



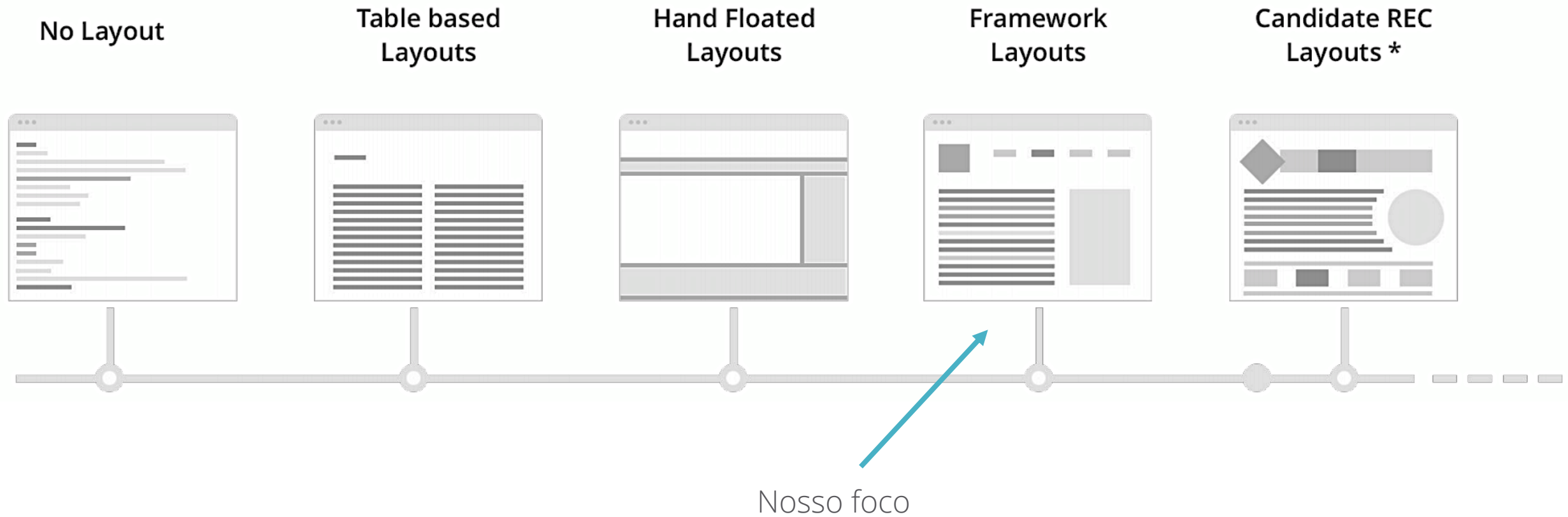
PUC Minas

Layout Flex e Grid – parte 1

Prof. Marcos Arrais



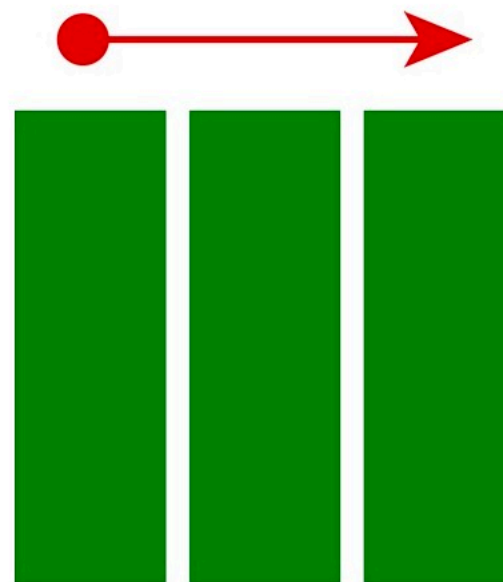
Um pouco de história



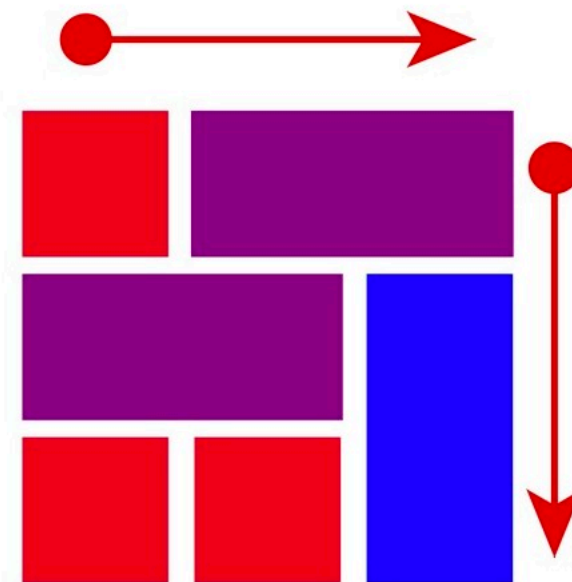
Fonte: <https://www.slideshare.net/valiac/css-grid-vs-flexbox>

Entendendo o Flex e o Grid

- São estruturas CSS que trabalham com a distribuição de elementos baseados em posição, tamanho e alinhamento;
- Hoje são a recomendação para construção de layouts responsivos;
- As duas técnicas podem trabalhar em conjunto;



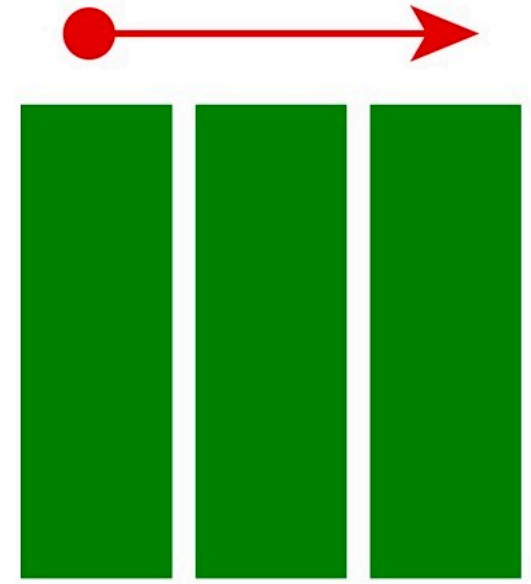
Flexbox
One Dimensions



CSS Grids
Two Dimensions

Layout Flex

- Mais indicado para alinhar componentes em um única dimensão, seja ela linha ou coluna;
- Não precisamos nos preocupar com a quantidade de itens pois os espaçamentos são calculados automaticamente;
- Não permite interação entre linha e coluna (por exemplo operações de mesclagem de divs);



Flexbox
One Dimensions

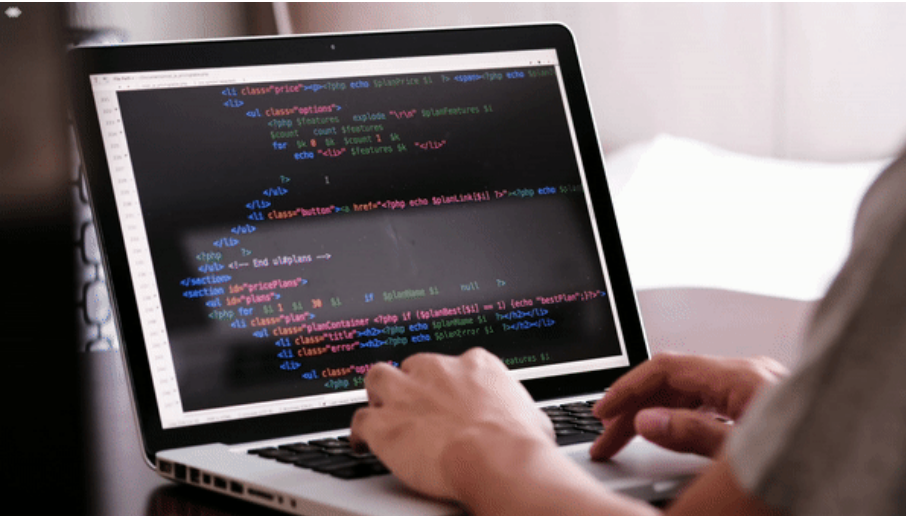
Layout Flex

```
.header {  
  background: #727272;  
  display: flex;  
  flex-wrap: wrap;  
  justify-content: space-between;  
  align-items: center;  
  padding: 20px;  
}
```

Exemplo simples:

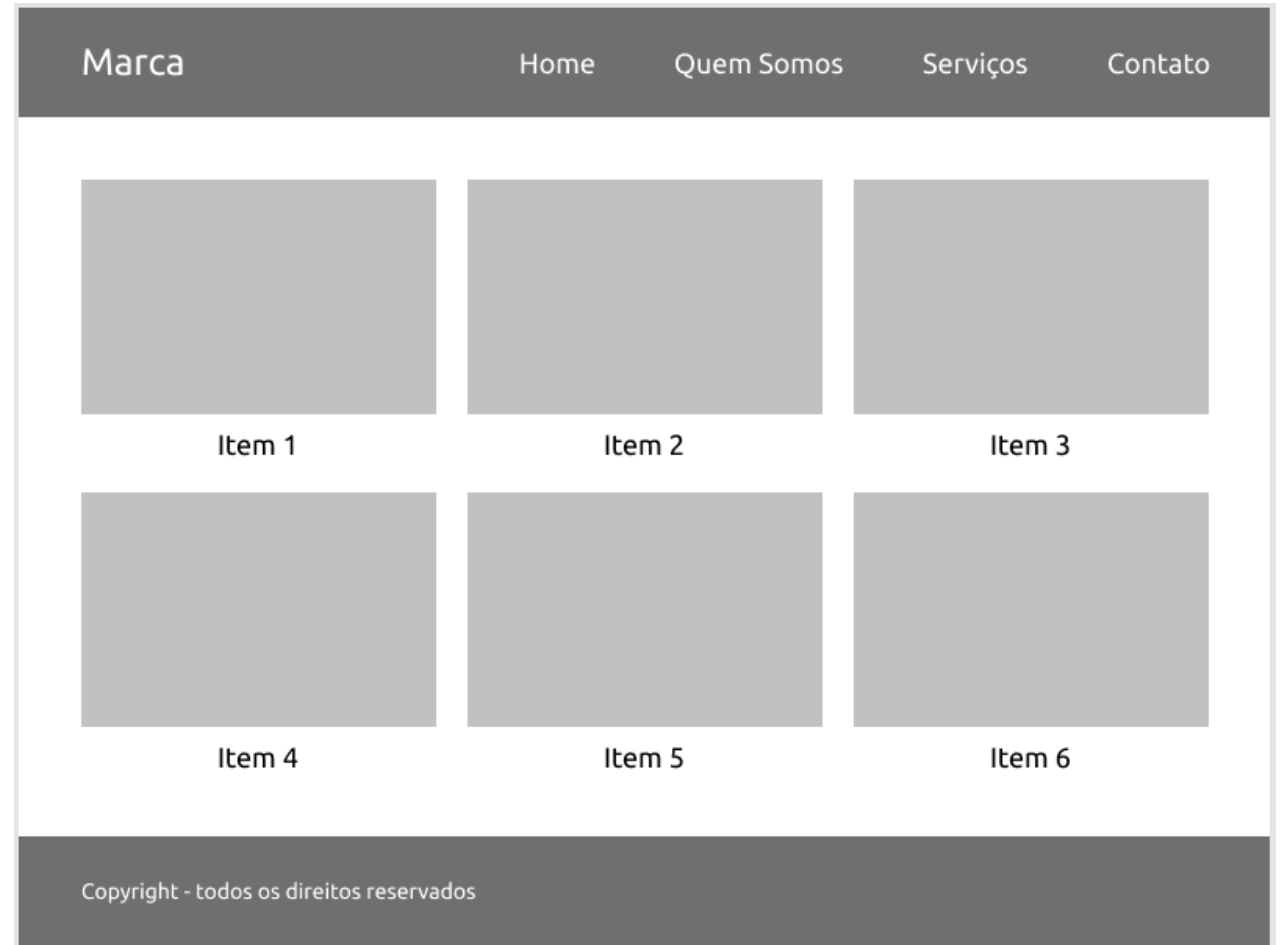
- Ativando o layout Flex;
- Definindo o que acontece quando não houver mais espaços para alocar os itens (no caso, quebramos a linha);
- Espaçamento entre os itens;
- Alinhamento;

Mão na massa



Codificar um exemplo de Layout Flex segundo o wireframe:

Fonte: Baseado nos exemplos de https://www.youtube.com/watch?v=x-4z_u8LcGc





PUC Minas

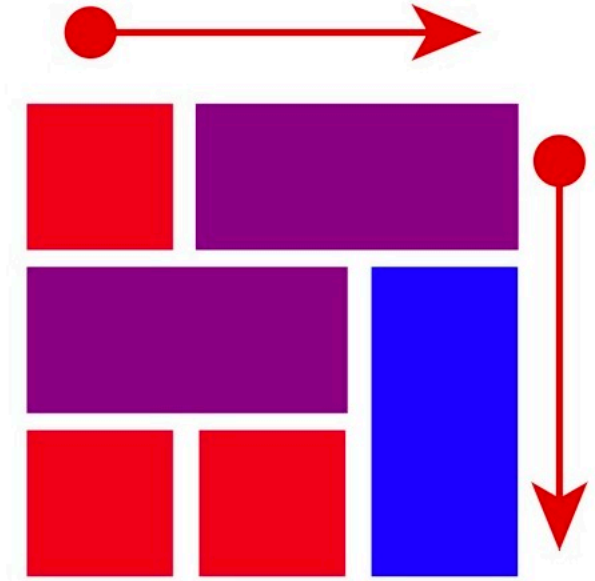
Layout Flex e Grid – parte 2

Prof. Marcos Arrais



Grid Layout

- Multidimensional;
- Indicado pra layouts mais complexos onde os elementos tem interação entre linha e coluna;
- É mais complexo de ser implementado pois é necessário explicitar o fracionamento das divisões (linha/coluna);



CSS Grids
Two Dimensions

Grid Layout

```
.grid1 {  
  display: grid;  
  grid-template-columns: repeat(3, 1fr);  
  max-width: 800px;  
  margin: 0 auto;  
  grid-gap: 20px;  
}
```

Exemplo simples:

- Ativando o Grid Layout;
- Definindo colunas e o fracionamento;
- Definindo o tamanho máximo da viewport que será usado;
- Margens;
- Espaçamento entre itens;

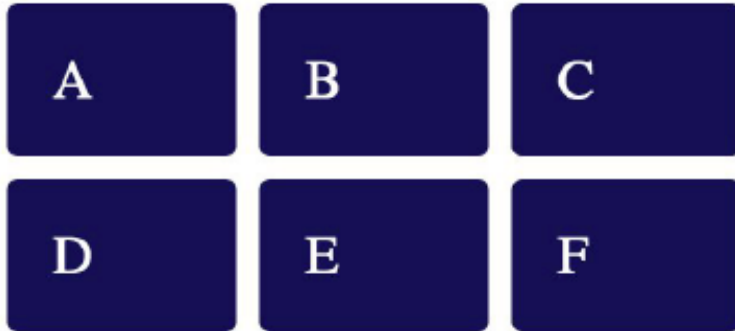
Grid Layout com Sobreposição

```
<div class="container">
  <div class="a">A</div>
  <div class="b">B</div>
  <div class="c">C</div>
  <div class="d">D</div>
  <div class="e">E</div>
  <div class="f">F</div>
</div>
```

Exemplo simples:

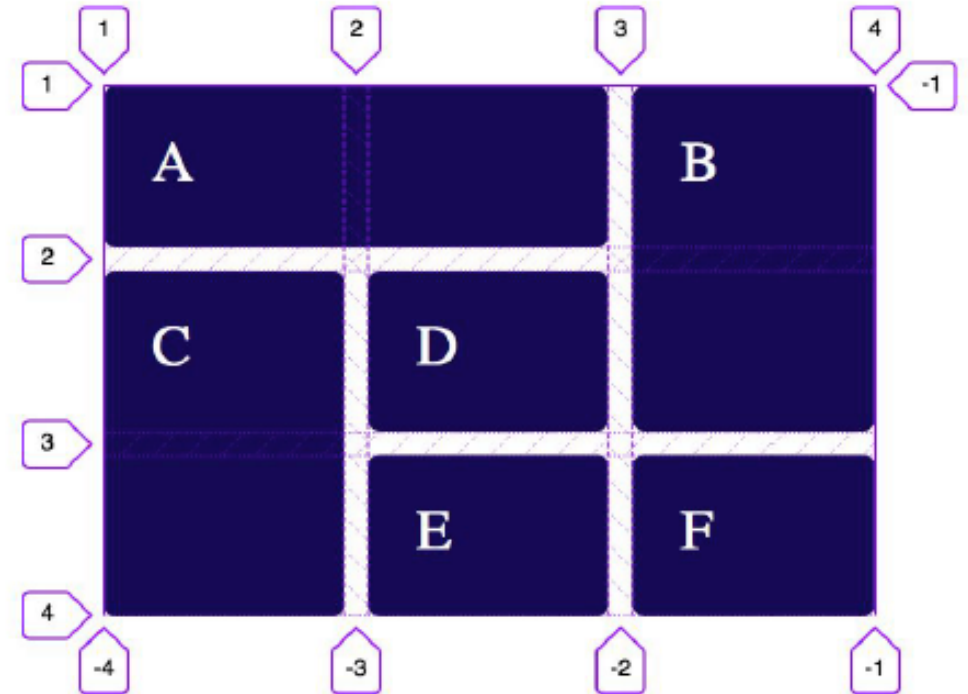
- Primeiro vamos criar um HTML simples.

Grid Layout com Sobreposição



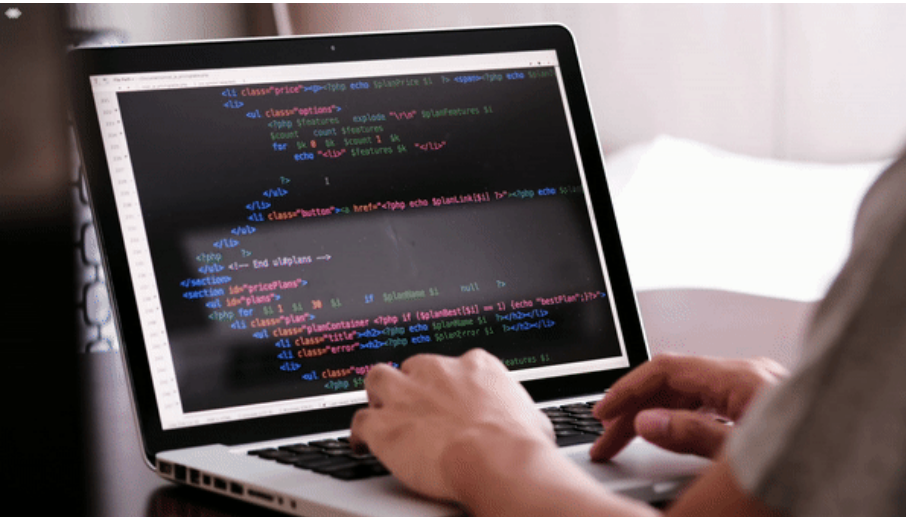
```
.a {  
}  
.b {  
}  
.c {  
}  
.container {  
  display: grid;  
  grid-gap: 10px;  
  grid-template-columns:  
    repeat(3, 100px);  
}
```

```
.a {  
  grid-column: 1 / 3;  
  grid-row: 1;  
}  
.b {  
  grid-column: 3;  
  grid-row: 1 / 3;  
}  
.c {  
  grid-column: 1;  
  grid-row: 2 / 4;  
}
```



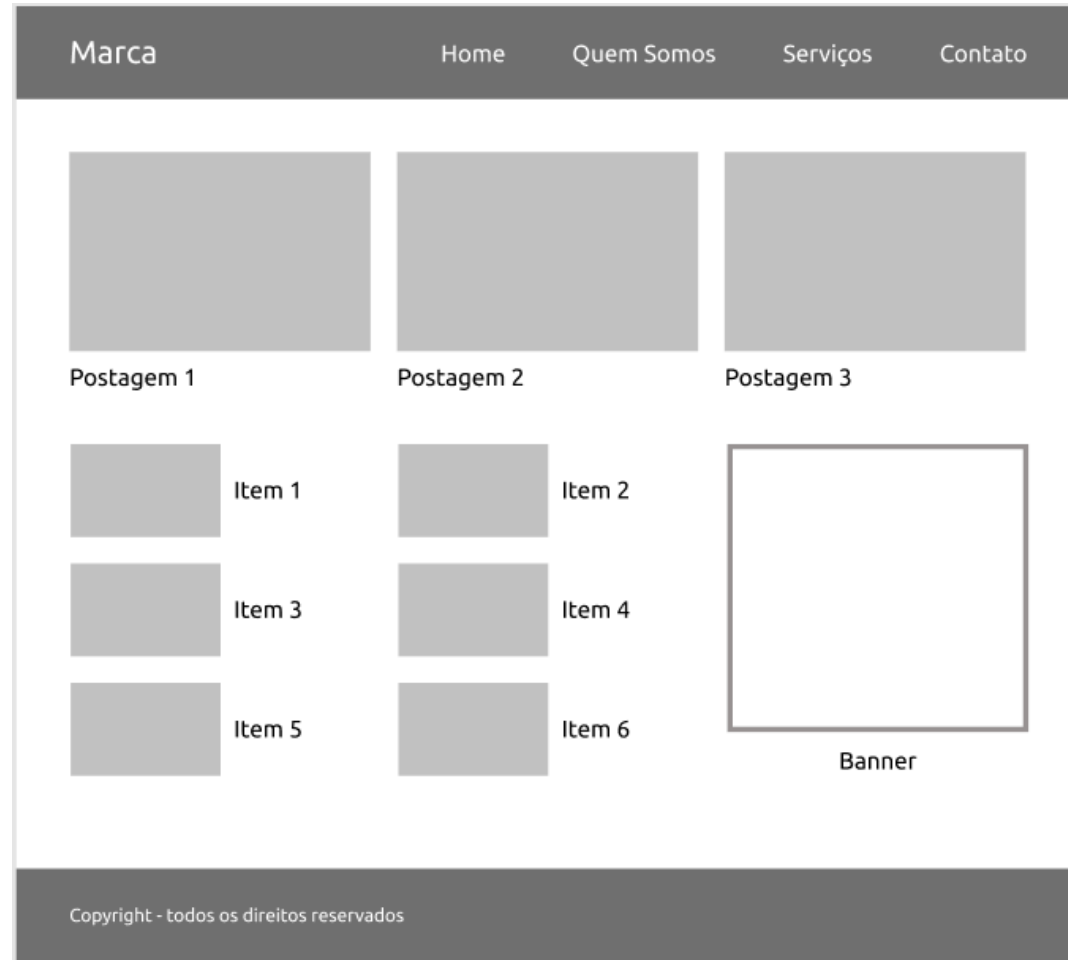
Fonte: <https://www.slideshare.net/valiac/css-grid-vs-flexbox>

Mão na massa



Codificar um exemplo de Grid Layout segundo o wireframe:

Fonte: Baseado nos exemplos de https://www.youtube.com/watch?v=x-4z_u8LcGc





PUC Minas

Layout Flex e Grid – parte 3

Prof. Marcos Arrais



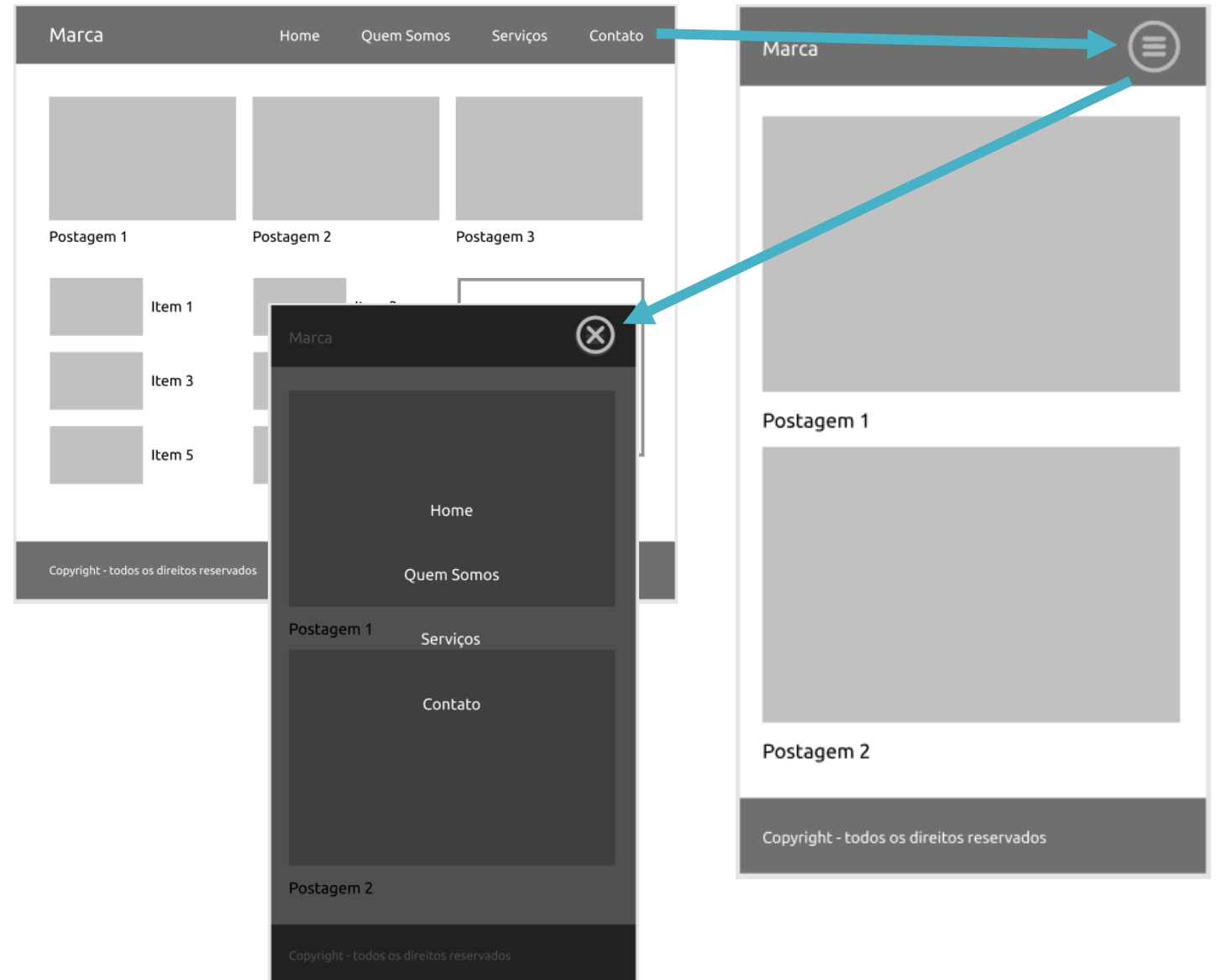
13



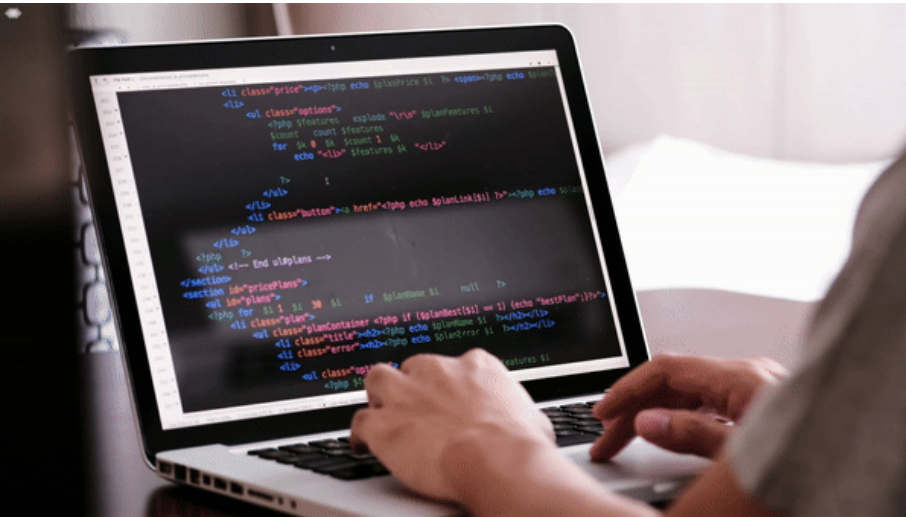
Grid Layout + Media Query

Para garantir uma maior consonância com o modelo de mobile é necessário ainda:

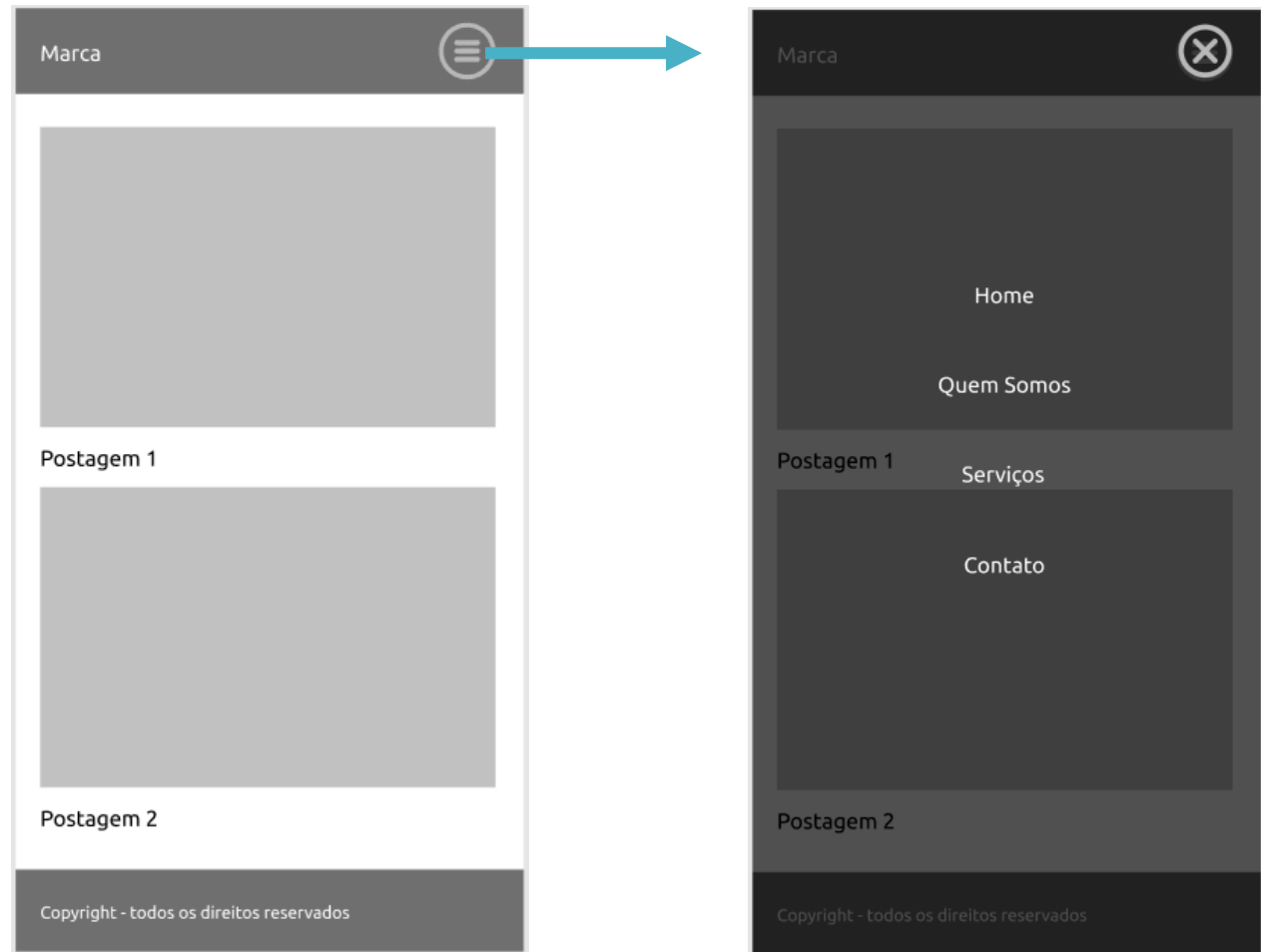
- Diminuir marca;
- Esconder menu;
- Criar ícone para chamar o menu em tela cheia;
- Estilizar menu com o overlay;
- Botão de fechar menu;



Mão na massa

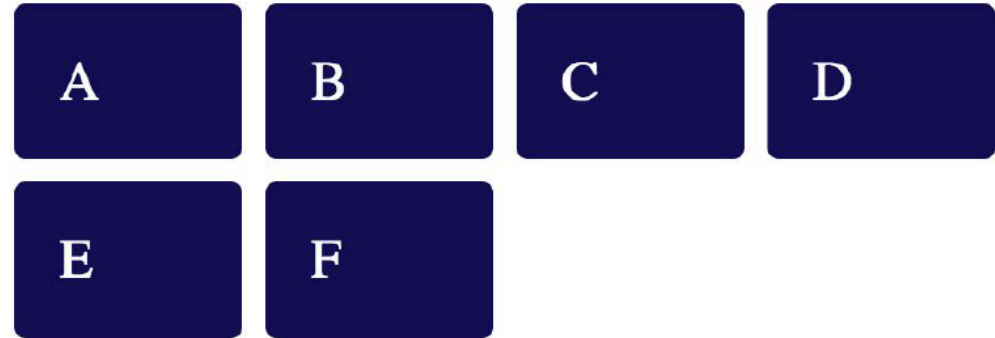


Codificar o menu no formato mobile:



Alternativas de Alinhamento

```
<div class="container">
  <div class="box a">A</div>
  <div class="box b">B</div>
  <div class="box c">C</div>
  <div class="box d">D</div>
  <div class="box e">E</div>
  <div class="box f">F</div>
</div>
```



Float

```
.container {
}

.box {
  float: left;
  margin: 5px;
  width: 100px;
}
```

Flex

```
.container {
  display: flex;
  flex-wrap: wrap;
}

.box {
  flex-basis: 100px;
  margin: 5px;
}
```

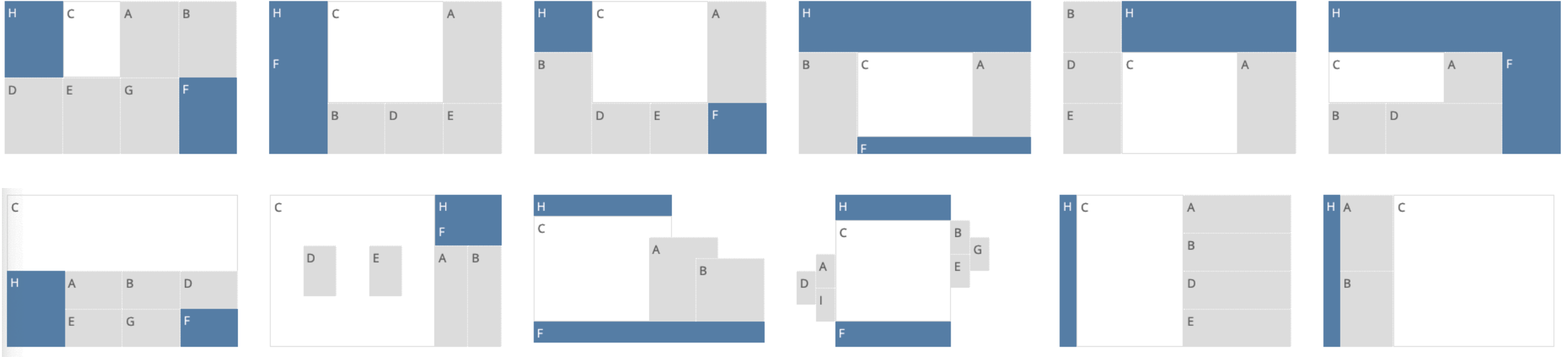
Grid

```
.container {
  display: grid;
  grid-gap: 10px;
  grid-template-columns:
    repeat(auto-fit, 100px);
}

.box {
}
```

Fonte: <https://www.slideshare.net/valiac/css-grid-vs-flexbox>

Tipos de Arranjos Possíveis com Grid Layout



Referencia de Estudo: <https://design.xwiki.org/xwiki/bin/view/Proposal/Skin10x/LayoutVariations>

Obrigado!

