

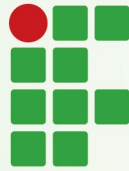
**INSTITUTO
FEDERAL**

Santa Catarina

**Curso Superior de Tecnologia em
Sistemas para Internet**

Programação para Internet I





**INSTITUTO
FEDERAL**
Santa Catarina

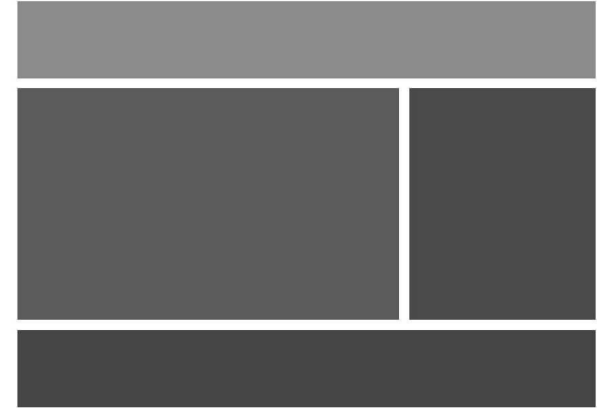
Tópicos desta aula:

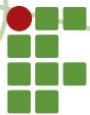
- **Introdução ao desenvolvimento de layouts para páginas WEB**
 - Grid



O que é GRID

- É uma técnica de construção de layouts com CSS em duas dimensões (linhas e colunas)
- A ideia principal é perceber a construção de uma interface como uma grade, sendo que alguns espaços podem ocupar múltiplas colunas ou múltiplas linhas.
- Assim como no Flexbox, temos propriedades para o container e propriedades para os itens
- Compartilha diversas propriedades com o flexbox.





**INSTITUTO
FEDERAL**
Santa Catarina

Grid elementos do container



Grid Container

```
<div class="container">
  <div class="item">A</div>
  <div class="item">B</div>
  <div class="item">C</div>
  <div class="item">D</div>
  <div class="item">E</div>
  <div class="item">F</div>
</div>
```

```
.container{
  display: grid;
}
```

- Assim como fizemos com o flexbox, o primeiro passo é informar ao navegador que nosso container trabalhará com grid.
- Neste caso o container irá utilizar a linha inteira (block)
- Também existe o valor inline-grid

Grid Container

```
<div class="container">
  <div class="item">A</div>
  <div class="item">B</div>
  <div class="item">C</div>
  <div class="item">D</div>
  <div class="item">E</div>
  <div class="item">F</div>
</div>
```

```
.container{
  display: grid;
  max-width: 650px;
}

.item{
  box-sizing: border-box;
  border: 1px solid black;
}
```

- Assim como fizemos com o flexbox, o primeiro passo é informar ao navegador que nosso container trabalhará com grid.
- Neste caso o container irá utilizar a linha inteira (block)
- Também existe o valor inline-grid

Grid Container

```
.container{  
  display: grid;  
  max-width: 650px;  
  grid-template-columns: 30% 70%;  
}
```

- As propriedades **grid-template-columns** e **grid-template-rows** determinam a quantidade e medidas das colunas e linhas, como numa grade

A	B
C	D
E	F

Grid Container

```
.container{  
  display: grid;  
  max-width: 650px;  
  grid-template-columns: 50% 3fr 2fr;  
}
```

- Nas propriedades de grid podem ser utilizadas quaisquer unidades de medida.
- Para nos ajudar a criar layouts flexíveis utilizando o grid, foi criada uma unidade nova.
- A unidade fr representa uma fração do espaço disponível no container do grid.

A	B	C
D	E	F

Grid Container

```
.container{  
  display: grid;  
  max-width: 650px;  
  grid-template-columns: 2fr 8fr 2fr;  
}
```

- Neste exemplo nosso espaço (grade) foi dividido em 12 “colunas”.

A	B	C
D	E	F

Grid Container

```
.container{  
  display: grid;  
  max-width: 650px;  
  grid-template-columns: 2fr 8fr 2fr;  
}
```

- Neste exemplo nosso espaço (grade) foi dividido em 12 “colunas”.

A	B	C
D	E	F

Grid Container

```
.container{  
  display: grid;  
  max-width: 650px;  
  grid-template-columns: 6fr repeat(3, 2fr);  
}
```

- Neste exemplo nosso espaço (grade) também foi dividido em 12 “colunas”, sendo 4 colunas reais.
- Utilizamos a função repeat para criar 3 colunas de 2fr

A	B	C	D
E	F		

Grid Container

```
.container{  
  display: grid;  
  max-width: 650px;  
  grid-template-columns: 150px 150px 150px;  
  grid-template-rows: 30px 50px;  
}
```

- Você também pode definir os tamanhos de suas linhas.
- Atenção com a questão da altura do conteúdo textual, que geralmente é imprevisível.

A	B	C
D	E	F

Grid Container

```
.container{  
  display: grid;  
  max-width: 650px;  
  grid-template-columns: 150px 150px 150px;  
  grid-template-rows: 30px 50px;  
}
```

- Você também pode definir os tamanhos de suas linhas.
- Atenção com a questão da altura do conteúdo textual, que geralmente é imprevisível.

A	B	C
D	E	Lorem ipsum dolor sit amet consectetur adipiscing elit. Facere ab est explicabo dicta provident magni saepe consequuntur quibusdam!

Grid Container

```
.container{  
  display: grid;  
  max-width: 650px;  
  grid-template-columns: 150px 150px 150px;  
  grid-template-rows: 30px minmax(50px, auto);  
}
```

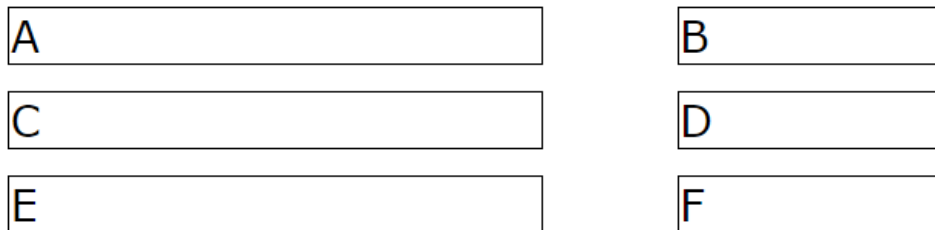
- Para contornar este problema, podemos utilizar a função `minmax()`

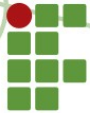
A	B	C
D	E	Lorem ipsum dolor sit amet consectetur adipisicing elit. Facere ab est explicabo dicta provident magni saepe consequuntur quibusdam!

Grid Container

```
.container{  
  display: grid;  
  max-width: 650px;  
  grid-template-columns: 200px 100px;  
  row-gap: 10px;  
  column-gap: 50px;  
}
```

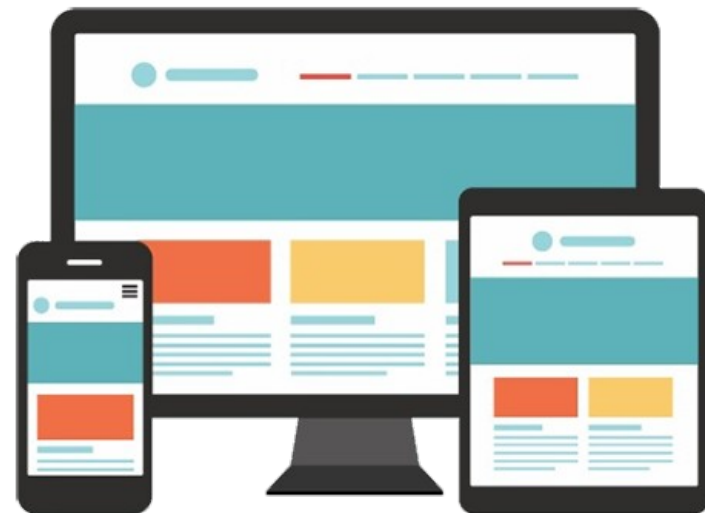
- Neste exemplo utilizamos as propriedades row-gap e column-gap para trabalhar com espaçamentos.



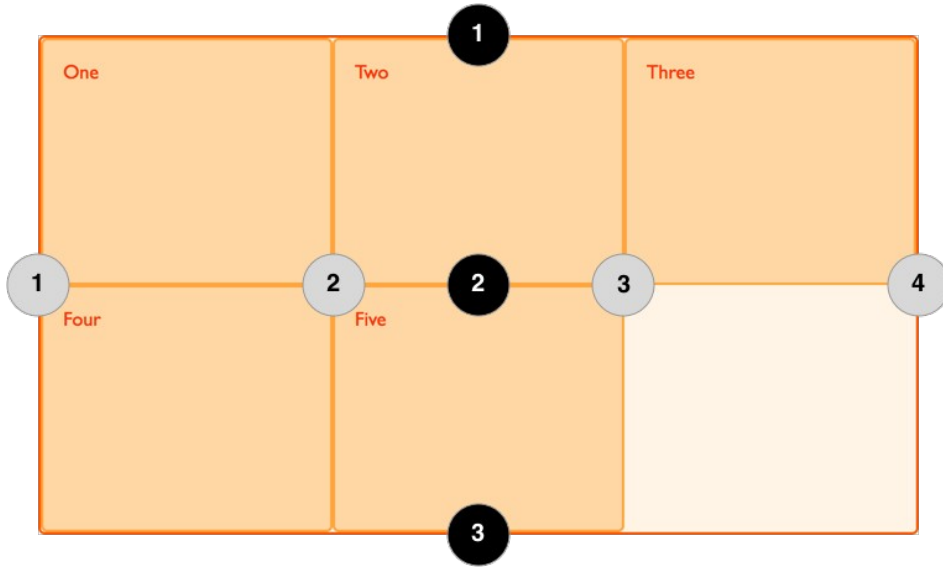


**INSTITUTO
FEDERAL**
Santa Catarina

Grid Layouts complexos



Linhas de grid



- Quando definimos um Grid, devemos observar as linhas e colunas externas e internas na contabilização do grid explícito para termos como base a construção de layouts complexos.

Grid – layouts complexos

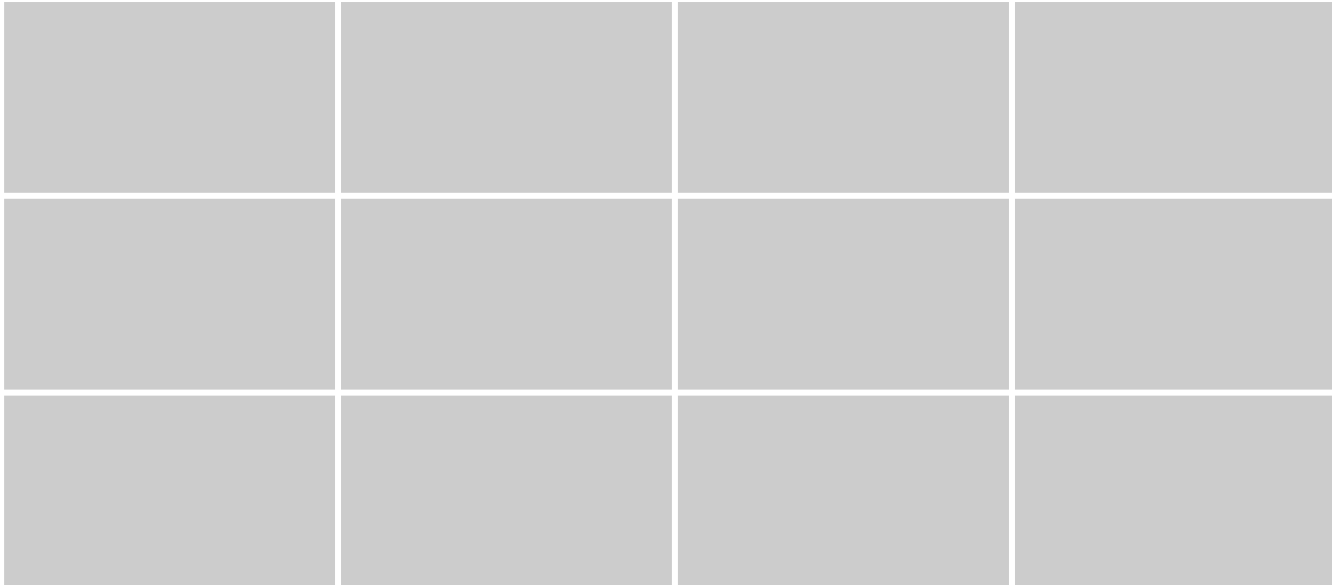
- Para os próximos exemplos, utilizaremos como base os códigos abaixo.

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

```
.container{
  max-width: 650px;
  background-color: #00FF00;;
  display: grid;
  grid-template-columns: repeat(4, auto);
  grid-template-rows: repeat(3, minmax(100px, auto));
}
```

Layout objetivo

- Definimos um grid explícito de 4 colunas e 3 linhas.



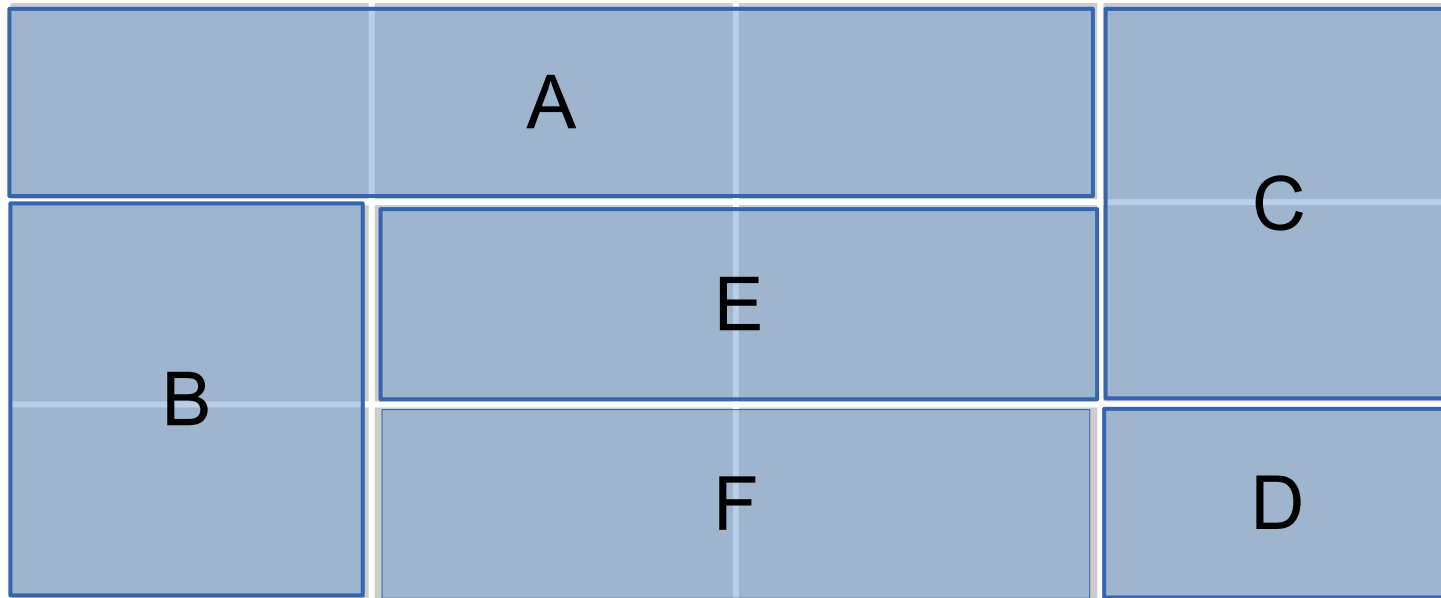
Layout objetivo

- Como apenas possuímos 6 itens, esta será a renderização.

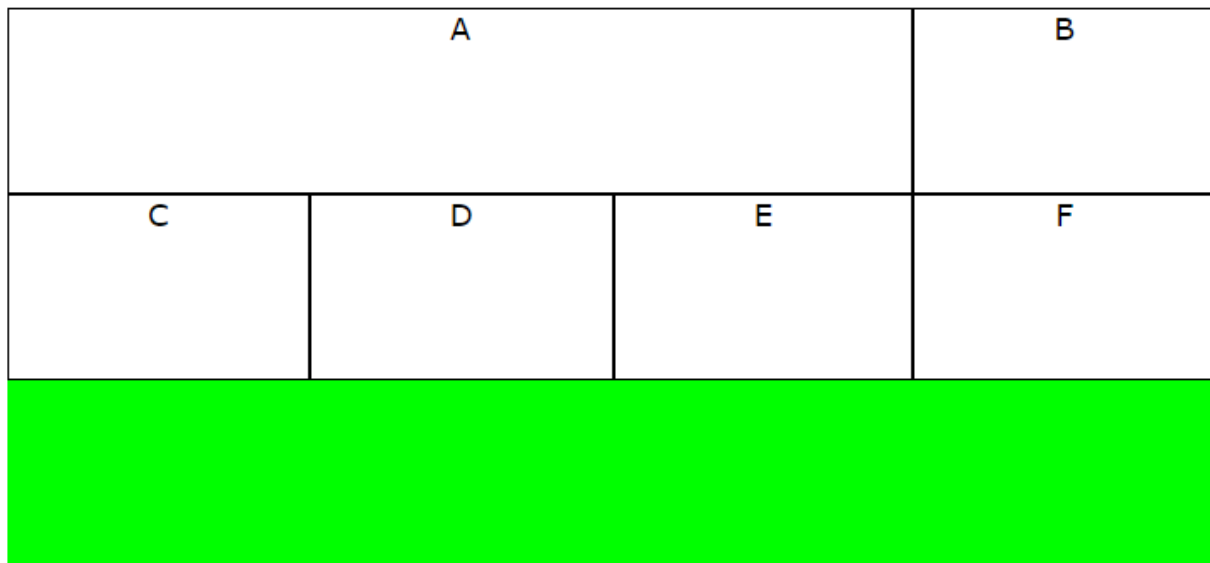
A	B	C	D
E	F		

Layout objetivo

- O objetivo é reorganizar os itens no grid explícito conforme plano abaixo
- Em que linha/coluna cada elemento começa/termina?

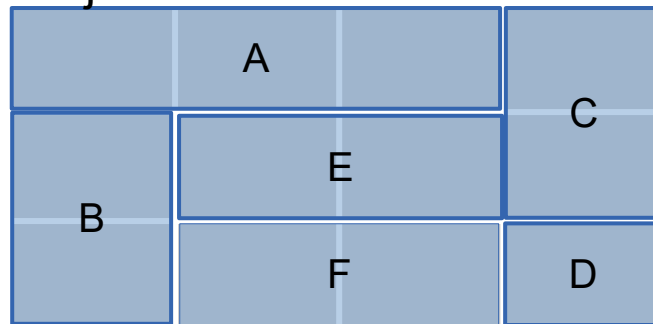


Grid – layouts complexos

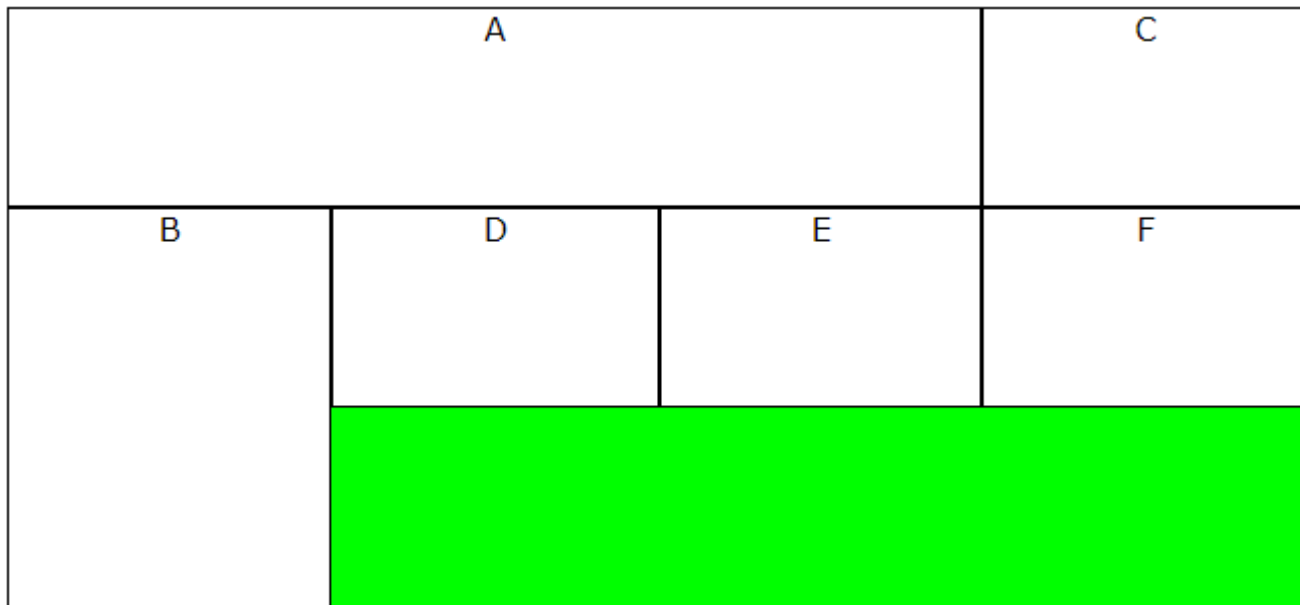


```
.item-a{  
  grid-column-start: 1;  
  grid-column-end: 4;  
}
```

Objetivo:

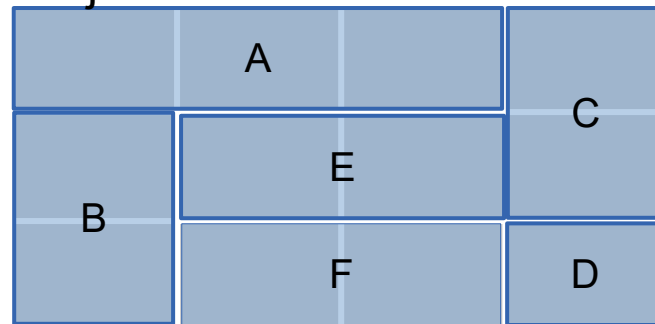


Grid – layouts complexos

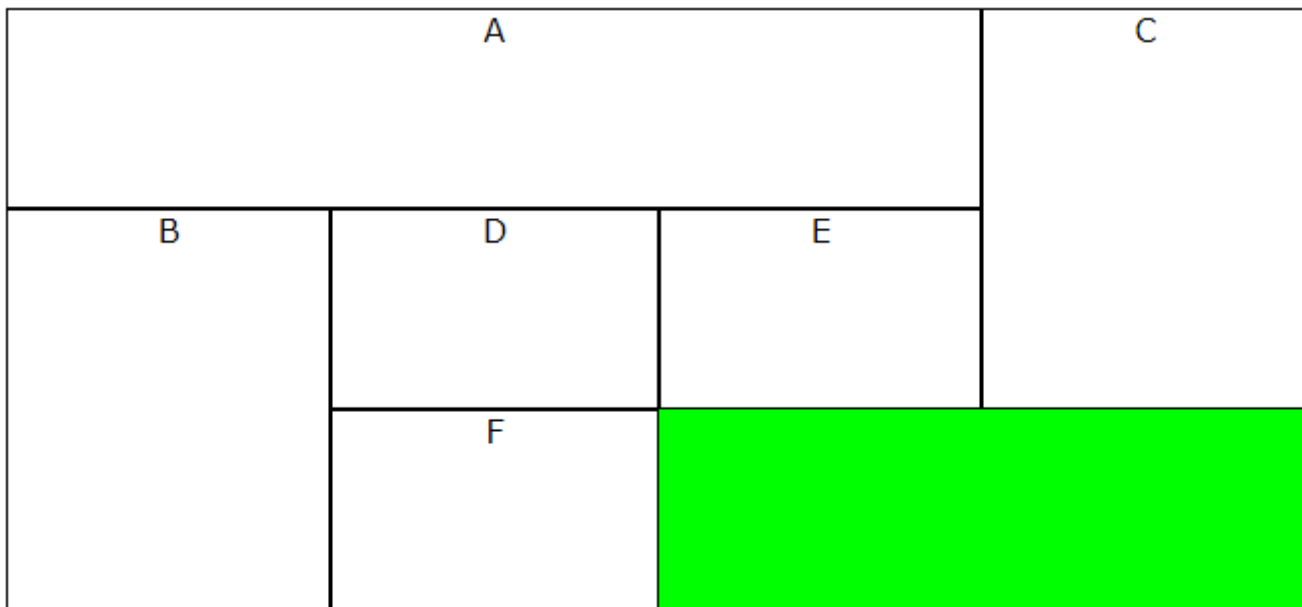


```
.item-b{  
  grid-row-start: 2;  
  grid-row-end: 4;  
}
```

Objetivo:

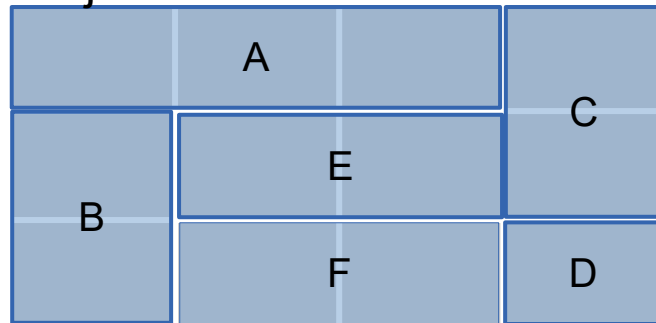


Grid – layouts complexos

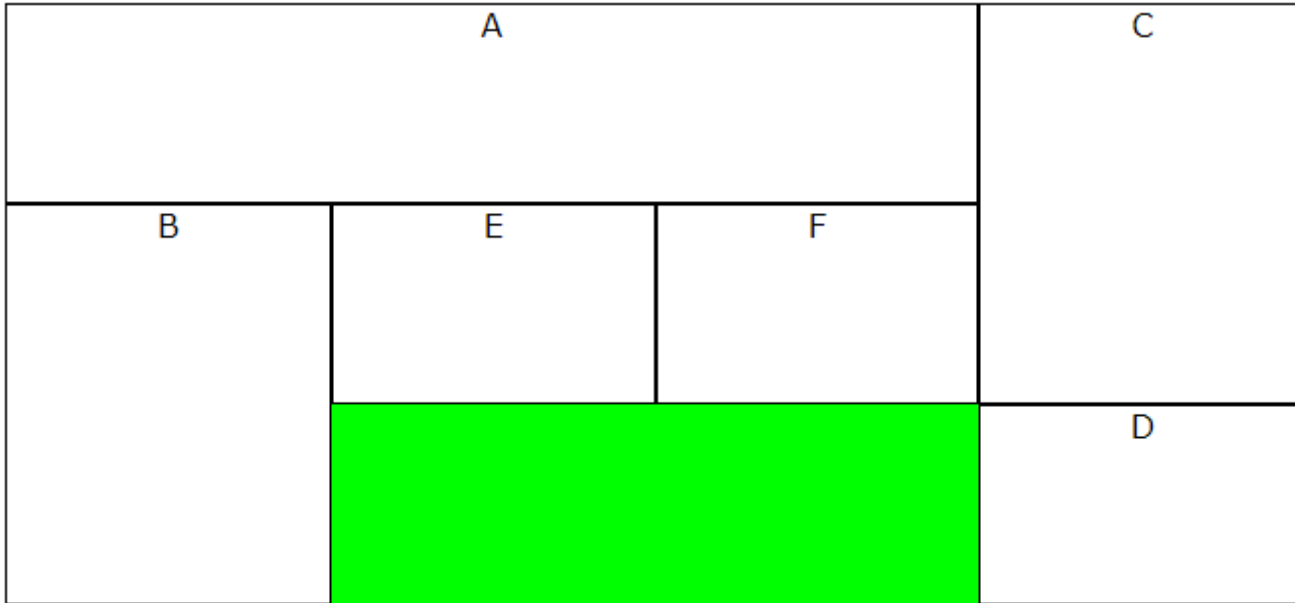


```
.item-c{  
  grid-row-start: 1;  
  grid-row-end: 3;  
  grid-column-start: 4;  
  grid-column-end: 5;  
}
```

Objetivo:

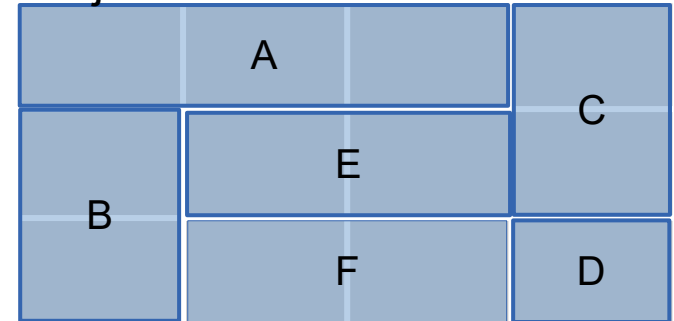


Grid – layouts complexos

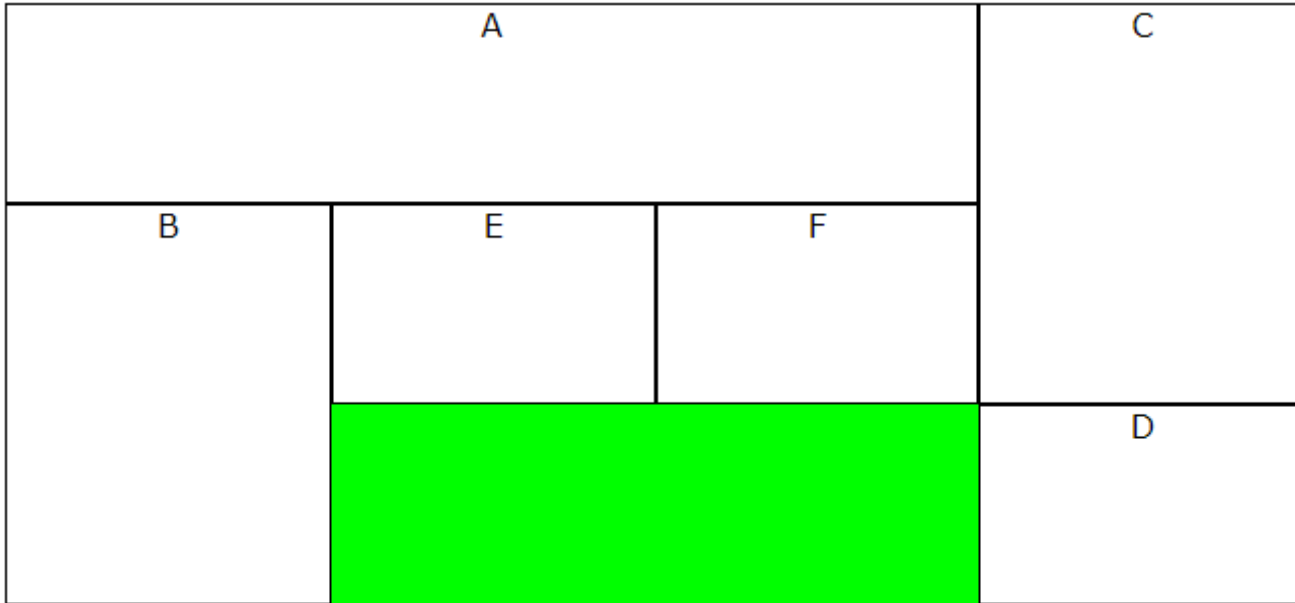


```
.item-d{  
  grid-row-start: 3;  
  grid-row-end: 4;  
  grid-column-start: 4;  
  grid-column-end: 5;  
}
```

Objetivo:

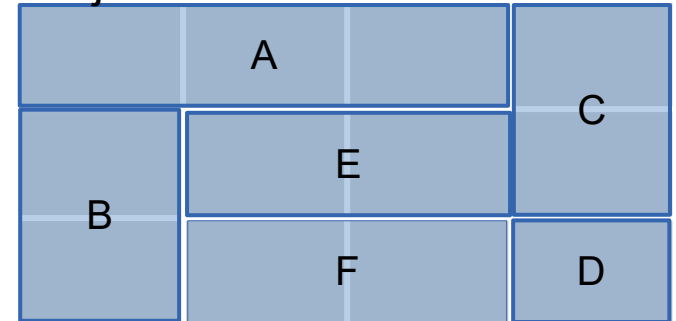


Grid – layouts complexos

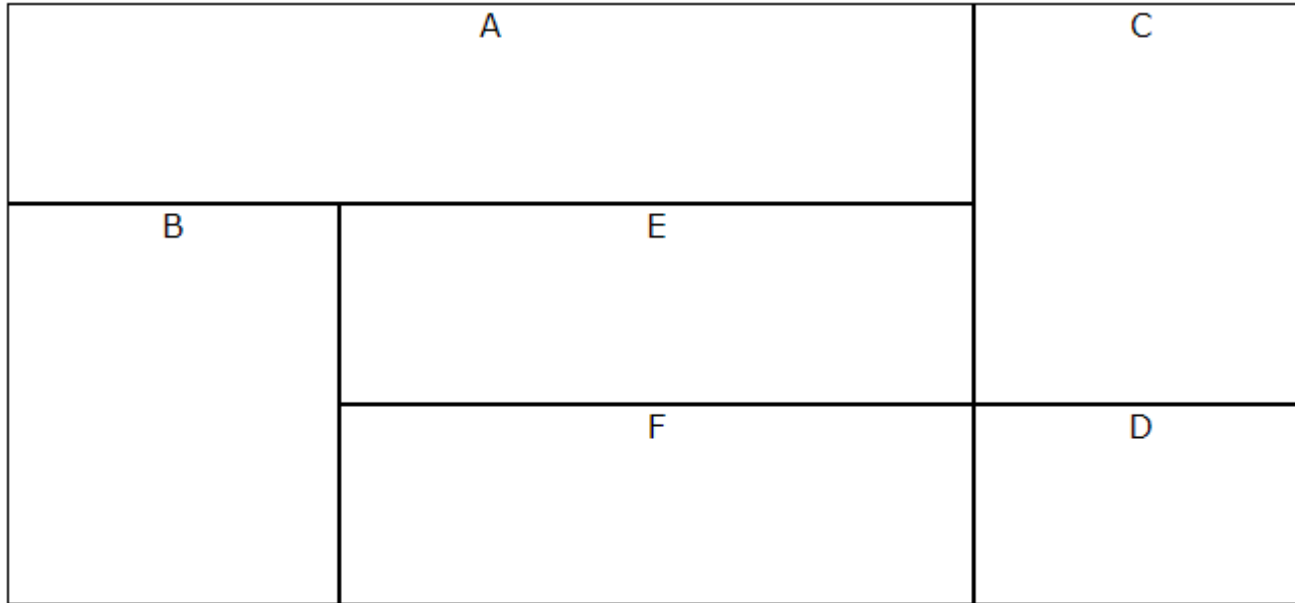


```
.item-e{  
  grid-row-start: 2;  
  grid-row-end: 3;  
  grid-column-start: 2;  
  grid-column-end: 4;  
}
```

Objetivo:

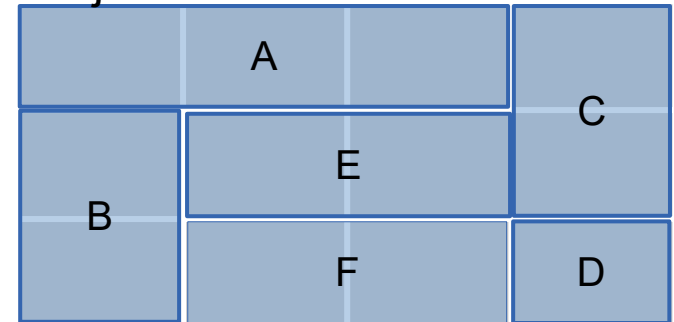


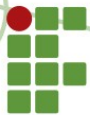
Grid – layouts complexos



```
.item-f{  
  grid-column-start: 2;  
  grid-column-end: 4;  
}
```

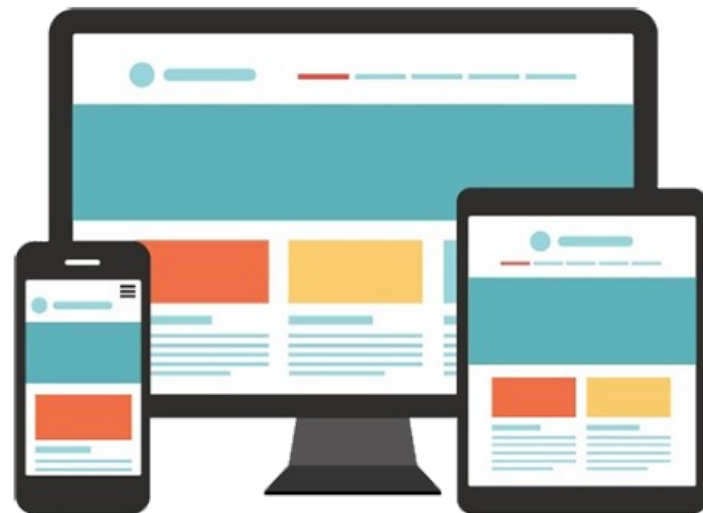
Objetivo:





**INSTITUTO
FEDERAL**
Santa Catarina

Grid areas





```
.container{
  max-width: 900px;
  margin:auto;
  background-color: #00FF00;;
  display: grid;
  grid-template-columns: 150px auto 150px;
  grid-template-rows: repeat(3, minmax(100px, auto));
  /* areas */
  grid-template-areas: "topo topo topo"
                      "menu principal banner"
                      "rodape rodape rodape";
}

.container > *{
  border:1px solid black;
  box-sizing: border-box;
  padding:20px;
}
```

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

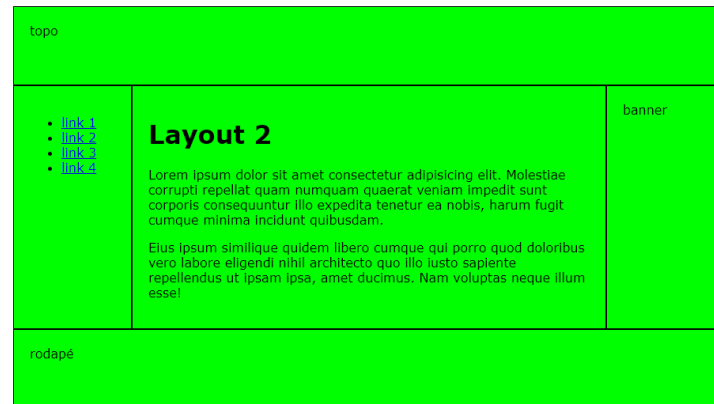
nav{
  grid-area: menu;
}

main{
  grid-area: principal;
}

aside{
  grid-area: banner;
}

footer{
  grid-area: rodape;
}
```

```
<div class="container">
  <header></header>
  <nav></nav>
  <main></main>
  <aside></aside>
  <footer></footer>
</div>
```



Desenvolvendo...

