

Flexbox + CSS position

Utilizando as ferramentas em conjunto

Veja o seguinte exemplo:



ENDEREÇO
Av. Bernadino de Campos, 98
São Paulo, SP - 23145-678

CONTATO
info@gelateria.com
Tel.: (11) 3456-7890

HORÁRIOS
Aberto todos os dias
10:00 - 22:00



© Gelateria Gourmet
Criado por dev.com.br



Neste caso os elementos foram posicionados com o Flexbox.



```
<div class="container-rodape">
  <div class="container-imagem">...
</div>
<div class="container-contato">...
</div>
<div class="container-descricao">
  <div class="container-redes-sociais">...
  </div>
  <div>...
</div>
  <a href="#topo"></a>
</div>
</div>
```

```
.container-rodape {
  padding: 30px 20px;
  display: flex;
  flex-direction: column;
  align-items: center;
  gap: 60px;
  background-color: #000000;
}

.container-contato {
  width: 100%;
  max-width: 1024px;
  display: flex;
  justify-content: space-between;
  flex-wrap: wrap;
  gap: 20px;
}

.container-contato div {
  padding: 10px 0;
  border: 1px solid #808080;
  border-radius: 4px;
  flex: 1 1 240px;
  text-align: center;
}

.container-descricao {
  width: 100%;
  display: flex;
  align-items: center;
  justify-content: space-around;
}

.container-descricao a {
  width: 50px;
  height: 50px;
  background-image: url("seta.png");
  background-position: center;
  background-size: cover;
  background-color: #cccccc;
  border-radius: 4px;
}

.container-redes-sociais {
  display: flex;
  gap: 10px;
}
```

Mas e se quisermos que o **link**
(botão com seta para cima) fique
flutuando sobre o container-descricao?



ENDEREÇO

Av. Bernadino de Campos, 98
São Paulo, SP - 23145-678

CONTATO

info@gelateria.com
Tel.: (11) 3456-7890

HORÁRIOS

Aberto todos os dias
10:00 - 22:00

container-descricao



Link

© Gelateria Gourmet
Criado por dev.com.br

Com o Flexbox, **não é possível** gerar esse comportamento.



```
.container-descricao {  
  width: 100%;  
  display: flex;  
  align-items: center;  
  justify-content: space-around;  
}
```

O **Flexbox** consegue posicionar o link **somente** dentro do container-descricao.

A forma **ideal** para posicionar esse link é utilizando o **CSS position**.



```
.container-descricao {  
  width: 100%;  
  display: flex;  
  align-items: center;  
  justify-content: space-around;  
  position: relative;  
}  
.container-descricao a {  
  width: 50px;  
  height: 50px;  
  background-image: url("seta.png");  
  background-position: center;  
  background-size: cover;  
  background-color: #dddddd;  
  border-radius: 4px;  
  position: absolute;  
  bottom: 60px;  
  right: 0px;  
}
```


Com isso o container-descricao terá tanto o **Flexbox** quanto o **position** aplicado em seu código.

```
.container-descricao {  
  width: 100%;  
  display: flex;  
  align-items: center;  
  justify-content: space-around;  
  position: relative;  
}
```

```
.container-descricao a {  
  width: 50px;  
  height: 50px;  
  background-image: url("seta.png");  
  background-position: center;  
  background-size: cover;  
  background-color: #dddddd;  
  border-radius: 4px;  
  position: absolute;  
  bottom: 60px;  
  right: 0px;  
}
```



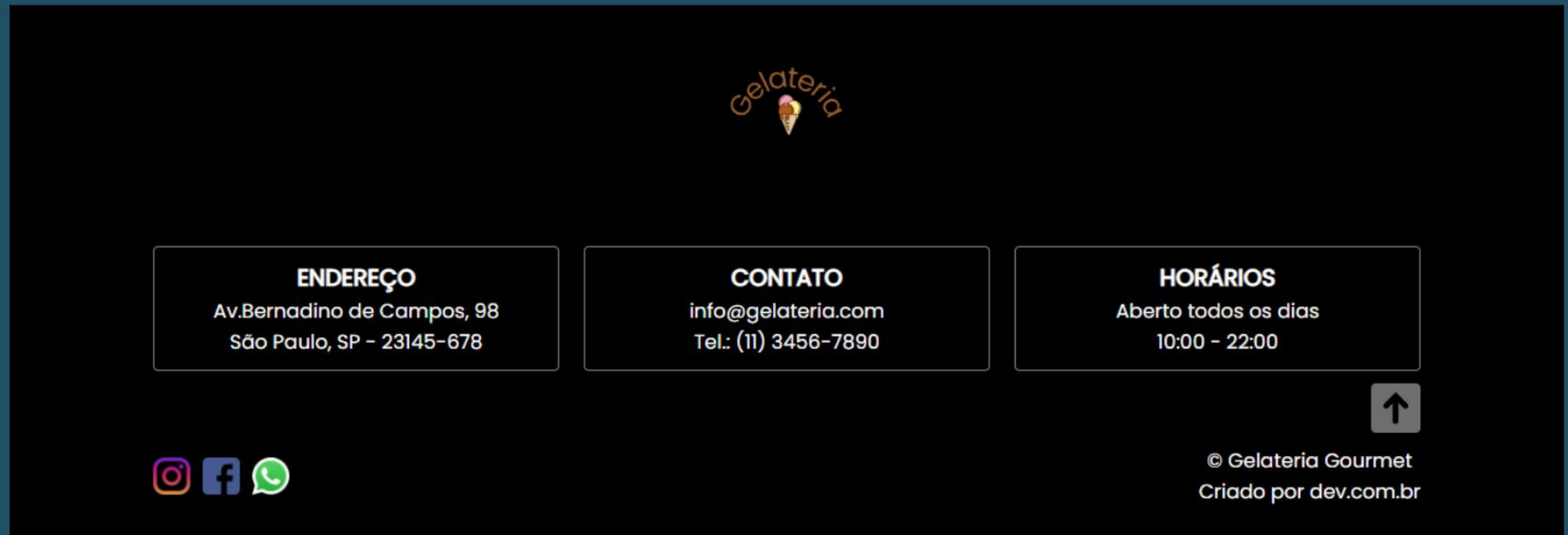
elemento posicionado com o position ←



© Gelateria Gourmet
Criado por dev.com.br

→ elementos posicionados com o Flexbox ←

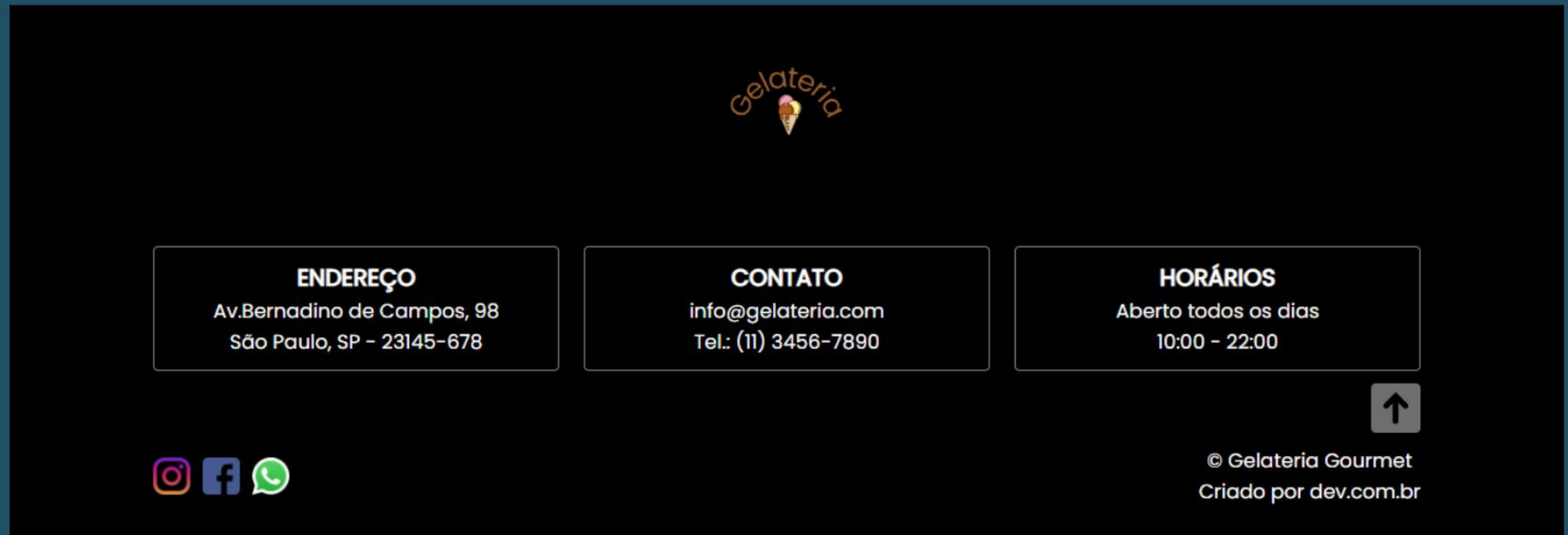
Ou seja, com o Flexbox + o CSS position podemos **criar novos efeitos** como visto neste exemplo.



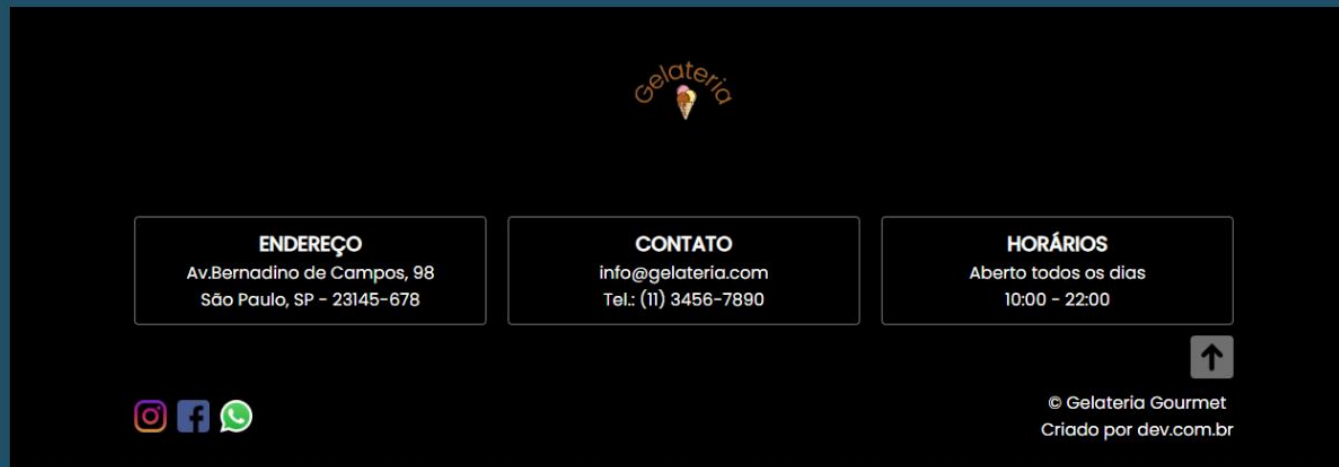
Flexbox + CSS position

Conhecendo o exemplo

O exemplo dessa aula é um rodapé de uma página HTML.



Veja o código HTML do exemplo.



```
<div class="container-rodape">
  <div class="container-imagem">
    
  </div>
  <div class="container-contato">
    <div>
      <h3>ENDEREÇO</h3>
      <p>Av. Bernadino de Campos, 98</p>
      <p>São Paulo, SP - 23145-678</p>
    </div>
    <div>
      <h3>CONTATO</h3>
      <p>info@gelateria.com</p>
      <p>Tel.: (11) 3456-7890</p>
    </div>
    <div>
      <h3>HORÁRIOS</h3>
      <p>Aberto todos os dias</p>
      <p>10:00 - 22:00</p>
    </div>
  </div>

  <div class="container-descricao">
    <div>
      
      
      
    </div>
    <div>
      <p>© Gelateria Gourmet</p>
      <p>Criado por dev.com.br</p>
    </div>
    <a href="#topo"></a>
  </div>
</div>
```

index.html

Neste exemplo temos um container-rodape (<div>) que possui três subcontainers (<div>).

container-rodape (<div>)



1º subcontainer (<div>)

2º subcontainer (<div>)

ENDEREÇO

Av. Bernadino de Campos, 98
São Paulo, SP - 23145-678

CONTATO

info@gelateria.com
Tel.: (11) 3456-7890

HORÁRIOS

Aberto todos os dias
10:00 - 22:00

3º subcontainer (<div>)



© Gelateria Gourmet
Criado por dev.com.br

O 1º subcontainer terá **uma imagem** ().

1º subcontainer (<div>)



```
<div class="container-imagem">  
    
</div>
```

O 2º subcontainer terá **três cards** (<div>).

2º subcontainer (<div>)

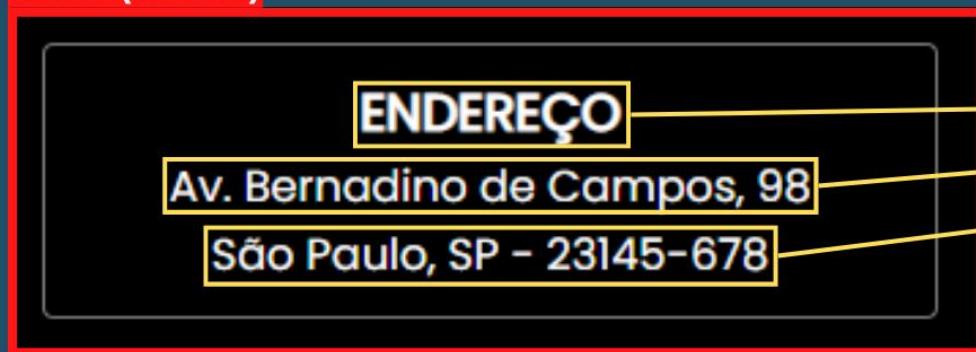


card (<div>)

```
<div class="container-contato">
  <div>
    <h3>ENDEREÇO</h3>
    <p>Av. Bernadino de Campos, 98</p>
    <p>São Paulo, SP - 23145-678</p>
  </div>
  <div>...
</div>
  <div>...
</div>
</div>
```


Cada card terá um subtítulo (<h3>) e dois parágrafos (<p>).

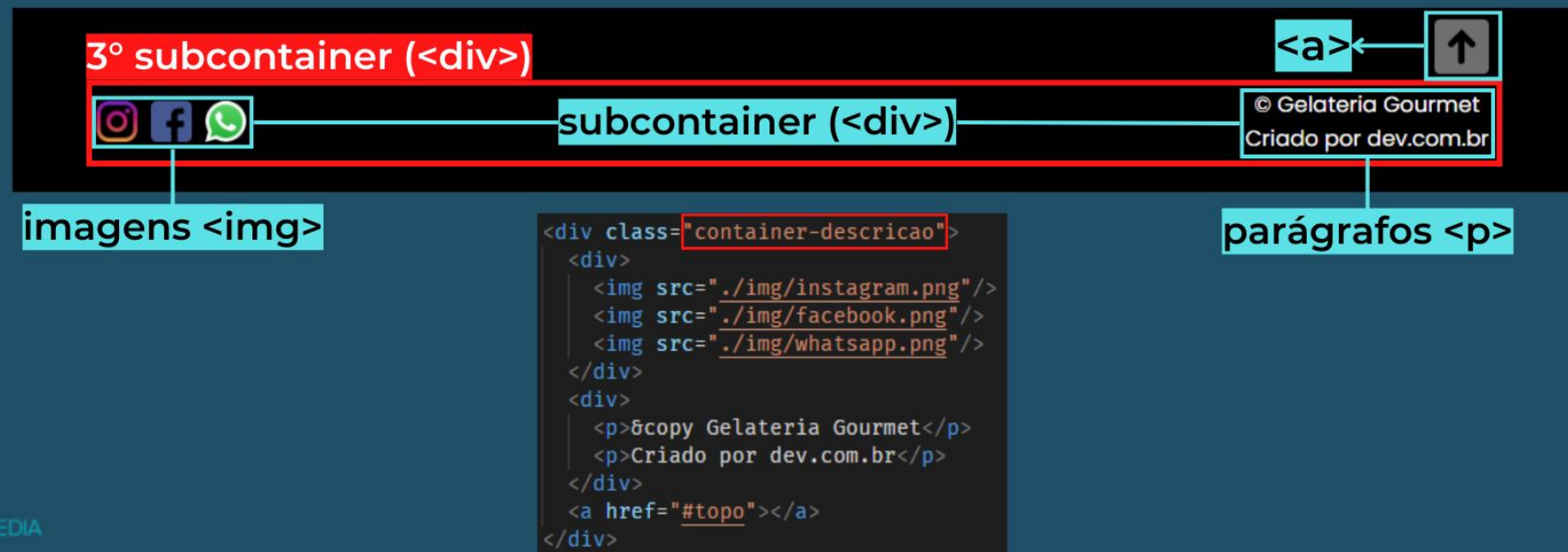
card (<div>)



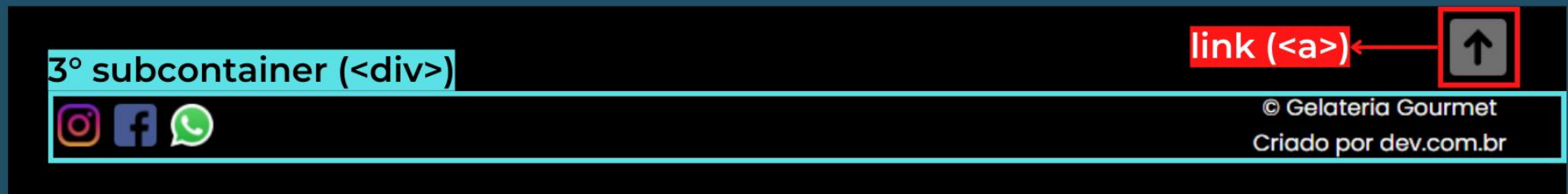
```
<div>  
  <h3>ENDEREÇO</h3>  
  <p>Av. Bernadino de Campos, 98</p>  
  <p>São Paulo, SP - 23145-678</p>  
</div>
```

O 3º subcontainer terá dois subcontainers (<div>) e um link (<a>).

Um subcontainer terá somente imagens () e outro terá somente texto (<p>).



O link deverá ser **posicionado** de forma que fique **flutuando** acima do 3º subcontainer.



Os cards devem começar a **quebrar linha** a partir de **resoluções** menores que **805px**.



←-----805px-----→

←-----540px-----→

←-----375px-----→

Veja o mesmo exemplo em uma versão wireframe.

***container-rodape**



***subcontainer**

***subcontainer**

<Subtítulo>

<parágrafo>

<parágrafo>

***card**

<Subtítulo>

<parágrafo>

<parágrafo>

***card**

<Subtítulo>

<parágrafo>

<parágrafo>

***card**

<link>

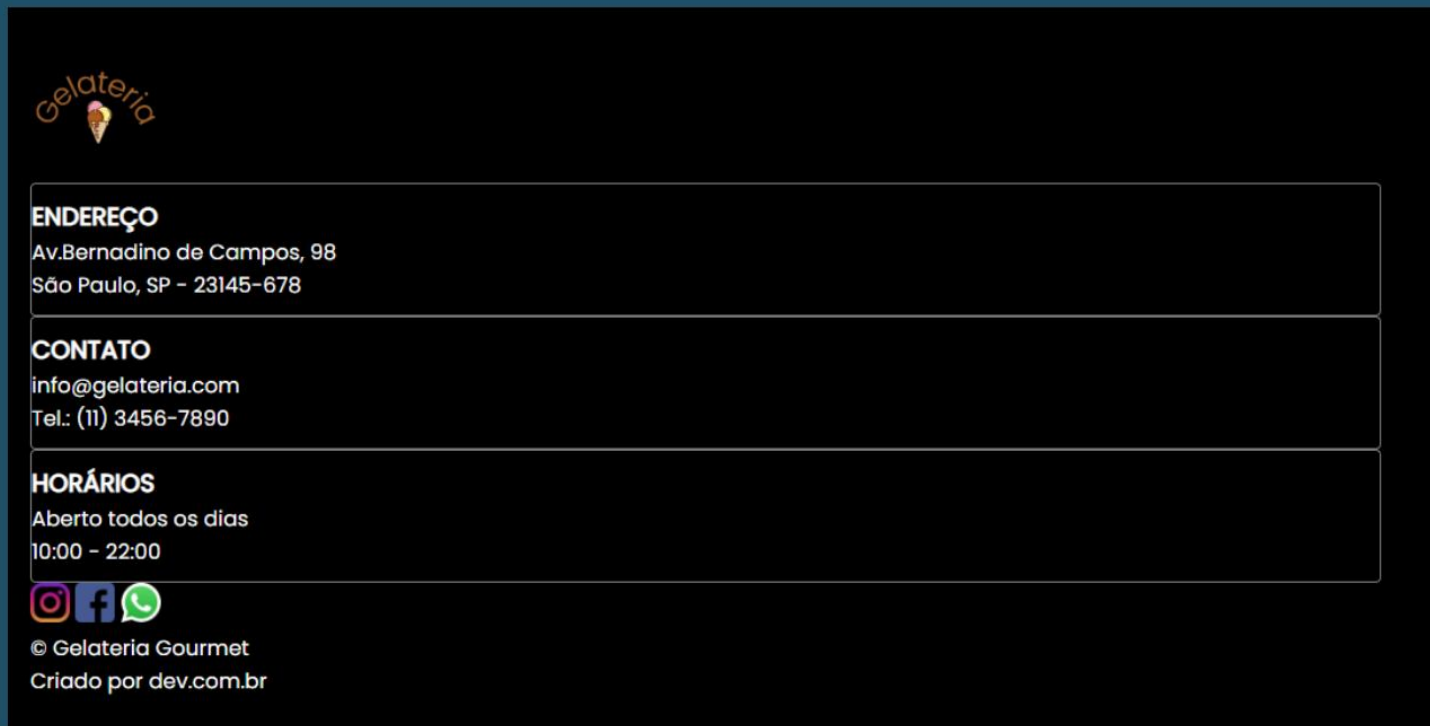
***subcontainer**

<paragrafo>
<paragrafo>

Flexbox + CSS position

Entendendo o código do exemplo

Primeiro, veja o exemplo **sem o Flexbox** e **sem o position** aplicado.



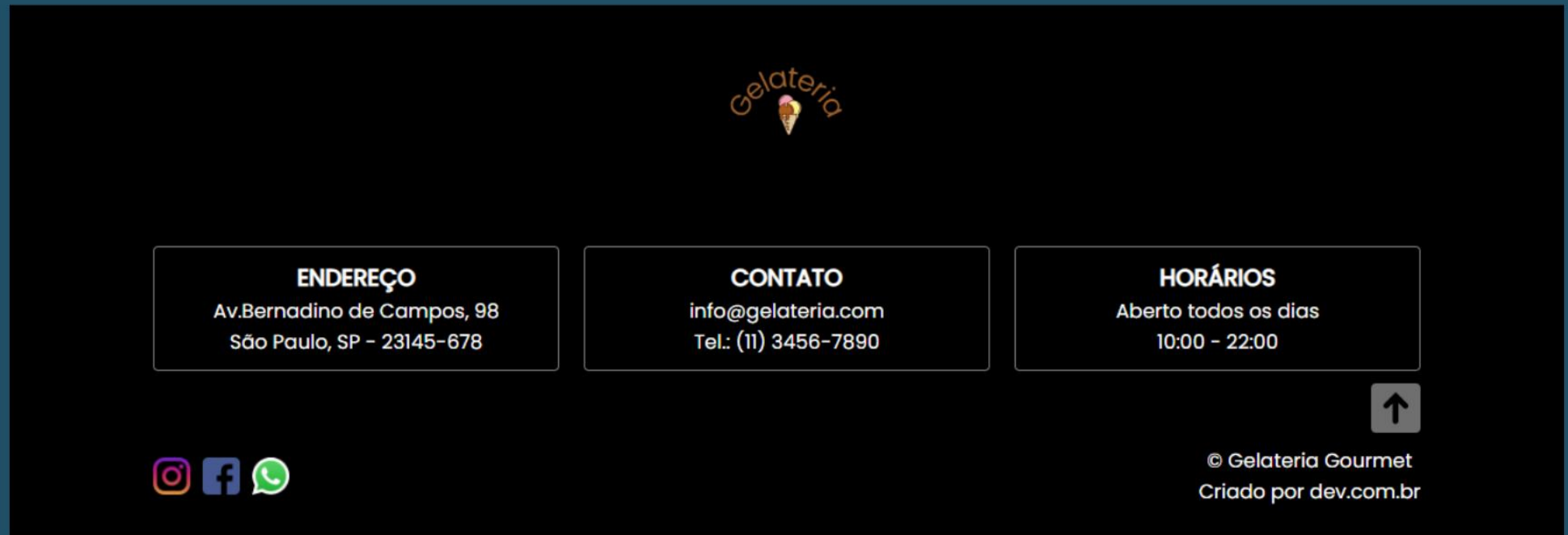
```
.container-rodape {  
  padding: 30px 20px;  
  background-color: black;  
}  
.container-imagem {  
  width: 100px;  
}  
.container-imagem img {  
  width: 100%;  
  height: 100%;  
}  
.container-contato {  
  width: 100%;  
  max-width: 1024px;  
}  
.container-contato div {  
  padding: 10px 0;  
  border: 1px solid gray;  
  border-radius: 4px;  
}  
.container-descricao {  
  width: 100%;  
  max-width: 1024px;  
}  
.container-descricao a {  
  width: 50px;  
  height: 50px;  
  background-image: url("seta.png");  
  background-position: center;  
  background-size: cover;  
  background-color: #dddddd;  
  border-radius: 4px;  
}
```

Repare que definimos uma **largura** para **todos** os subcontainers.



```
.container-imagem {  
  width: 100px;  
}  
.container-imagem img {  
  width: 100%;  
  height: 100%;  
}  
.container-contato {  
  width: 100%;  
  max-width: 1024px;  
}  
.container-descricao {  
  width: 100%;  
  max-width: 1024px;  
}
```

O objetivo é que no **final** o exemplo **seja**
apresentado da seguinte forma:



Para começar, aplicamos o **Flexbox** no **container-rodape**.



```
.container-rodape {  
  padding: 30px 20px;  
  display: flex;  
  flex-direction: column;  
  align-items: center;  
  gap: 60px;  
  background-color: black;  
}
```

Os subcontainers ficarão em **coluna**, **alinhados** ao **centro** e **espaçados** entre si (*gap: 60px*).

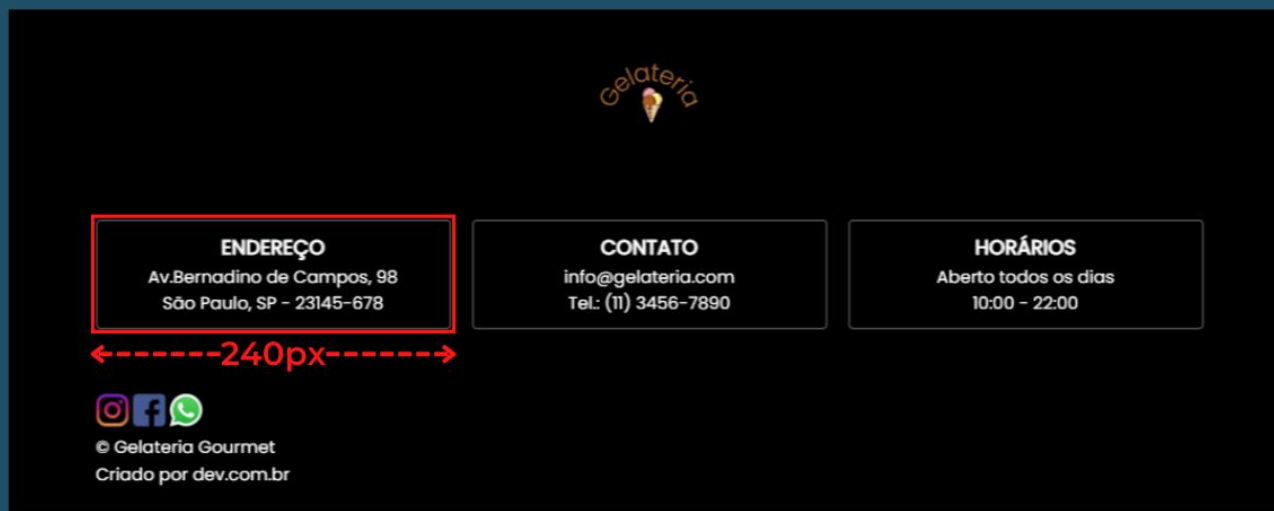
Em seguida, aplicamos o **Flexbox** no 2º subcontainer.



```
.container-contato {  
  width: 100%;  
  max-width: 1024px;  
  display: flex;  
  justify-content: space-evenly;  
}
```

Os cards ficarão em **linha** e **igualmente espaçados** entre si.

Depois, definimos um **crescimento**, uma **largura** e também **alinhamos** o seu conteúdo interno.



```
.container-contato div {  
  padding: 10px 0;  
  border: 1px solid gray;  
  border-radius: 4px;  
  flex: 1 1 240px;  
  text-align: center;  
}
```

Agora os **cards** possuem uma largura de **240px** e o texto está **alinhado ao centro**.

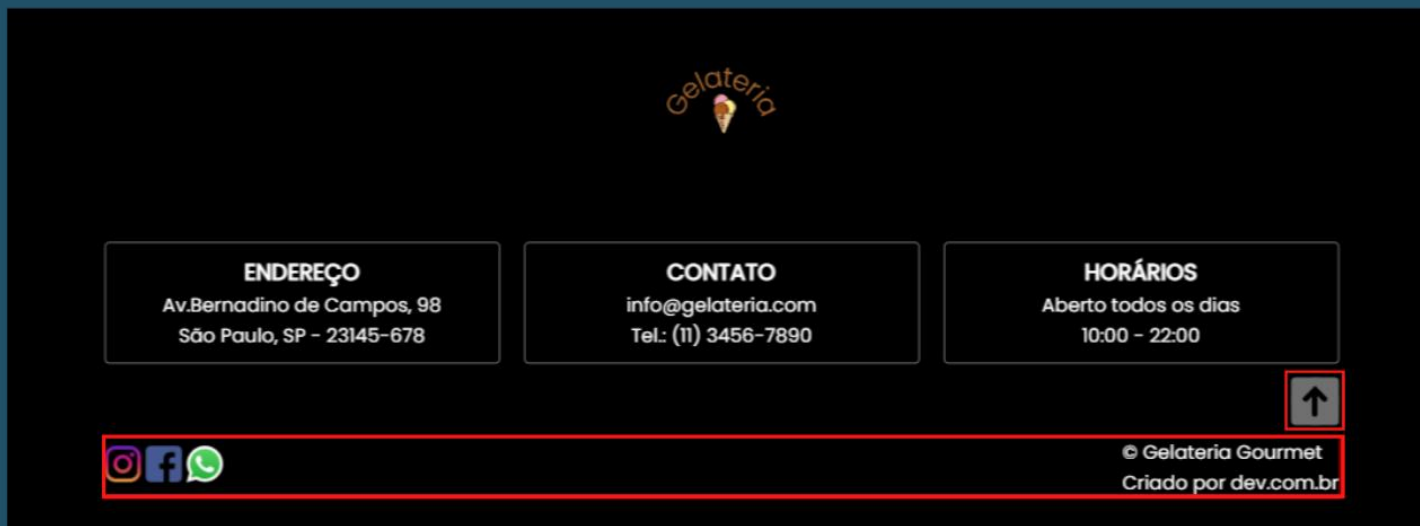
Como próximo passo, aplicamos o **Flexbox** no 3º subcontainer.



```
.container-descricao {  
  width: 100%;  
  max-width: 1024px;  
  display: flex;  
  align-items: center;  
  justify-content: space-between;  
}
```

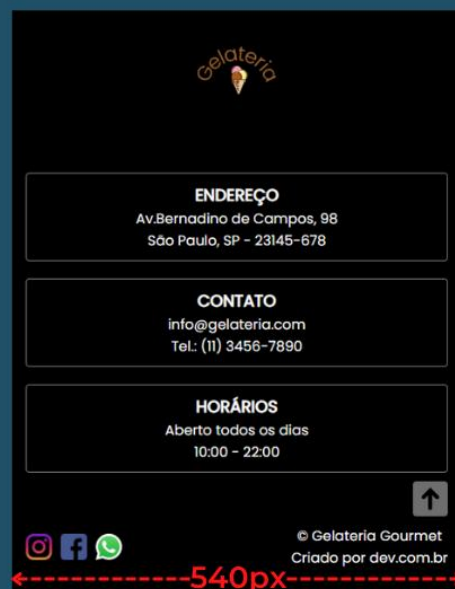
Os elementos ficarão em **linha**, **centralizados** na vertical e **totalmente espaçados** entre si.

Para deixar o link flutuando sobre o 3º subcontainer, aplicamos o CSS position.



```
.container-descricao {  
  width: 100%;  
  max-width: 1024px;  
  display: flex;  
  align-items: center;  
  justify-content: space-between;  
  position: relative;  
}  
  
.container-descricao a {  
  width: 40px;  
  height: 40px;  
  background-image: url("seta.png");  
  background-position: center;  
  background-size: cover;  
  background-color: #dddddd;  
  border-radius: 4px;  
  position: absolute;  
  bottom: 60px;  
  right: 0px;  
}
```

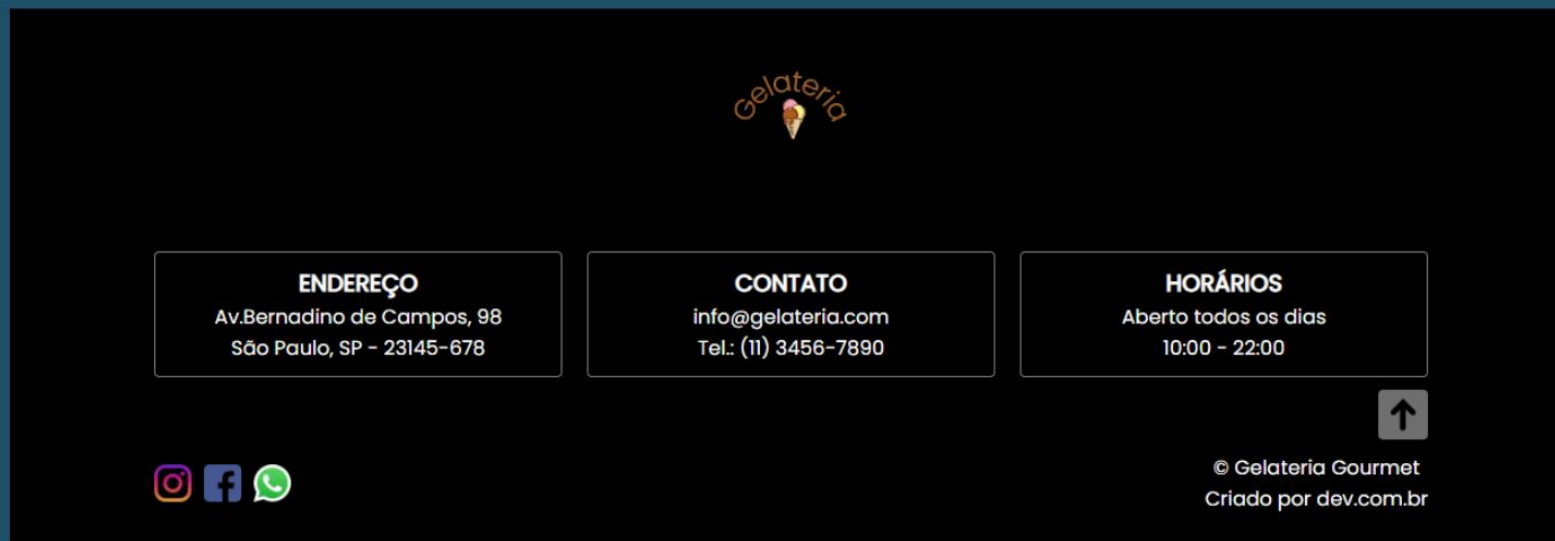
Por fim, definimos a **quebra de linha**
do **2º subcontainer**.



```
.container-contato {  
  width: 100%;  
  max-width: 1024px;  
  display: flex;  
  justify-content: space-evenly;  
  flex-wrap: wrap;  
  gap: 20px;  
}
```

Note que mesmo com a **mudança** de resolução, o
link continua posicionado sobre o 3º subcontainer.

Veja o resultado do exemplo com o Flexbox e o position aplicados.



```
.container-rodape {
  padding: 30px 20px;
  display: flex;
  flex-direction: column;
  align-items: center;
  gap: 60px;
  background-color: #000000;
}
.container-imagem {
  width: 100px;
}
.container-imagem img {
  width: 100%;
  height: 100%;
}
.container-contato {
  width: 100%;
  max-width: 1024px;
  display: flex;
  justify-content: space-between;
  flex-wrap: wrap;
  gap: 20px;
}
.container-contato div {
  padding: 10px 0;
  border: 1px solid #cccccc;
  border-radius: 4px;
  flex: 1 1 240px;
  text-align: center;
}
.container-descricao {
  width: 100%;
  max-width: 1024px;
  display: flex;
  align-items: center;
  justify-content: space-between;
  position: relative;
}
.container-descricao a {
  width: 40px;
  height: 40px;
  background-image: url("seta.png");
  background-position: center;
  background-size: cover;
  background-color: #cccccc;
  border-radius: 4px;
  position: absolute;
  bottom: 60px;
  right: 0px;
}
```