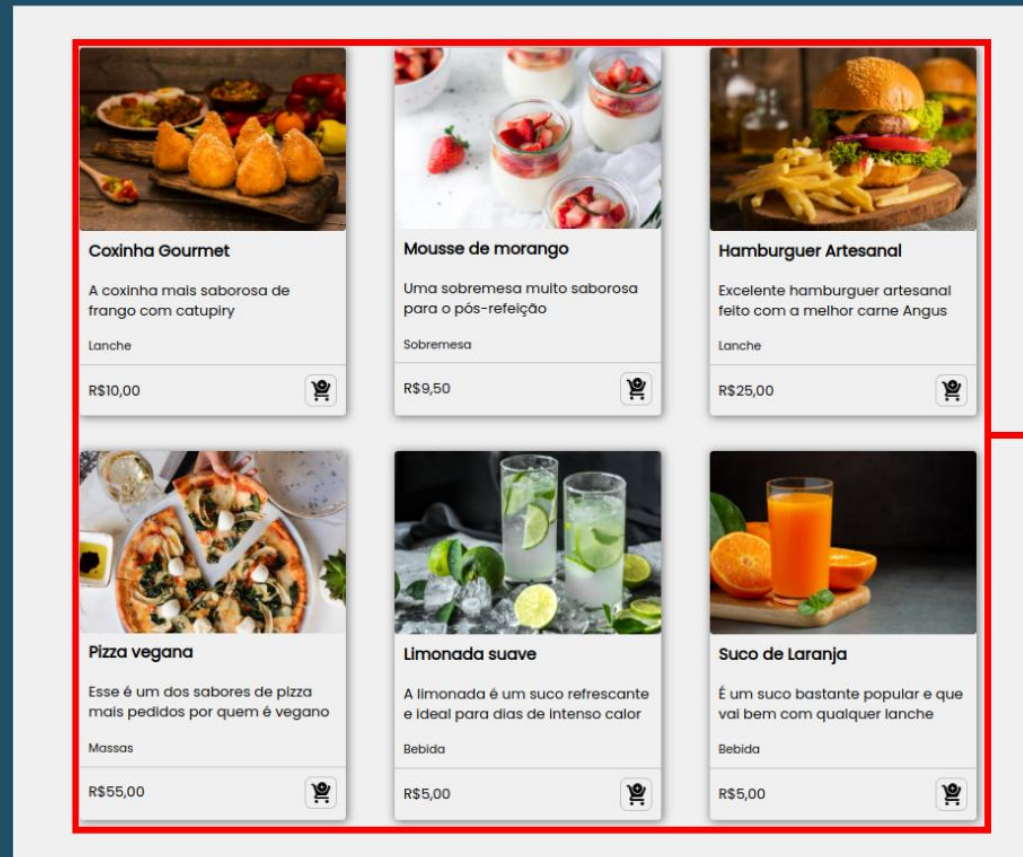


Terceiro exemplo prático

Flexbox aplicado em outros contextos

Até agora, vimos o **Flexbox** sendo usado para **posicionar cards** e seus elementos.



Cards e seus
elementos
posicionados
com o Flexbox

Contudo, o Flexbox pode ser usado para **posicionar qualquer elemento** de uma página HTML.



Observe a **diferença** entre o comportamento da página com o **CSS position** e com o **Flexbox**.



Com o Flexbox podemos posicionar os elementos de uma **página** e deixa-lá **flexível**.

Nesta aula veremos como **posicionar o Topo** de uma página.



AlfaTech

Soluções em hospedagem

INFORMAÇÕES PLANOS PREÇOS CONTATO

←-----1200px-----→



AlfaTech

Soluções em hospedagem

INFORMAÇÕES PLANOS PREÇOS CONTATO

←-----770px-----→

Terceiro exemplo prático

Conhecendo o exemplo

O terceiro exemplo é o **topo** de uma **página** HTML.



AlfaTech

Soluções em hospedagem

[INFORMAÇÕES](#) [PLANOS](#) [PREÇOS](#) [CONTATO](#)

Veja o código HTML do exemplo:

```
<body>
  <div class="container-topo">
    <div class="container-logo">
      <div class="container-logo-imagem">
        
      </div>
      <div>
        <h1>AlfaTech</h1>
        <p>Soluções em hospedagem</p>
      </div>
    </div>
    <div class="container-links">
      <a href="#informacoes">INFORMAÇÕES</a>
      <a href="#planos">PLANOS</a>
      <a href="#precos">PREÇOS</a>
      <a href="#contato">CONTATO</a>
    </div>
  </div>
</body>
```

index.html



Neste exemplo temos **um container (<div>)** que possui **dois subcontainers (<div>)**.

container-topo (<div>)



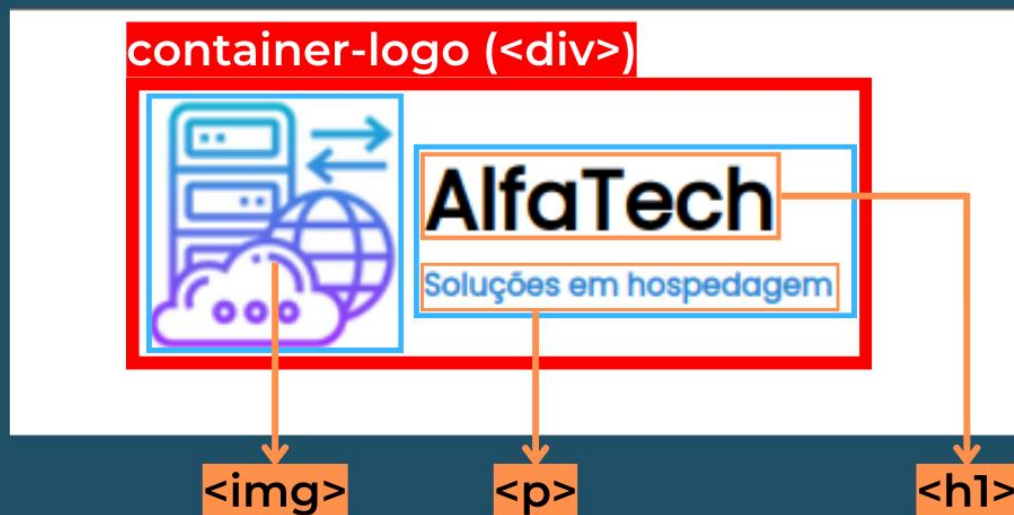
1º subcontainer (<div>)

INFORMAÇÕES PLANOS PREÇOS CONTATO

2º subcontainer (<div>)

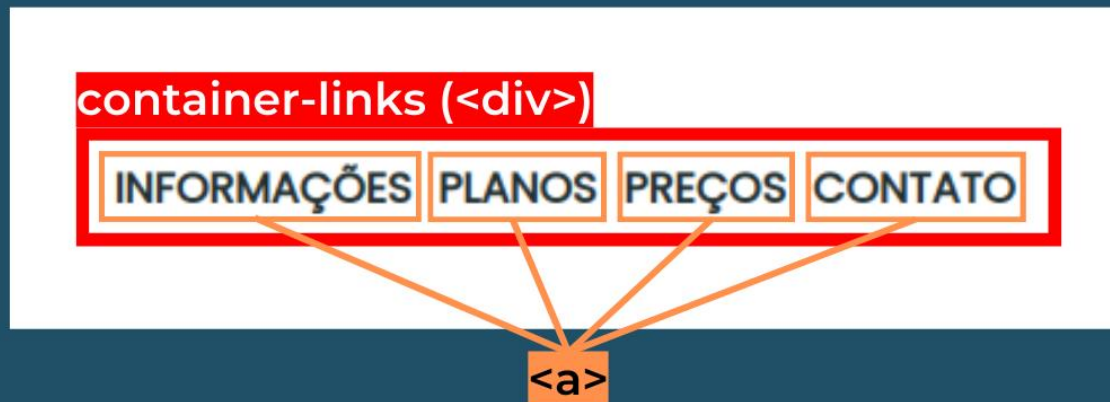
O 1º subcontainer terá **dois subcontainers** (<div>).

Um terá **uma imagem** () e o outro terá **um título** (<h1>) e **um parágrafo** (<p>).



```
<div class="container-logo">  
  <div class="container-logo-imagem">  
      
  </div>  
  <div>  
    <h1>AlfaTech</h1>  
    <p>Soluções em hospedagem</p>  
  </div>  
</div>
```

O 2º subcontainer terá **quatro links** (<a>).



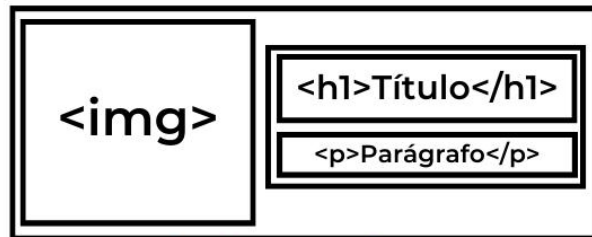
```
<div class="container-links">  
  <a href="#informacoes">INFORMAÇÕES</a>  
  <a href="#planos">PLANOS</a>  
  <a href="#precos">PREÇOS</a>  
  <a href="#contato">CONTATO</a>  
</div>
```

Os **subcontainers** devem começar a **quebrar linha** a partir de resoluções menores que **770px**.



Veja o mesmo exemplo em uma versão wireframe.

*Container-topo



*subcontainer

`<a>Link <a>Link <a>Link <a>Link`

*subcontainer

Neste exemplo o **Flexbox** será utilizado para **posicionar** os elementos do **Topo**.



AlfaTech

Soluções em hospedagem

INFORMAÇÕES PLANOS PREÇOS CONTATO

Exemplo sem Flexbox



AlfaTech

Soluções em hospedagem

INFORMAÇÕES PLANOS PREÇOS CONTATO

Exemplo com Flexbox

Terceiro exemplo prático

Entendendo o Flexbox do exemplo

Primeiro veja o exemplo **sem o Flexbox** aplicado.



AlfaTech

Soluções em hospedagem

INFORMAÇÕES PLANOS PREÇOS CONTATO

Aqui definimos uma **largura** para o
1º subcontainer (**275px**) e para o
2º subcontainer (**375px**).

```
.container-topo {  
  background-color: white;  
  padding: 35px 50px;  
}  
.container-logo {  
  max-width: 310px;  
}  
.container-links {  
  max-width: 350px;  
}  
.container-links a {  
  font-weight: 600;  
  text-decoration: none;  
  color: #233032;  
}  
.container-logo-imagem {  
  width: 80px;  
  height: 80px;  
}  
.container-logo-imagem img {  
  width: 100%;  
  height: 100%;  
}  
.container-logo p {  
  font-size: 12px;  
  font-weight: 600;  
  color: #2f86d6;  
}
```

Note que o container **não tem** uma **altura** pré-definida.



```
.container-topo {  
  background-color: white;  
  padding: 35px 50px;  
}
```

←-----100%-----→

Dessa forma, o container **assumirá** a **largura** **total** da tela.

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



[INFORMAÇÕES](#) [PLANOS](#) [PREÇOS](#) [CONTATO](#)

Para isso, começaremos aplicando o **Flexbox** no **container**.



AlfaTech

Soluções em hospedagem

INFORMAÇÕES PLANOS PREÇOS CONTATO

```
.container-topo {  
  background-color: white;  
  padding: 35px 50px;  
  display: flex;  
  justify-content: space-around;  
}
```

Os subcontainers ficarão em **linha** e **espaçados entre si** com uma margem.

Em seguida, aplicamos o Flexbox no 1º subcontainer.



AlfaTech
Soluções em hospedagem

INFORMAÇÕES PLANOS PREÇOS CONTATO

```
.container-logo {  
  width: 285px;  
  display: flex;  
  align-items: center;  
  justify-content: space-between;  
}
```

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

Depois, aplicamos o Flexbox no 2º subcontainer.



AlfaTech

Soluções em hospedagem

INFORMAÇÕES

PLANOS

PREÇOS

CONTATO

```
.container-links {  
  width: 385px;  
  display: flex;  
  align-items: center;  
  justify-content: space-between;  
}
```

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

Como próximo passo, definimos a **quebra de linha** do container.



```
.container-topo {  
  background-color: white;  
  padding: 35px 50px;  
  display: flex;  
  justify-content: space-around;  
  flex-wrap: wrap;  
  gap: 20px;  
}
```

←-----770px-----→

Para **espaçar os subcontainers** após a quebra de linha, aplicamos o **gap: 20px**.

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



```
.container-links {  
  width: 385px;  
  display: flex;  
  align-items: center;  
  justify-content: space-between;  
  flex-wrap: wrap;  
  gap: 10px;  
}
```

Para **espaçar os links** após a quebra de linha, aplicamos o **gap: 10px**.

Veja o resultado do exemplo com o Flexbox aplicado.



INFORMAÇÕES PLANOS PREÇOS CONTATO

```
.container-topo {
  background-color: white;
  padding: 35px 50px;
  display: flex;
  justify-content: space-around;
  flex-wrap: wrap;
  gap: 20px;
}

.container-logo {
  width: 285px;
  display: flex;
  align-items: center;
  justify-content: space-between;
}

.container-links {
  width: 385px;
  display: flex;
  align-items: center;
  justify-content: space-between;
  flex-wrap: wrap;
  gap: 10px;
}

.container-links a {
  font-weight: 600;
  text-decoration: none;
  color: #233032;
}

.container-logo-imagem {
  width: 100px;
  height: 100px;
}

.container-logo-imagem img {
  width: 100%;
  height: 100%;
}

.container-logo p {
  font-size: 12px;
  font-weight: 600;
  color: #2f86d6;
}
```