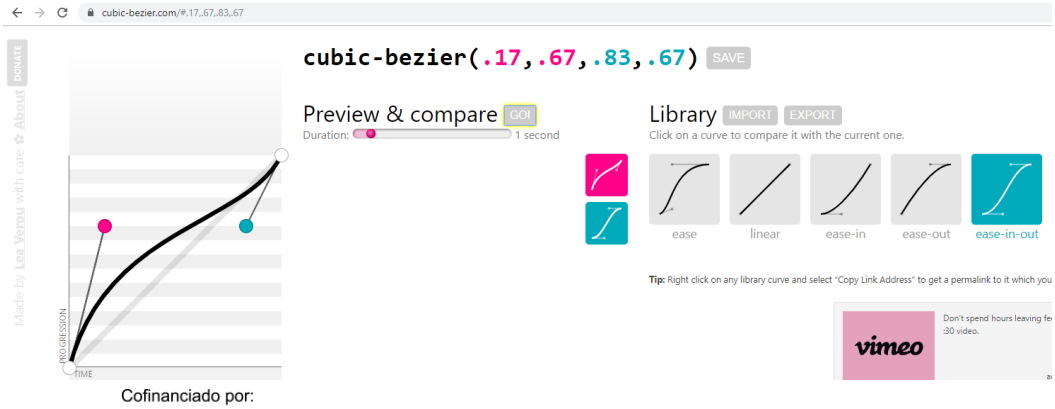
MADEIRA 14-20 · REGIÃO AUTÓNOMA DA MADEIRA · PORTUGAL 2020 · UNIÃO EUROPEIA Fundo Social Europeu

48. Foi usado o translate para posicionar o ponto na gap da forma:

49. Vamos agora adicionar as animações. Para evitar que sejam lineares vamos aplicar um “easing” mas, em vez de usar um pré-definido com o “ease-in-out”, vamos definir a curva da aceleração com uma cubic-bezier:

```
        polygon{
            stroke-dasharray: 145 76 145 76;
            stroke-dashoffset: 0;
            animation: pathTriangle 3s cubic-bezier(.785, .135, .15, .86) infinite;
        }
        rect{
            stroke-dasharray: 192 64 192 64;
            stroke-dashoffset: 0;
            animation: pathRect 3s cubic-bezier(.785, .135, .15, .86) infinite;
        }
        circle{
            stroke-dasharray: 150 50 150 50;
            stroke-dashoffset: 75;
            animation: pathCircle 3s cubic-bezier(.785, .135, .15, .86) infinite;
        }
        .loader3::before,
        .loader4::before,
        .loader5::before{
            content: '';
            width: 16px;
            height: 16px;
            border-radius: 50%;
            position: absolute;
            top: 74px;
            left: 38px;
            background: limegreen;
            display: block;
            transform: translate(-36px, -36px);
            animation: dotRect 3s cubic-bezier(.785, .135, .15, .86) infinite;
        }

        .triangle::before{
            left: 42px;
            transform: translate(-20px, -36px);
            animation: dotTriangle 3s cubic-bezier(.785, .135, .15, .86) infinite;
        }
```

50. Podemos obter (e visualizar) estas curvas a partir de sites como https://cubic-bezier.com/:



Cofinanciado por:

MADEIRA 14-20 · REGIÃO AUTÓNOMA DA MADEIRA · PORTUGAL 2020 · UNIÃO EUROPEIA · Fundo Social Europeu

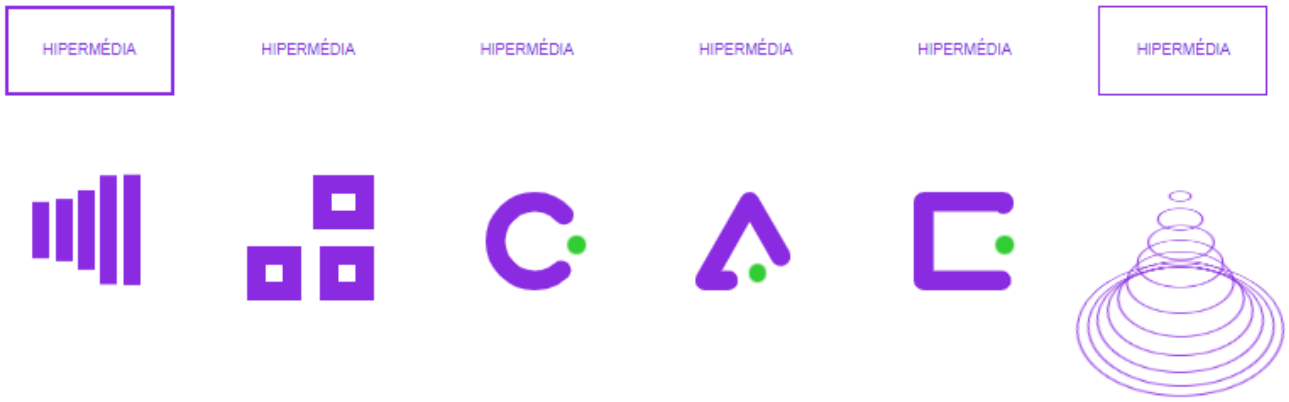
51. A animação das formas faz variar o offset do traço:

```
index.html > html > body > style > @keyframes pathTriangle
105             animation: dotTriangle 3s cubic-bezier(.785, .135, .15, .86) infinite;
106         }
107         @keyframes pathTriangle{
108             33%{
109                 stroke-dashoffset: 74;
110             }
111             66%{
112                 stroke-dashoffset: 147;
113             }
114             100%{
115                 stroke-dashoffset: 221;
116             }
117         }
118         @keyframes pathRect{
119             25%{
120                 stroke-dashoffset: 64;
121             }
122             50%{
123                 stroke-dashoffset: 128;
124             }
125             75%{
126                 stroke-dashoffset: 192;
127             }
128             100%{
129                 stroke-dashoffset: 256;
130             }
131         }
132         @keyframes pathCircle{
133             25%{
134                 stroke-dashoffset: 125;
135             }
136             50%{
137                 stroke-dashoffset: 175;
138             }
139             75%{
140                 stroke-dashoffset: 225;
141             }
142             100%{
143                 stroke-dashoffset: 275;
144             }
145         }
```

52. A animação do ponto já altera a sua posição em relação à forma:

```
146         @keyframes dotRect{
147             25%{
148                 transform: translate(0, 0);
149             }
150             50%{
151                 transform: translate(36px, -36px);
152             }
153             75%{
154                 transform: translate(0, -72px);
155             }
156             100%{
157                 transform: translate(-36px, -36px);
158             }
159         }
160         @keyframes dotTriangle{
161             33%{
162                 transform: translate(0, 0);
163             }
164             66%{
165                 transform: translate(20px, -36px);
166             }
167             100%{
168                 transform: translate(-20px, -36px);
169             }
170         }
```

53. Por último vamos criar uma animação que explore as propriedades de 3d do CSS



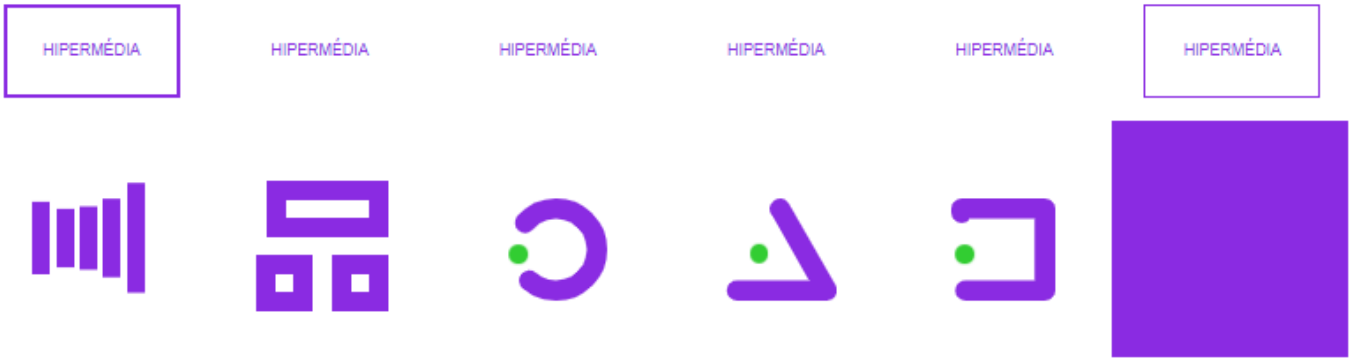
54. O markup consiste apenas de dez tags obtidos com **.loader6>span*6**

```
index.html > html > body > div.loaders > div.loader6
        </div>
        <div class="loader6">
            <span></span>
            <span></span>
            <span></span>
            <span></span>
            <span></span>
            <span></span>
            <span></span>
            <span></span>
            <span></span>
            <span></span>
        </div>
    </div>
```

55. Começamos por definir as dimensões para este sexto loader

```
index.html > html > body > style > .loader6
    </div>
    <style>
        .loader6 {
            position: relative;
            width: 200px;
            height: 200px;
            background-color: blueviolet;
        }
```

56. Kj



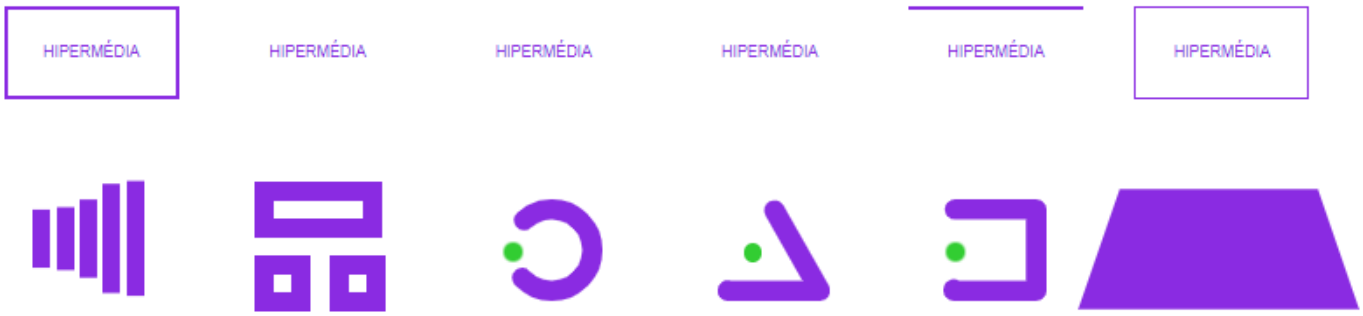
Cofinanciado por:

MADEIRA 14-20 · REGIÃO AUTÓNOMA DA MADEIRA · PORTUGAL 2020 · UNIÃO EUROPEIA Fundo Social Europeu

57. Colocamos o elemento em perspetiva 3D

```
.loader6 {
    position: relative;
    width: 200px;
    height: 200px;
    background-color: blueviolet;
    transform-style: preserve-3d;
    transform: perspective(500px) rotateX(60deg);
}
```

58. Neste momento, este elemento e todos os nele contidos (os 10 spans) ficam nesta orientação 3D



59. Retiramos o fundo do loader e formatamos todos o com uma linha de 2px:

index.html > html > body > style > .loader6 span

```
<style>
    .loader6 {
        position: relative;
        width: 200px;
        height: 200px;
        transform-style: preserve-3d;
        transform: perspective(500px) rotateX(60deg);
    }
    .loader6 span {
        position: absolute;
        display: block;
        border: 2px solid blueviolet;
        border-radius: 50%;
        transform: translateZ(-100px);
    }
```

60. Agora **PARA CADA** alteramos as dimensões a cada 10px com o seletor nth-child. Complete para os 10:

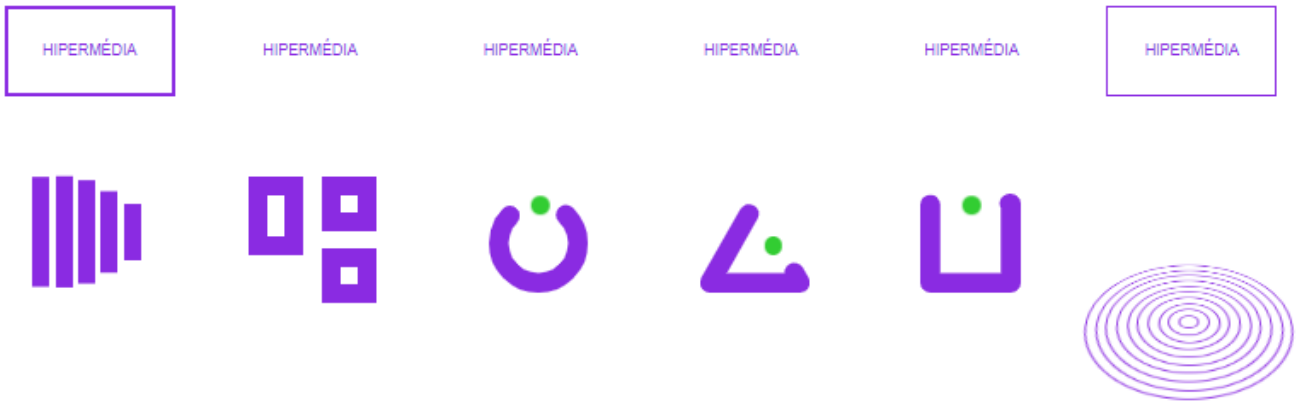
index.html > html > body > style > .loader6 span:nth-child(1)

```
.loader6 span {
    position: absolute;
    display: block;
    border: 2px solid blueviolet;
    border-radius: 50%;
    transform: translateZ(-100px);
}
.loader6 span:nth-child(1) {
    top: 0;
    left: 0;
    right: 0;
    bottom: 0;
}
.loader6 span:nth-child(2) {
    top: 10px;
    left: 10px;
    right: 10px;
    bottom: 10px;
}
.loader6 span:nth-child(3) {
    top: 20px;
```

Cofinanciado por:

MADEIRA 14-20 REGIÃO AUTÓNOMA DA MADEIRA PORTUGAL 2020 UNIÃO EUROPEIA Fundo Social Europeu

61. Deverá ter obtido:



62. Insira agora a animação dos . Neste momento os círculos movem-se todos ao mesmo tempo:

index.html > html > body > style > .loader6 span

```
.loader6 span {
    position: absolute;
    display: block;
    border: 2px solid blueviolet;
    border-radius: 50%;
    transform: translateZ(-100px);
    animation: animate6 3s ease-in-out infinite;
}
@keyframes animate6 {
    0%,
    100% {
        transform: translateZ(-100px);
    }
    50% {
        transform: translateZ(100px);
    }
}
.loader6 span:nth-child(1) {
    top: 0;
```

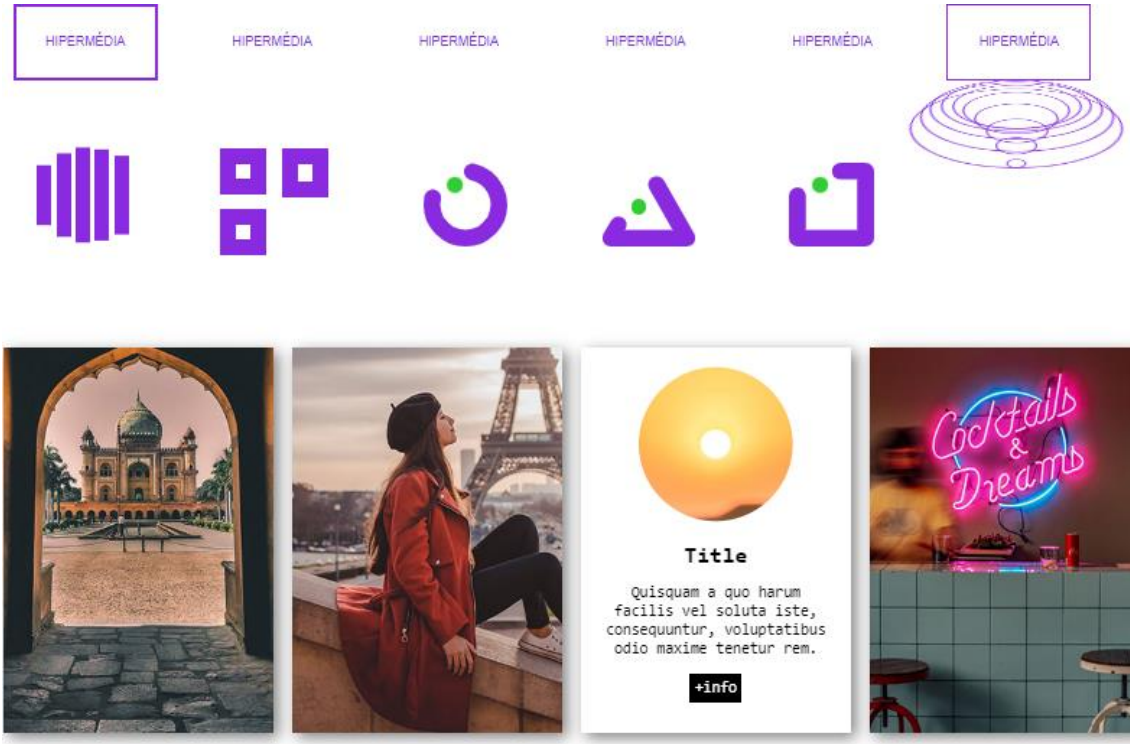
63. Tal como em animações anteriores, usamos um animation-delay de 100ms entre os

```
.loader6 span:nth-child(1) {
    top: 0;
    left: 0;
    right: 0;
    bottom: 0;
    animation-delay: 1.4s;
}
.loader6 span:nth-child(2) {
    top: 10px;
    left: 10px;
    right: 10px;
    bottom: 10px;
    animation-delay: 1.3s;
}
.loader6 span:nth-child(3) {…
}
.loader6 span:nth-child(4) {…
}
.loader6 span:nth-child(5) {…
}
.loader6 span:nth-child(6) {…
}
.loader6 span:nth-child(7) {…
}
.loader6 span:nth-child(8) {…
}
.loader6 span:nth-child(9) {…
}
.loader6 span:nth-child(10) {
    top: 90px;
    left: 90px;
    right: 90px;
    bottom: 90px;
    animation-delay: 0.5s;
}
```

Cofinanciado por:

MADEIRA 14-20 · REGIÃO AUTÓNOMA DA MADEIRA · PORTUGAL 2020 · UNIÃO EUROPEIA Fundo Social Europeu

64. Pretende-se agora definir animações para cartões com informação



65. Pesquise 4 imagens na internet para aplicar nos cartões, de preferência com a mesma dimensão. O markup gerado será **.cards1>.box*4>(.imgBx>img[src="img$.jpg"])+.content>(h2>{Title})+(p>lorem14)+a>{+info}**

index.html > html > body > div.cards1

```
16  <div class="loaders">…
56  </div>
57  <div class="cards1">
58      <div class="box">
59          <div class="imgBx">
60              <img src="img1.jpg">
61          </div>
62          <div class="content">
63              <h2>Title</h2>
64              <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod
65              tempor incididunt ut labore et dolore magna aliqua.</p>
66              <a href="#">+ INFO</a>
67          </div>
68      </div>
69      <div class="box">
70          <div class="imgBx">
71              <img src="img2.jpg">
72          </div>
73          <div class="content">
74              <h2>Title</h2>
75              <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod
76              tempor incididunt ut labore et dolore magna aliqua.</p>
77              <a href="#">+ INFO</a>
78          </div>
79      </div>
80      <div class="box">
81          <div class="imgBx">
82              <img src="img3.jpg">
83          </div>
84          <div class="content">
85              <h2>Title</h2>
86              <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod
87              tempor incididunt ut labore et dolore magna aliqua.</p>
88              <a href="#">+ INFO</a>
89          </div>
90      </div>
91      <div class="box">
92          <div class="imgBx">
93              <img src="img4.jpg">
94          </div>
95          <div class="content">
96              <h2>Title</h2>
97              <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod
98              tempor incididunt ut labore et dolore magna aliqua.</p>
99              <a href="#">Read More</a>
100         </div>
101     </div>
102 </div>
```

Cofinanciado por:

MADEIRA 14-20 · REGIÃO AUTÓNOMA DA MADEIRA · PORTUGAL 2020 · UNIÃO EUROPEIA Fundo Social Europeu

66. Criamos uma nova linha com flexbox para alinhar os cartões e definimos as dimensões iniciais:

```
index.html > html > body > style > .cards1
103     <style>
104         .cards1 {
105             position: relative;
106             width: 1200px;
107             display: flex;
108             justify-content: space-around;
109             margin: 20px 0;
110             font-family: consolas;
111         }
112         .cards1 .box {
113             position: relative;
114             width: 280px;
115             height: 400px;
116             margin: 20px 0;
117             box-sizing: border-box;
118             box-shadow: 5px 5px 15px rgba(0,0,0,0.5);
119             overflow: hidden;
120         }
```

67. Neste momento já deverá ter os cartões nesta disposição:

68. Queremos agora que a imagem fique visível apenas num círculo acima do texto. Para definimos essa "máscara" usamos o clip-path, que depois animamos com uma transition:

```
index.html > html > body > style > .cards1 .box:hover .imgBx
119             overflow: hidden;
120         }
121         .cards1 .box .imgBx{
122             position: absolute;
123             top: 0;
124             left: 0;
125             width: 100%;
126             height: 100%;
127             background: #000;
128             clip-path: circle(400px at center 100px);
129             transition: 0.5s;
130             transition-delay: 0.5s;
131         }
132         .cards1 .box:hover .imgBx{
133             clip-path: circle(80px at center 100px);
134             transition-delay: 0s;
135         }
```

69. Neste momento já ao passar com o rato por cima da imagem esta muda as dimensões até ficar num círculo:

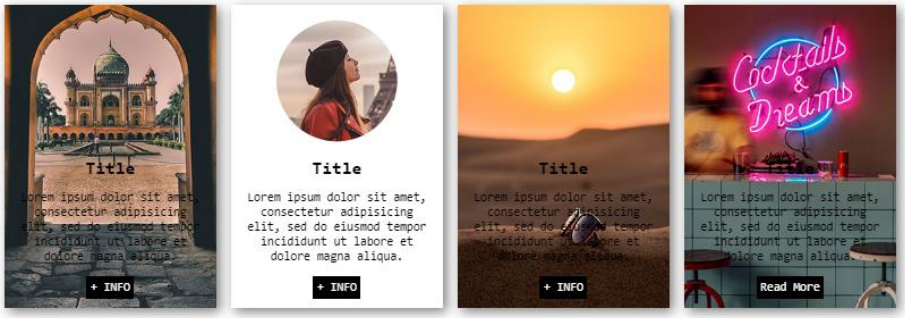
Cofinanciado por:

MADEIRA 14-20
REGIÃO AUTÓNOMA DA MADEIRA
PORTUGAL 2020
UNIÃO EUROPEIA
Fundo Social Europeu

70. Vamos agora posicionar os restantes elementos:

```
index.html > html > body > style > .cards1 .box .content
134             transition-delay: 0s;
135         }
136         .cards1 .box .content{
137             position: absolute;
138             left: 0;
139             bottom: 0;
140             width: 100%;
141             height: 55%;
142             padding: 20px;
143             box-sizing: border-box;
144             text-align: center;
145         }
146         .cards1 .box .content h2{
147             margin: 0;
148             padding: 0;
149         }
150         .cards1 .box .content a{
151             text-decoration: none;
152             background: #000;
153             color: #fff;
154             padding: 5px;
155             display: inline-block;
156         }
157         .cards1 .box .imgBx img{
158             position: absolute;
159             top: 0;
160             left: 0;
161             width: 100%;
162             height: 100%;
163             object-fit: cover;
164         }
```

71. Apesar dos elementos já estarem no sítio certo, ficam ainda sobrepostos



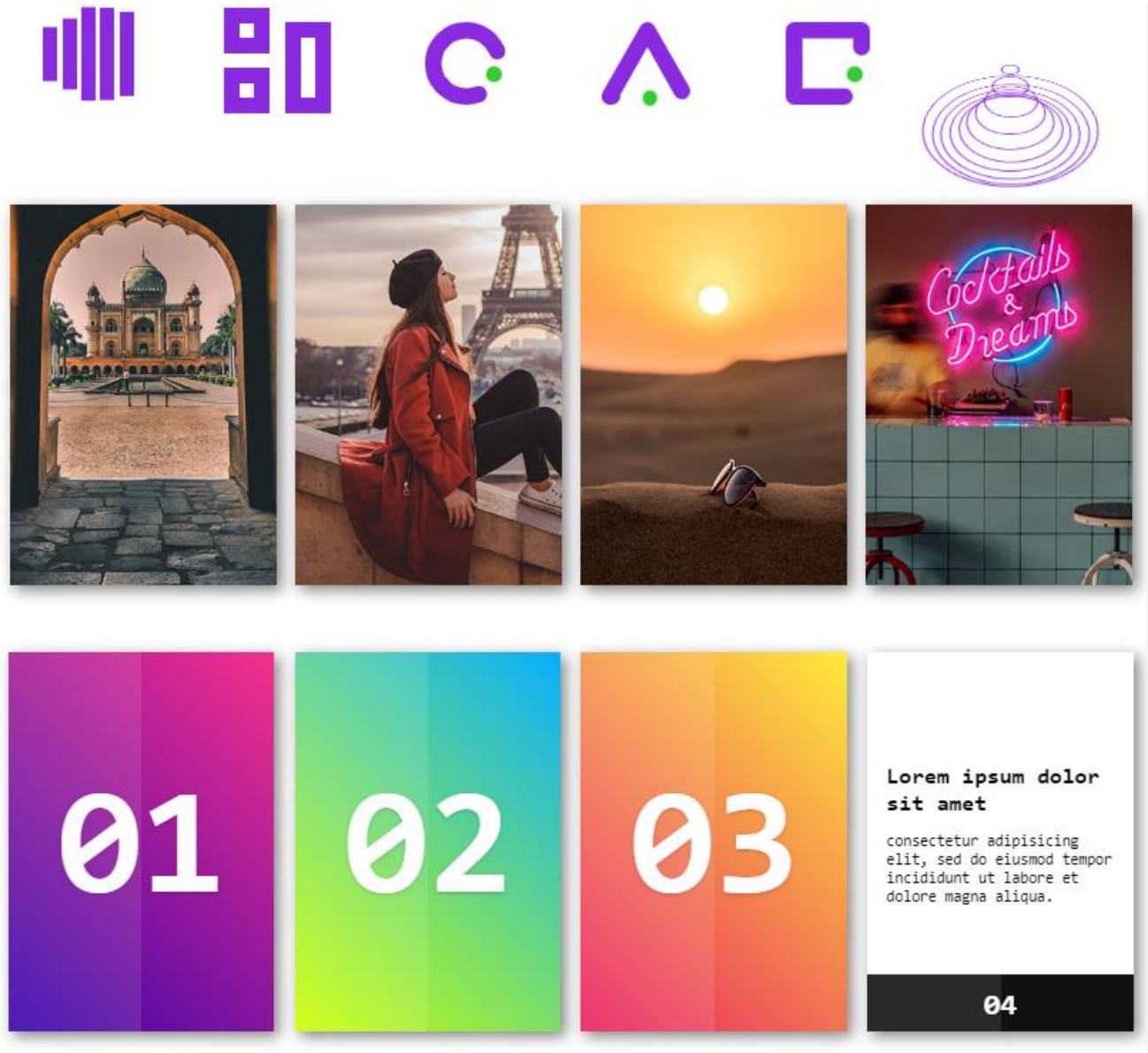
72. Em vez só alterarmos o z-index como antes, vamos aproveitar para também animar a entrada dos textos

```
index.html > html > body > style > .cards1 .box:hover .content a
165         .cards1 .box .content h2,
166         .cards1 .box .content p,
167         .cards1 .box .content a{
168             opacity: 0;
169             transition: 0.5s;
170             transform: translateY(20px);
171         }
172         .cards1 .box:hover .content h2{
173             opacity: 1;
174             transform: translateY(0);
175             transition-delay: 0.5s
176         }
177         .cards1 .box:hover .content p{
178             opacity: 1;
179             transform: translateY(0);
180             transition-delay: 0.7s
181         }
182         .cards1 .box:hover .content a{
183             opacity: 1;
184             transform: translateY(0);
185             transition-delay: 0.9s
186         }
```

Cofinanciado por:

MADEIRA 14-20 · REGIÃO AUTÓNOMA DA MADEIRA · PORTUGAL 2020 · UNIÃO EUROPEIA Fundo Social Europeu

73. Neste novo conjunto vamos utilizar gradientes em vez de imagens nos cartões:



74. Vamos utilizar uma grid para dispor os cartões:

```
index.html > html > body > style > .cards2
        </div>
    <style>
        .cards2{
            width: 1200px;
            display: grid;
            grid-template-columns: repeat(auto-fit, minmax(280px, 1fr));
            grid-gap: 5px;
            margin: 50px 0;
            font-family: consolas;
        }
        .cards2 .box2{
            position: relative;
            width: 280px;
            height: 400px;
            margin: 0 auto;
            background: #fff;
            box-shadow: 5px 5px 15px rgba(0,0,0,.5);
        }
```

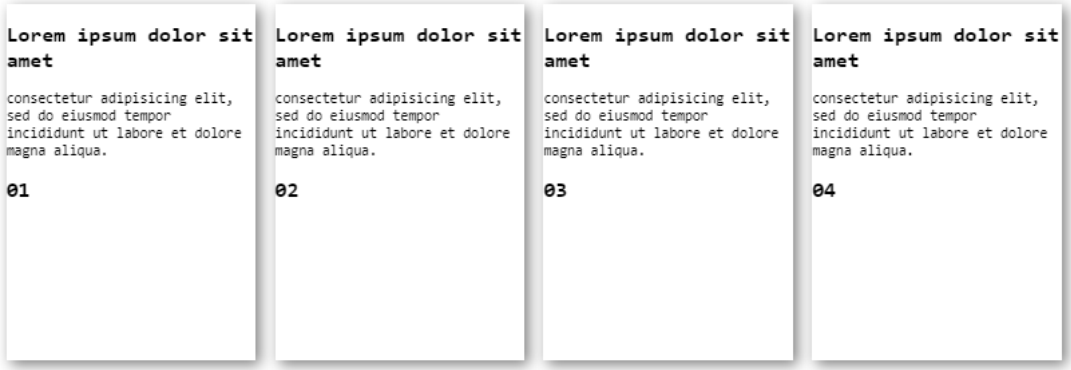
Cofinanciado por:

MADEIRA 14-20 · REGIÃO AUTÓNOMA DA MADEIRA · PORTUGAL 2020 · UNIÃO EUROPEIA Fundo Social Europeu

75. O markup é semelhante ao exemplo anterior e pode ser gerado com emmet da mesma forma:

```
index.html > html > body > div.cards2
<div class="cards2">
    <div class="box2">
        <div class="face face1">
            <div class="content">
                <h2>Lorem ipsum dolor sit amet</h2>
                <p>consectetur adipisicing elit, sed do eiusmod
                tempor incididunt ut labore et dolore magna aliqua.</p>
            </div>
        </div>
        <div class="face face2">
            <h2>01</h2>
        </div>
    </div>
    <div class="box2">
        <div class="face face1">
            <div class="content">
                <h2>Lorem ipsum dolor sit amet</h2>
                <p>consectetur adipisicing elit, sed do eiusmod
                tempor incididunt ut labore et dolore magna aliqua.</p>
            </div>
        </div>
        <div class="face face2">
            <h2>02</h2>
        </div>
    </div>
    <div class="box2">
        <div class="face face1">
            <div class="content">
                <h2>Lorem ipsum dolor sit amet</h2>
                <p>consectetur adipisicing elit, sed do eiusmod
                tempor incididunt ut labore et dolore magna aliqua.</p>
            </div>
        </div>
        <div class="face face2">
            <h2>03</h2>
        </div>
    </div>
    <div class="box2">
        <div class="face face1">
            <div class="content">
                <h2>Lorem ipsum dolor sit amet</h2>
                <p>consectetur adipisicing elit, sed do eiusmod
                tempor incididunt ut labore et dolore magna aliqua.</p>
            </div>
        </div>
        <div class="face face2">
            <h2>04</h2>
        </div>
    </div>
</div>
<style>
```

76. Neste momento deverá obter:



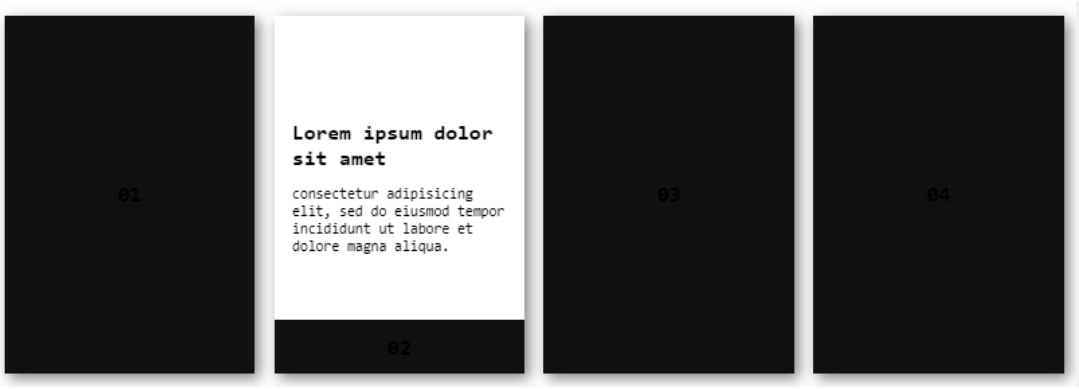
Cofinanciado por:

MADEIRA 14-20 REGIÃO AUTÓNOMA DA MADEIRA PORTUGAL 2020 UNIÃO EUROPEIA Fundo Social Europeu

77. Começamos por formatar o conteúdo em face1 e para o face2 animamos uma transição da altura:

```
index.html > html > body > style > .cards2 .box2 .face
226         .cards2 .box2 .face{
227             position: absolute;
228             bottom: 0;
229             left: 0;
230             width: 100%;
231             height: 100%;
232             display: flex;
233             justify-content: center;
234             align-items: center;
235         }
236         .cards2 .box2 .face.face1{
237             box-sizing: border-box;
238             padding: 20px;
239         }
240         .cards2 .box2 .face.face1 h2{
241             margin: 0;
242             padding: 0;
243         }
244         .cards2 .box2 .face.face2{
245             background: #111;
246             transition: 0.5s;
247         }
248         .cards2 .box2:hover .face.face2{
249             height: 60px;
250         }
```

78. Devemos obter:



79. Aplicamos os gradientes às face2

```
index.html > html > body > style > .cards2 .box2:nth-child(1) .face.face2
193             height: 60px;
194         }
195         .cards2 .box2:nth-child(1) .face.face2{
196             background: linear-gradient(45deg, #3503ad, #f7308c);
197         }
198         .cards2 .box2:nth-child(2) .face.face2{
199             background: linear-gradient(45deg, #ccff00, #09afff);
200         }
201         .cards2 .box2:nth-child(3) .face.face2{
202             background: linear-gradient(45deg, #e91e63, #ffeb3b);
203         }
204         .cards2 .box2:nth-child(4) .face.face2{
205             background: linear-gradient(45deg, #000, #444);
206         }
```

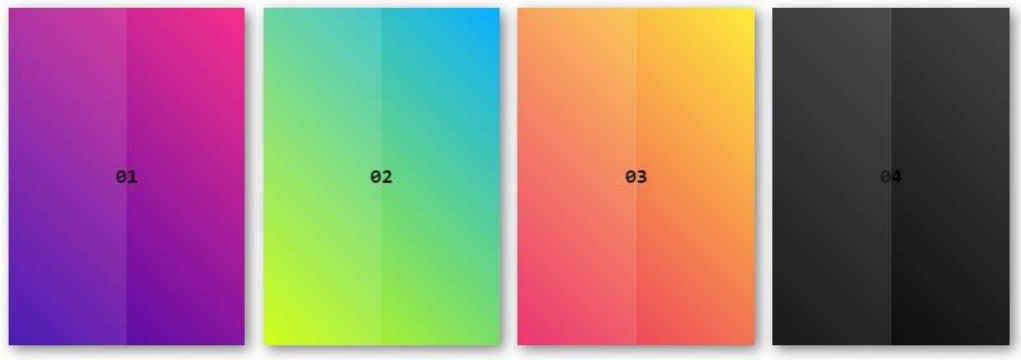
Cofinanciado por:

MADEIRA 14-20 REGIÃO AUTÓNOMA DA MADEIRA PORTUGAL 2020 UNIÃO EUROPEIA Fundo Social Europeu

80. Para quebrar um pouco a predominância do gradiente vamos adicionar um efeito de reflexão

```
.cards2 .box2 .face.face2::before{
    content: '';
    position: absolute;
    top: 0;
    left: 0;
    width: 50%;
    height: 100%;
    background: rgba(255,255,255,.1);
}
```

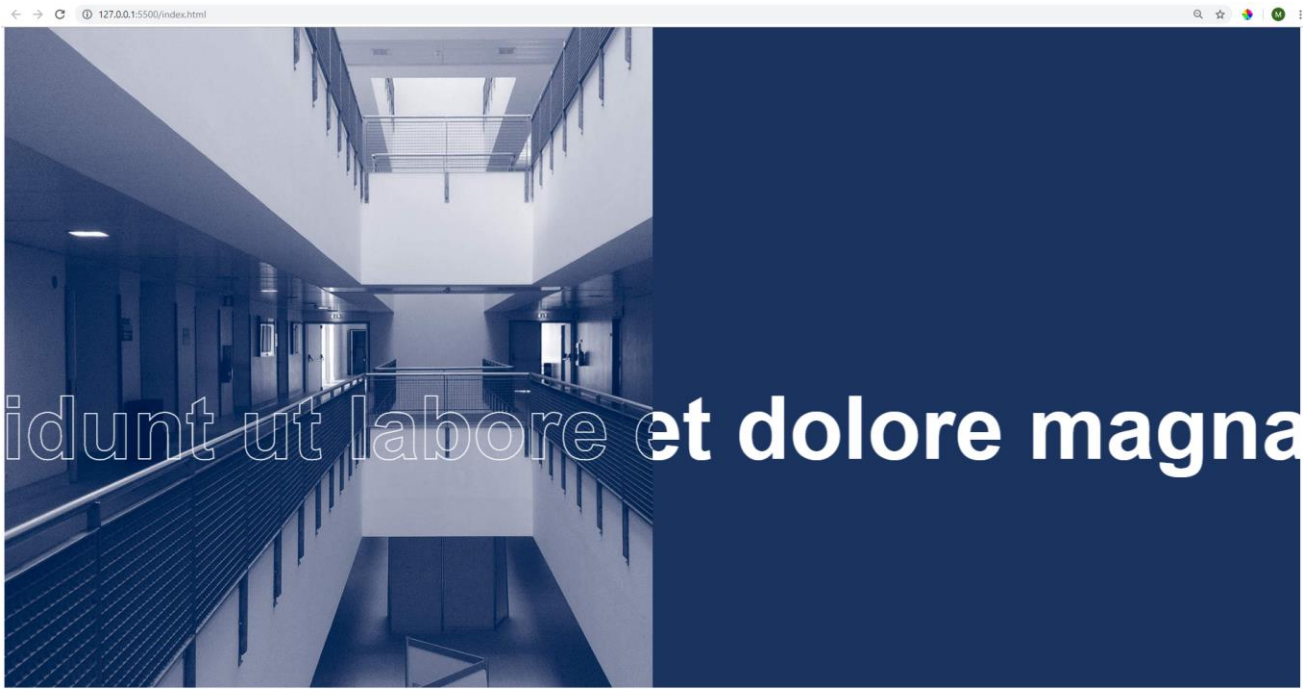
81. Terá obtido:



82. Concluímos com a formatação do número que fica sobre os gradientes

```
.cards2 .box2 .face.face2 h2{
    margin: 0;
    padding: 0;
    font-size: 10em;
    color: #fff;
    transition: 0.5s;
    text-shadow: 0 2px 5px rgba(0,0,0,.2);
}
.cards2 .box2:hover .face.face2 h2{
    font-size: 2em;
}
```

83. Neste exemplo será efetuada a animação de um texto num header:



Cofinanciado por:

MADEIRA 14-20
REGIÃO AUTÓNOMA DA MADEIRA
PORTUGAL 2020
UNIÃO EUROPEIA
Fundo Social Europeu

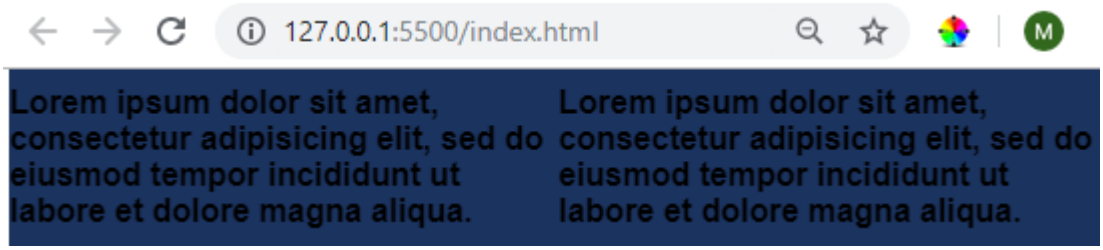
84. No markup temos de repetir o mesmo texto duas vezes:

```
index.html > html > body > div.header1
153         <div class="header1">
154             <div class="com-imagem">
155                 <h2>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod
156                 tempor incididunt ut labore et dolore magna aliqua.</h2>
157             </div>
158             <div class="sem-imagem">
159                 <h2>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod
160                 tempor incididunt ut labore et dolore magna aliqua.</h2>
161             </div>
162         </div>
163     <style>
```

85. Começamos por formatar o header para ocupar toda a viewport do browser usando uma flexbox

```
index.html > html > body > style > .header1
163     <style>
164         .header1{
165             position: relative;
166             font-family: sans-serif;
167             width: 100%;
168             height: 100vh;
169             display: flex;
170             overflow: hidden;
171         }
172         .header1 .com-imagem,
173         .header1 .sem-imagem{
174             position: relative;
175             width: 50%;
176             background: #1B335F;
177             overflow: hidden;
178         }
```

86. Neste momento os textos terão esta disposição



87. Colocamos agora uma imagem para ocupar metade do viewport no div com a classe .com-imagem.

```
179         .header1 .com-imagem{
180             background: url(uma.jpg);
181             background-size: cover;
182         }
```

88. Como usamos o cover no background-size esta imagem vai se adaptar, mas convém que selecione uma imagem com uma dimensão mais quadrada ou ao alto

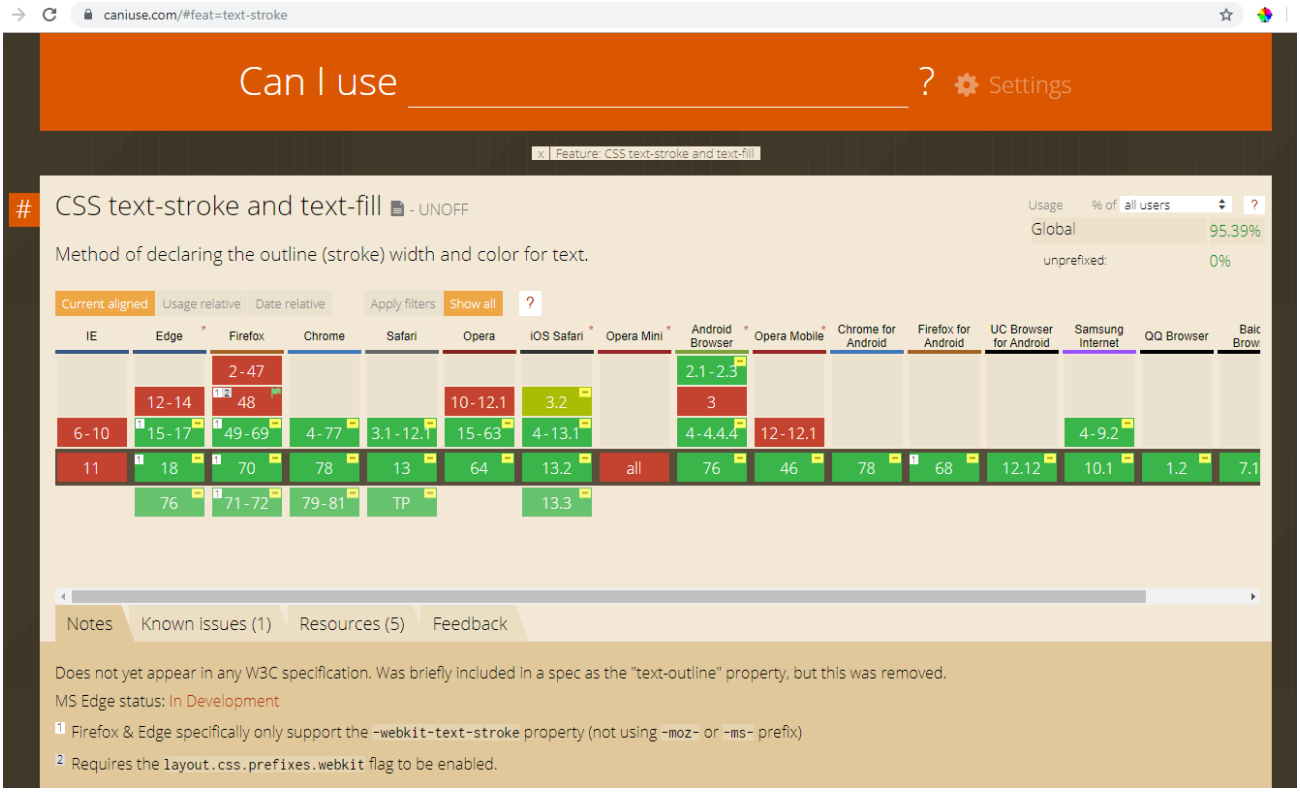
Cofinanciado por:

MADEIRA 14-20 · REGIÃO AUTÓNOMA DA MADEIRA · PORTUGAL 2020 · UNIÃO EUROPEIA · Fundo Social Europeu

89. Queremos que o texto fique sem quebras, numa linha contínua, e usamos para isso o white-space: nowrap;

```
.header1 .com-imagem h2,
.header1 .sem-imagem h2{
    position: absolute;
    line-height: 100vh;
    white-space: nowrap;
    font-size: 12em;
}
```

90. Para criar um contraste entre os dois textos colocamos o que está sobre a imagem apenas com o contorno. Para tal usamos o texto-stroke que ainda não está implementado em todos os browsers, pelo que usamos o prefixo -webkit:



91. Falta agora animar ambos os textos de forma a parecer que é o mesmo:

```
.header1 .com-imagem h2{
    color: transparent;
    -webkit-text-stroke: 2px;
    -webkit-text-stroke-color: #fff;
    left: 100%;
    animation: animate-text 60s linear infinite;
}
.header1 .sem-imagem h2{
    color: #fff;
    animation: animate-text 60s linear infinite;
}
@keyframes animate-text{
    0%{
        transform: translateX(0);
    }
    100%{
        transform: translateX(-100%);
    }
}
```

Cofinanciado por:
MADEIRA 14-20
REGIÃO AUTÓNOMA DA MADEIRA
PORTUGAL 2020
UNIÃO EUROPEIA
Fundo Social Europeu