

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

Unidade Curricular: Design Hipermedia

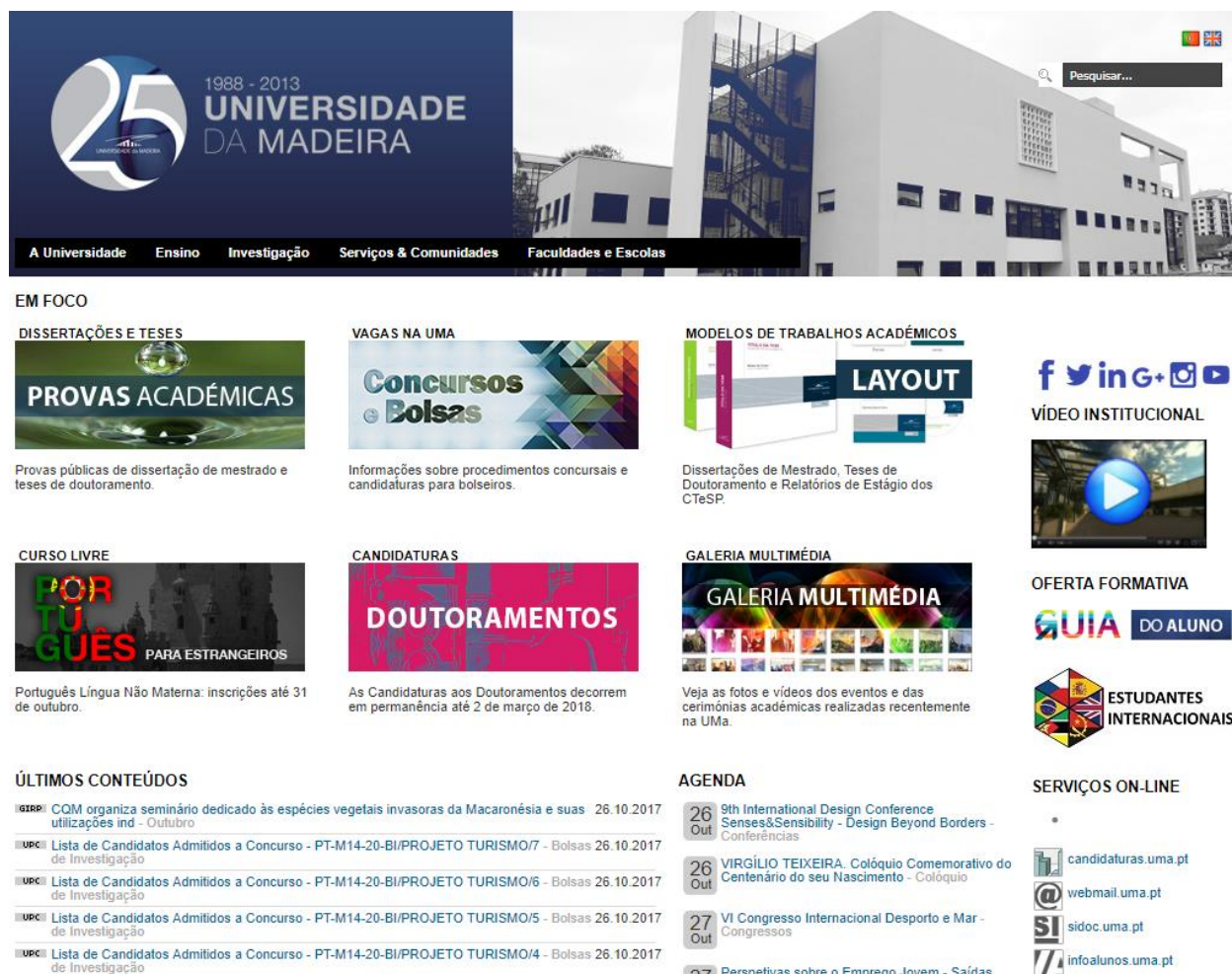
1º Ano/1º Semestre

Docente: Marco Miguel Olival Olim

Data 02/11/2017

ESTE EXERCÍCIO PRETENDE APLICAR OS PRINCÍPIOS DE RESPONSIVE DESIGN COM GRID

1- Pretende-se replicar o website da Universidade da Madeira num layout baseado em Grid:



Cofinanciado por:

2- Iniciar um ficheiro de html com a seguinte estrutura:

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <link rel="stylesheet" href="grid-site.css">
  <title>UMa</title>
</head>

<body>
  <div class="gridwrapper">
    <header class="gridItems"></header>
    <nav class="gridItems"></nav>
    <section class="gridItems foco">em foco</section>
    <section class="gridItems ultimas">Últimas</section>
    <section class="gridItems agenda">agenda</section>
    <section class="gridItems links">links</section>
    <footer class="gridItems"></footer>
  </div>
</body>

</html>
```

3- Crie o ficheiro CSS e inicialize com os seguintes estilos

```
body {
  background-color: #eee;
  font-family: Arial, Helvetica, sans-serif;
  margin: 0;
  padding: 0;
}
```

4- Aplicamos Grid ao site e atribuímos 100vh de altura e 100% de largura na classe gridwrapper, possibilitando assim que o site se ajuste às dimensões do browser, atendendo a que as unidades utilizadas para as colunas são frações (fr), tornando esta uma solução responsiva. É também definido como *auto* a secção dos conteúdos nas linhas do Grid.

```
.gridwrapper {
  height: 100vh;
  width: 100%;
  display: grid;
  grid-gap: 1px;
  grid-template-columns: 2fr 2fr 1fr;
  grid-template-rows: 10rem 2.75rem 10rem auto 2.75rem;
}

.gridItems {
  background-color: white;
}
```

Cofinanciado por:



- 5- Pode agora optar por definir os conteúdos da mesma forma que o exercício anterior utilizando também o `grid-column-start`, `grid-column-end`, `grid-row-start` e `grid-row-end` ou a abreviação correspondente

```
header {  
  grid-column: 1 / span 3;  
  grid-row: 1 / 2;  
}
```

- 6- Outro processo para atribuir conteúdos à Grid é a utilização de Áreas, onde definimos uma template de colocação dos conteúdos

```
.gridwrapper {  
  height: 100vh;  
  width: 100%;  
  display: grid;  
  grid-gap: 1px;  
  grid-template-columns: 2fr 2fr 1fr;  
  grid-template-rows: 10rem 2.75rem 10rem auto 2.75rem;  
  grid-template-areas: "header header header"  
                      "nav nav nav"  
                      "foco foco links"  
                      "ultimas agenda links"  
                      "footer footer footer";  
}
```

- 7- Depois de definida a template, são indicadas nas classes correspondentes a qual área pertencem

```
header {  
  grid-area: header;  
}
```

- 8- Falta agora preencher a informação de cada uma das secções. Neste caso temos os dados para o rodapé

```
<footer class="gridItems">  
  <p>Colégio dos Jesuítas - Rua dos Ferreiros</p>  
  <p>9000-082 Funchal - Portugal</p>  
  <p>Telef. + 351 291 209 400 | Fax + 351 291 209 410</p>  
  <p>email: gabinetedareitoria@uma.pt</p>  
</footer>
```

- 9- Aplicamos então os estilos correspondentes. Foi utilizado “space-between” à semelhança do Flexbox para calcular automaticamente o espaçamento entre parágrafos.

```
footer {  
  grid-area: footer;  
  display: grid;  
  grid-auto-flow: column;  
  align-content: space-between;  
}
```

Cofinanciado por:



- 10- O conteúdo em si carece de ser mais expandido. Identificaram-se as seguintes secções no atual site, necessitando ainda de serem replicadas do site original.

```
nav {
  grid-area: nav;
  display: grid;
  grid-auto-flow: column;
  align-content: space-evenly;
}

.foco {
  grid-area: foco;
}

.ultimas {
  grid-area: ultimas;
}

.agenda {
  grid-area: agenda;
}

.links {
  grid-area: links;
}
```

- 11- Como o Grid ainda não foi universalmente implementado com a mesma notação, podemos, no entanto, detetar se este é suportado no browser atual usando o feature query do CSS.

```
@supports (display: grid) {
```

- 12- Podemos agora propor uma solução para utilizar browsers mais antigos, aplicando só os nossos estilos correspondentes à Grid se esta for suportada pelo browser. Refira-se que tanto a Grid como o FlexBox fazem o override de floats, inline-block, etc. No entanto há sempre particularidades (como as larguras no nosso caso) que tornam necessário este recurso à feature query.

```
body {
  margin: 0;
  padding: 0;
  background-color: #eee;
}
```

Cofinanciado por:



```

.gridItems {
    float: left;
}

header {
    width: 100%;
    height: 10rem;
}

nav {
    width: 100%;
    height: 2.75rem;
}

nav .LastItem {
    float: right;
}

.foco {
    width: 100%;
    height: 10rem;
}

.agenda {
    width: 40%;
}

.ultimas {
    width: 40%;
}

.links {
    width: 20%;
}

footer {
    position: fixed;
    bottom: 0;
    left: 0;
    height: 2.75rem;
    width: 100%;
}

footer p {
    display: inline-block;
}

```

Cofinanciado por:



```

@supports (display: grid) {
  .gridwrapper {
    height: 100vh;
    width: 100%;
    display: grid;
    grid-gap: 1px;
    grid-template-columns: 2fr 2fr 1fr;
    grid-template-rows: 10rem 2.75rem 10rem auto 2.75rem;
    grid-template-areas: "header header header" "nav nav nav" "foco foco links"
"ultimas agenda links" "footer footer footer";
  }
  .gridItems {
    background-color: steelblue;
  }
  header {
    grid-area: header;
  }
  nav {
    grid-area: nav;
    display: grid;
    grid-auto-flow: column;
    align-content: space-evenly;
  }
  .LastItem {
    margin-left: auto;
  }
  footer {
    grid-area: footer;
    display: grid;
    grid-auto-flow: column;
    align-content: space-between;
    position: static;
  }
  .foco {
    grid-area: foco;
  }
  .ultimas {
    grid-area: ultimas;
    width: 100%;
  }
  .agenda {
    grid-area: agenda;
    width: 100%;
  }
  .links {
    grid-area: links;
    width: 100%;
  }
}

```

Cofinanciado por:



```
@supports (display: grid) {
  .gridwrapper {
    height: 100vh;
    width: 100%;
    display: grid;
    grid-gap: 1px;
    grid-template-columns: 2fr 2fr 1fr;
    grid-template-rows: 10rem 2.75rem 2.75rem 10rem auto 2.75rem;
  }
  .gridItems {
    background-color: steelblue;
  }
  header {
    height: unset;
    grid-column: 1 / 4;
    grid-row: 1 / 4;
    background-image: url('http://www.uma.pt/portal/imagens/UMaBanner25.jpg');
    background-size: cover;
  }
  nav {
    display: grid;
    grid-auto-flow: column;
    align-content: space-evenly;
    grid-column: 1 / 3;
    grid-row: 2 / 3;
    background-color: black;
    padding: 0 0.75rem 0 0.75rem;
  }
  .LastItem {
    margin-left: auto;
  }
  footer {
    display: grid;
    grid-auto-flow: column;
    align-content: space-between;
    position: static;
    grid-column: 1 / 4;
    grid-row: 6 / 7;
  }
  .foco {
    grid-column: 1 / 3;
    grid-row: 4 / 5;
  }
  .ultimas {
    width: 100%;
    grid-column: 1 / 2;
    grid-row: 5 / 6;
  }
  .agenda {
    width: 100%;
    grid-column: 2 / 3;
  }
}
```

Cofinanciado por:



```

    grid-row: 5 / 6;
}
.links {
    grid-column: 3 / 4;
    grid-row: 4 / 6;
    width: 100%;
}
}
}

```

14- Neste momento deverá ter obtido um resultado semelhante à figura abaixo apresentada:



15- Apresente agora uma solução para obter um resultado mais próximo do site da Universidade da Madeira. Note que é possível conjugar FlexBox com o Grid.

Cofinanciado por:

