

Segundo e-book

Curso: Front-End do Bem

100% Gratuito | Sem pegadinhas

- Objetivo desse Segundo e-book
 - O que vamos desenvolver
 - Estrutura HTML do Site
 - Font Awesome (Ícones)

Buscando uma Fonte Externa (Google Fonts)

- Estilos CSS 3 do Site
- Estilos Responsivos do Site
- Créditos das Fotos Usadas no Site
 - Sugestões de Prática



/igor-rebolla

Conteúdo:

Objetivo Deste Segundo e-Book

Apresentação do que Será Desenvolvido

Estrutura de Arquivos e Pastas do Site

Buscando as Fotos as Quais Usaremos no Desenvolvimento do Site

1 - Estrutura Básica do Documento HTML do Site

2 - Código HTML do Site (NavBar)

- Explicação de Cada Elemento(Tag) do Código HTML do Site (NavBar)

3 - Código HTML do Site (Header)

- Explicação de Cada Elemento(Tag) do Código HTML do Site (Header)

4 - Código HTML do Site SECTION (Clients)

- Explicação de Cada Elemento(Tag) do Código HTML do Site (SECTION - Clients)

5 - Código HTML do Site SECTION (Products)

- Criando uma Conta no Font Awesome
- Buscando os Ícones no Font Awesome - SECTION (Products)
- Explicação de Cada Elemento(Tag) do Código HTML do Site (SECTION - Products)



/igor-rebolla

Conteúdo:

6 - Código HTML do Site SECTION (Services)

- Buscando os Ícones no Font Awesome - SECTION (Services)
- Explicação de Cada Elemento(Tag) do Código HTML do Site (SECTION - Services)

7 - Código HTML do Site SECTION (Depoimentos)

- Explicação de Cada Elemento(Tag) do Código HTML Site (SECTION - Depoimentos)

8 - Código HTML do Site SECTION (Photos)

- Explicação de Cada Elemento(Tag) do Código HTML do Site (SECTION - Photos)

9 - Código HTML do Site FOOTER (Rodapé)

- Explicação de Cada Elemento(Tag) do Código HTML do Site (FOOTER - Rodapé)

10 - Estrutura Básica do Arquivo Style.css

11 - Buscando uma Fonte Externa no Google Fonts

12 - Definindo alguns Resets do Arquivo Style.css



/igor-rebolla

Conteúdo:

13 - Aplicando Estilos - NAVBAR #navbar

- Explicação de Cada Propriedade do Estilo CSS aplicada na NAVBAR (1 #navbar)
- Explicação de Cada Propriedade do Estilo CSS aplicada na NAVBAR (2 .navbar-container)
- Explicação de Cada Propriedade do Estilo CSS aplicada na NAVBAR (3 .logo)
- Explicação de Cada Propriedade do Estilo CSS aplicada na NAVBAR (4 .navbar-items)
- Explicação de Cada Propriedade do Estilo CSS aplicada na NAVBAR (5 .navbar-items li)

14 - Aplicando Estilos - HEADER #banner

- Explicação de Cada Propriedade do Estilo CSS aplicada no HEADER (1 #banner)
- Explicação de Cada Propriedade do Estilo CSS aplicada no HEADER (2 .banner-container)
- Explicação de Cada Propriedade do Estilo CSS aplicada no HEADER (2 .banner-container h2)
- Explicação de Cada Propriedade do Estilo CSS aplicada no HEADER (2 .banner-container p)



/igor-rebolla

Conteúdo:

15 - Aplicando Estilos - SECTION #clients

- Explicação de Cada Propriedade do Estilo CSS aplicada na SECTION Clients (1 #clients)
- Explicação de Cada Propriedade do Estilo CSS aplicada na SECTION Clients (2 .headtext)
- Explicação de Cada Propriedade do Estilo CSS aplicada na SECTION Clients (3 .clients-container)
- Explicação de Cada Propriedade do Estilo CSS aplicada na SECTION Clients (4 .client)

16 - Aplicando Estilos - SECTION #products

- Explicação de Cada Propriedade do Estilo CSS aplicada na SECTION Products (1 #products)
- Explicação de Cada Propriedade do Estilo CSS aplicada na SECTION Products (2 .headtext)
- Explicação de Cada Propriedade do Estilo CSS aplicada na SECTION Products (3 .products-container)
- Explicação de Cada Propriedade do Estilo CSS aplicada na SECTION Products (4 .product)
- Explicação de Cada Propriedade do Estilo CSS aplicada na SECTION Products (5 .product-title)



/igor-rebolla

Conteúdo:

17 - Aplicando Estilos - SECTION Services (1 #services)

- Explicação de Cada Propriedade do Estilo CSS aplicada na SECTION Services (1 #services)
- Explicação de Cada Propriedade do Estilo CSS aplicada na SECTION Services (2 .headtext)
- Explicação de Cada Propriedade do Estilo CSS aplicada na SECTION Services (3 .services-container)
- Explicação de Cada Propriedade do Estilo CSS aplicada na SECTION Services (4 .services)
- Explicação de Cada Propriedade do Estilo CSS aplicada na SECTION Services (5 .services .service-item)

18 - Aplicando Estilos - SECTION Depoimentos #depoimentos

- Explicação de Cada Propriedade do Estilo CSS aplicada na SECTION Depoimentos (1 #depoimentos)
- Explicação de Cada Propriedade do Estilo CSS aplicada na SECTION Depoimentos (2 .headtext)
- Explicação de Cada Propriedade do Estilo CSS aplicada na SECTION Depoimentos (3 .depoimentos-container)
- Explicação de Cada Propriedade do Estilo CSS aplicada na SECTION Depoimentos (4 .depoimento)
- Explicação de Cada Propriedade do Estilo CSS aplicada na SECTION Depoimentos (5 .depoimento .depoimento-image)



/igor-rebolla

Conteúdo:

19 - Aplicando Estilos - SECTION Photos (#photos)

- Explicação de Cada Propriedade do Estilo CSS aplicada na SECTION Photos (1 #photos)
- Explicação de Cada Propriedade do Estilo CSS aplicada na SECTION Photos (2 .headtext)
- Explicação de Cada Propriedade do Estilo CSS aplicada na SECTION Photos (3 .photos-container)
- Explicação de Cada Propriedade do Estilo CSS aplicada na SECTION Photos (4 .photos-image)

20 - Aplicando Estilos - FOOTER (#rodape)

- Explicação de Cada Propriedade do Estilo CSS aplicada no FOOTER Rodapé (1 #rodape)

21 - Aplicando Estilos Responsivos ao Site

- Explicação de Cada Propriedade do Estilo Responsivo do Site

22 - Sugestões de prática para esse e-Book

23 - Links das 14 aulas do Curso Front-End do Bem

24 - Créditos das Fotos Utilizadas no Desenvolvimento do Site



/igor-rebolla

Objetivo Deste Segundo e-Book

Este Segundo e-Book tem o objetivo de consolidar os conhecimentos vistos nas 14 aulas do Curso Front-End do Bem com o desenvolvimento de um site (responsivo)

Alguns dos conhecimentos que iremos aplicar:

- Tags Semânticas
- Grid
- Display Flex
- Responsividade

Vamos ver como buscar ícones e fontes externas

E muito mais...

E tudo isso de uma forma leve e bem humorada... tanto que o projeto desenvolvido será um PESHOP do BEM HUMORADO o qual traremos um pouco de humor (sem perder o foco principal que é o aprendizado) durante o desenvolvimento.

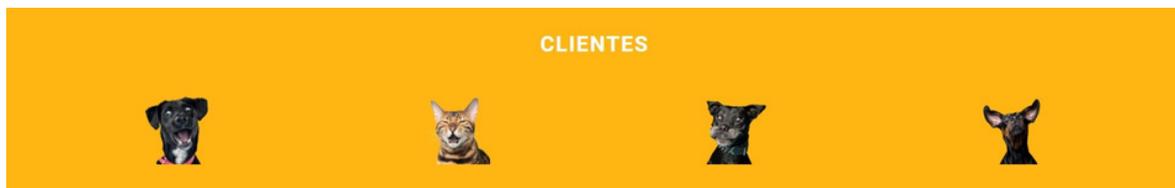
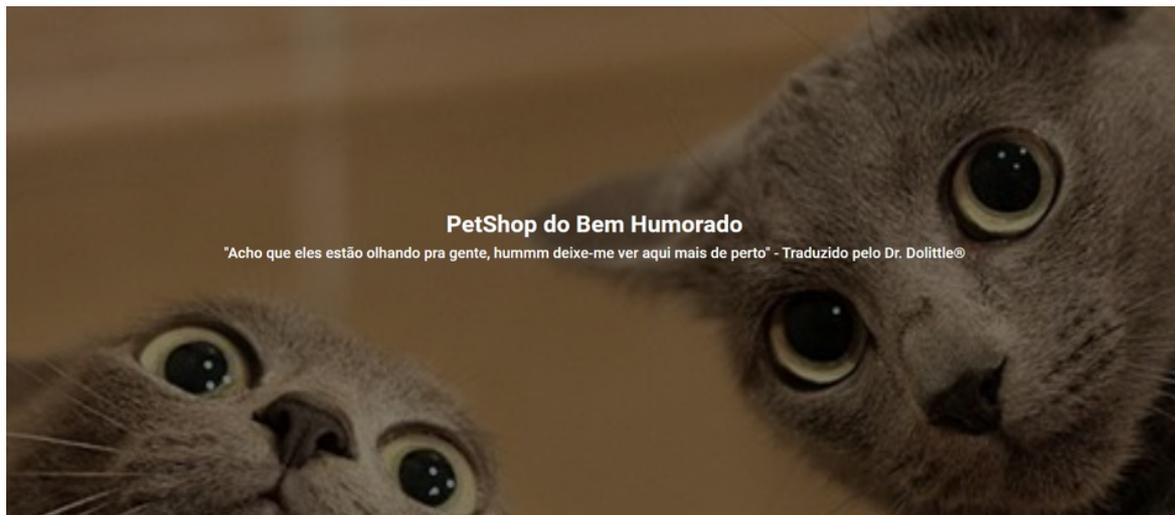
Este e-Book tem como finalidade ser educacional.

Os créditos das Fotos utilizadas estão referenciados em duas páginas no final desse e-Book



Apresentação do que Será Desenvolvido

É com grande alegria que lhes apresento as telas(layout) que desenvolveremos do PESHOP DO BEM HUMORADO



Apresentação do que Será Desenvolvido

SERVIÇOS



🐾 Levamos o seu Pet para passear. Ele vai gostar tanto de passear conosco que você terá que trazê-lo todos os dias.

📷 Temos fotógrafo especialista em Pets para tirar aquela foto bonita do seu amigo

🛁 Depois do passeio nada melhor do que aquele banho para deixar o seu pet limpinho e cheiroso.

🏠 Enquanto o seu pet (passela, toma banho e tira fotos) você pode aguardá-lo em nossa cãeseteria

DEPOIMENTOS



"Não gostei!! Me falaram que era passeio só que era Vacina" - Traduzido pelo Dr. Dolittle®



"Vim só pelo petisco - Quero dizer Meowwwwwwwwwwwwwwwww" - Traduzido pelo Dr. Dolittle®



/igor-rebolla

Apresentação do que Será Desenvolvido

FOTOS



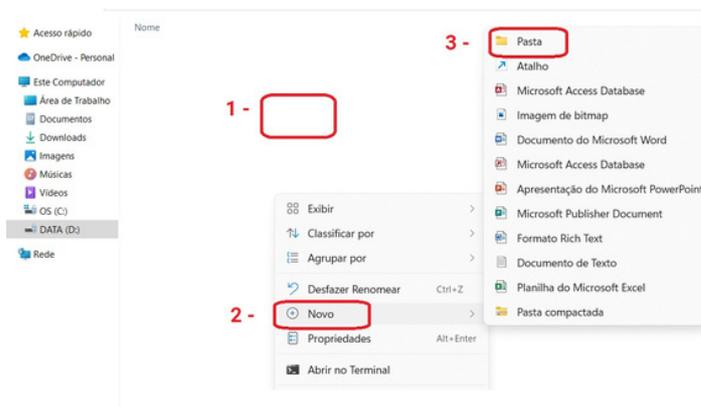
Copyright © 2022 - Todos os Direitos Reservados
Desenvolvido por um aluno do Curso FrontEnd do Bem

Estrutura de Arquivos e Pastas do Site

Vamos criar uma pasta (local onde os arquivos) do nosso site vão ficar alocados e assim poderemos entender de forma visual como a estrutura de pastas funciona dentro do Microsoft Visual Studio Code. Eu estou usando o Microsoft Windows 11, os passos aqui para criar uma pasta são:

- Abrir o Explorador de Arquivos ou Windows Explorer;
- Selecionar o Local onde a pasta do curso será criada (eu escolhi D:/);

- 1 - Clicar com o botão direito do mouse;
- 2 - Clicar em Novo;
- 3 - Clicar em Pasta.

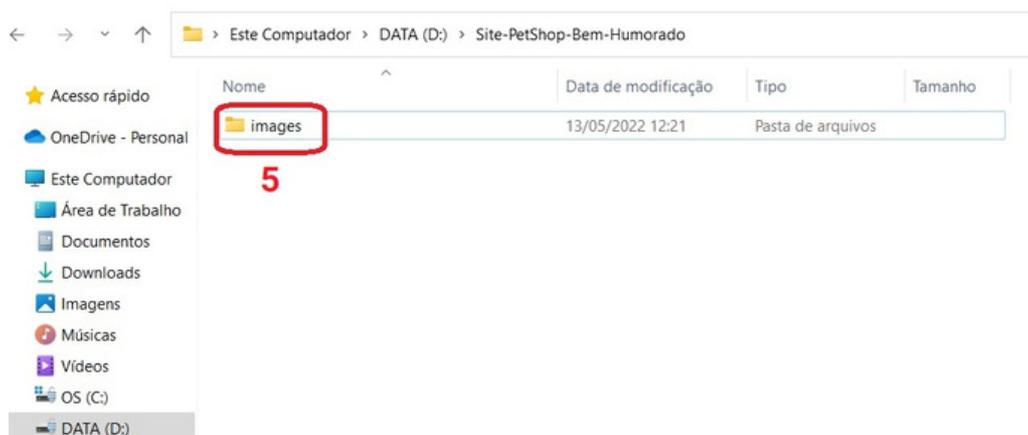


Estrutura de Arquivos e Pastas do Site

4 - Digitar o nome da Pasta - Aqui eu optei por: Site-PetShop-Bem-Humorado

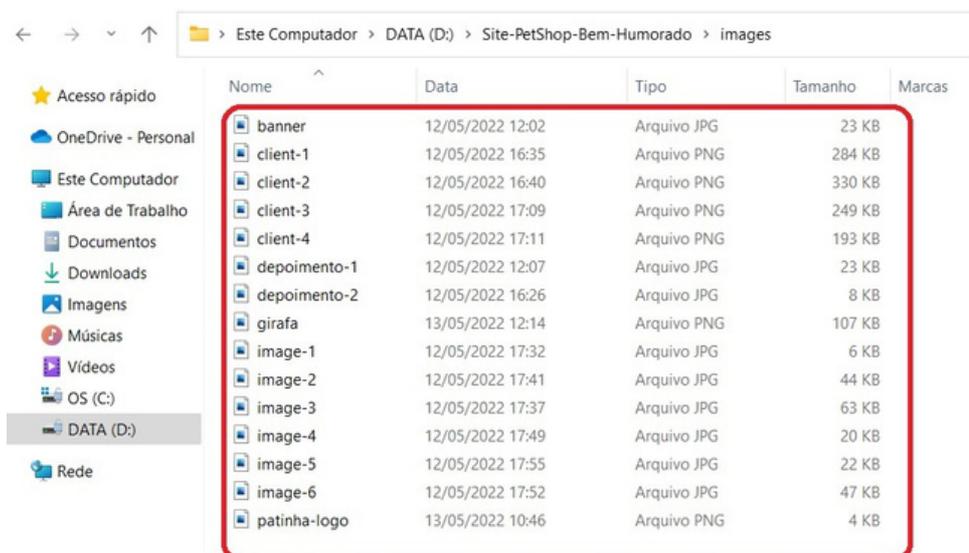


5 - Depois que criamos a Pasta Principal (Site-PetShop-Bem-Humorado) vamos dentro dessa pasta criar uma subpasta chamada images (local onde todas as imagens do no nosso site ficarão guardadas).



Estrutura de Arquivos e Pastas do Site

6 - E aqui temos as 15 imagens que iremos usar no desenvolvimento do site



Nome	Data	Tipo	Tamanho	Marcas
banner	12/05/2022 12:02	Arquivo JPG	23 KB	
client-1	12/05/2022 16:35	Arquivo PNG	284 KB	
client-2	12/05/2022 16:40	Arquivo PNG	330 KB	
client-3	12/05/2022 17:09	Arquivo PNG	249 KB	
client-4	12/05/2022 17:11	Arquivo PNG	193 KB	
depoimento-1	12/05/2022 12:07	Arquivo JPG	23 KB	
depoimento-2	12/05/2022 16:26	Arquivo JPG	8 KB	
girafa	13/05/2022 12:14	Arquivo PNG	107 KB	
image-1	12/05/2022 17:32	Arquivo JPG	6 KB	
image-2	12/05/2022 17:41	Arquivo JPG	44 KB	
image-3	12/05/2022 17:37	Arquivo JPG	63 KB	
image-4	12/05/2022 17:49	Arquivo JPG	20 KB	
image-5	12/05/2022 17:55	Arquivo JPG	22 KB	
image-6	12/05/2022 17:52	Arquivo JPG	47 KB	
patinha-logo	13/05/2022 10:46	Arquivo PNG	4 KB	

6 - Aqui temos as 15 imagens usadas no site

"Beleza Igor e onde eu faço o download dessas imagens?"

Eu vou disponibilizar o link do **github** para que vocês possam fazer o **git clone** das imagens para os Vossos computadores e assim colocar as 15 imagens dentro da pasta images.

"E porque você não disponibiliza por exemplo essas imagens em um Google Drive?"

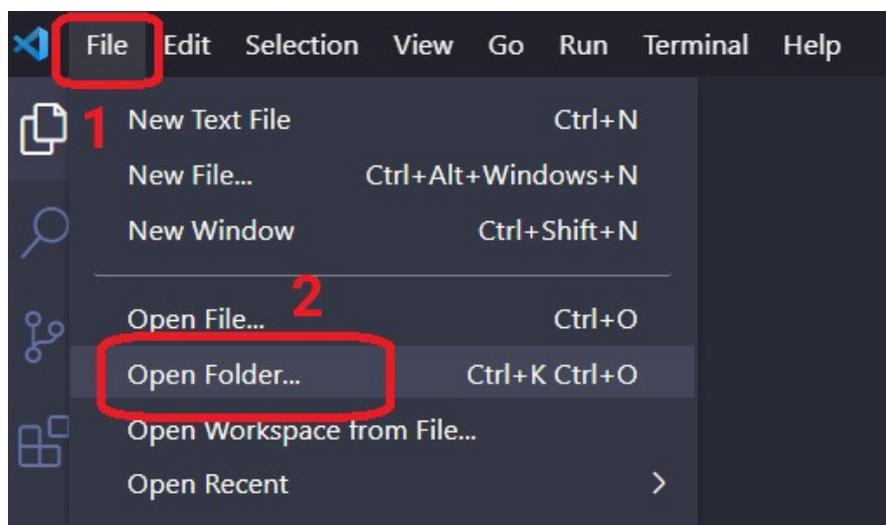
O objetivo aqui é simular ao máximo o dia a dia de um desenvolvedor e fazendo isso... eu estou forçando vocês a utilizarem as ferramentas que o mercado utiliza e quando mais vocês praticarem e tiverem o contato com essas ferramentas... mais rápido será o aprendizado.

Estrutura de Arquivos e Pastas do Site

Pasta Principal (Site-PetShop-Bem-Humorado) e subpasta images criadas, vamos abri-las dentro do Microsoft Visual Studio Code.

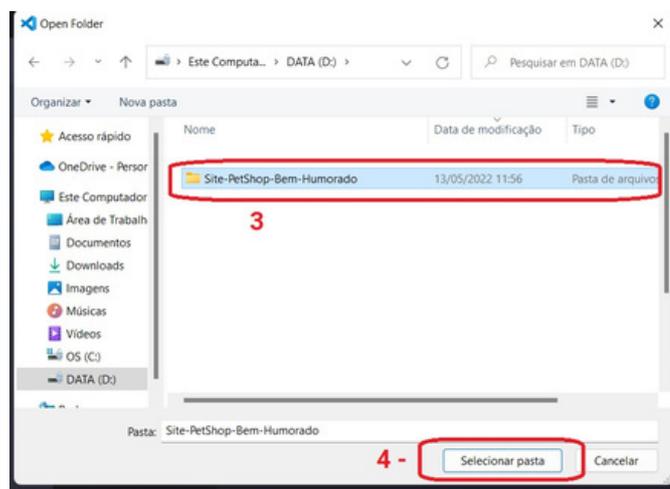
1 - Vamos clicar em File (Arquivo)

2 - Open Folder (Abrir Pasta)



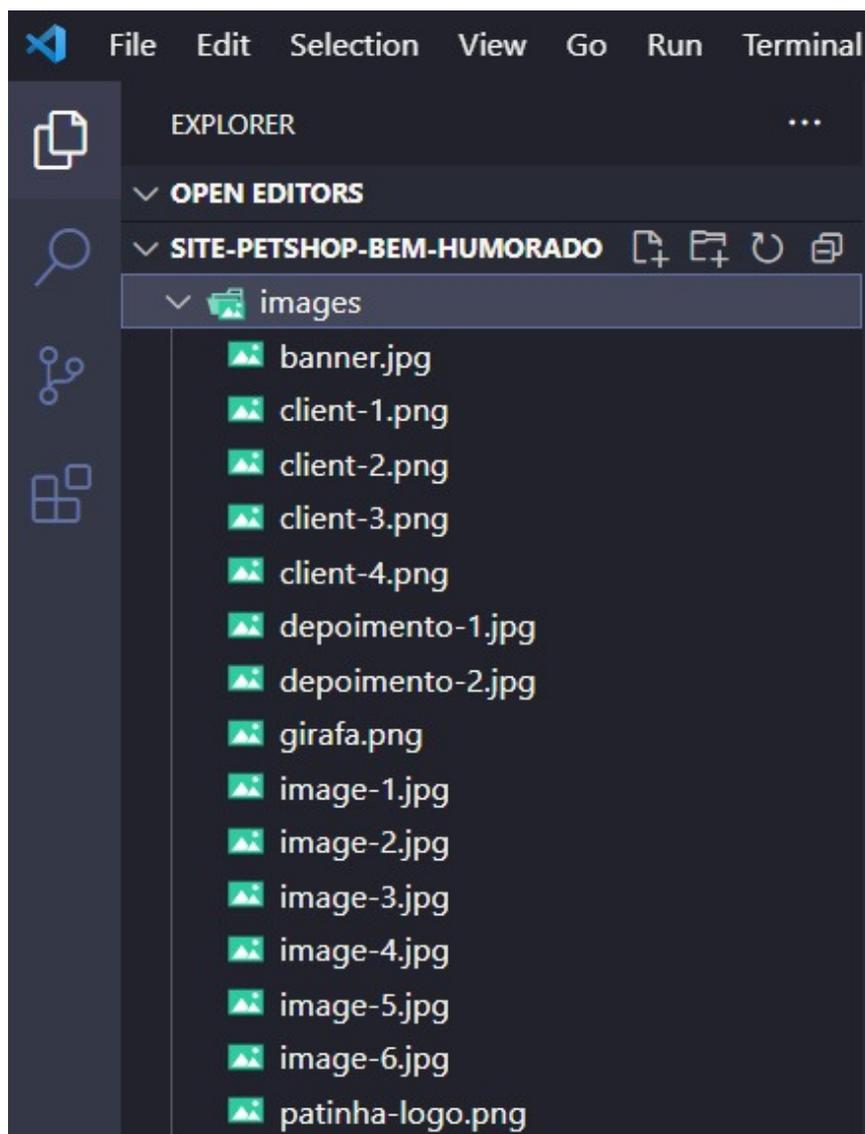
3 - Agora vamos selecionar a Pasta (Site-PetShop-Bem-Humorado)

4 - Vamos clicar em Selecionar Pasta



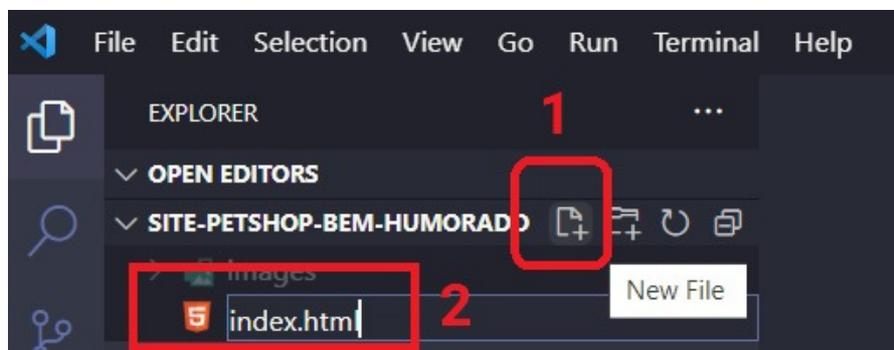
Estrutura de Arquivos e Pastas do Site

Pasta Principal junto com a subpasta images abertas no Microsoft Visual Studio Code.

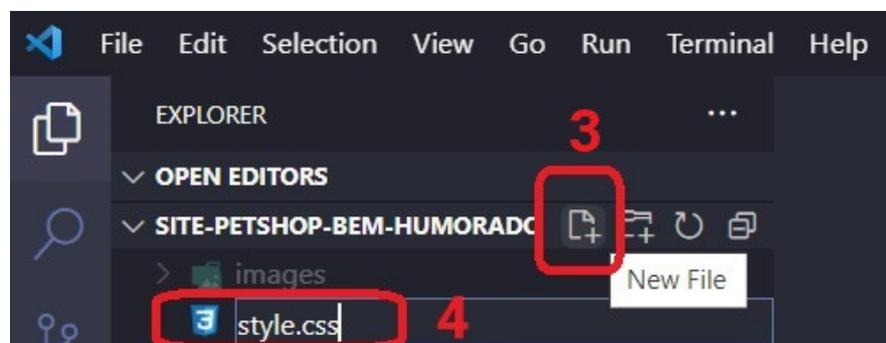


Estrutura de Arquivos e Pastas do Site

Pronto a Pasta Pet-Shop-Bem-Humorado está aberta do lado esquerdo da tela (o lado esquerdo é o local onde o Microsoft Visual Studio Code disponibiliza toda a estrutura de pastas e arquivos que iremos utilizar em nosso projeto. Vamos clicar no desenho do arquivo sinalizado pelo número 1 e criaremos o arquivo index.html do site.

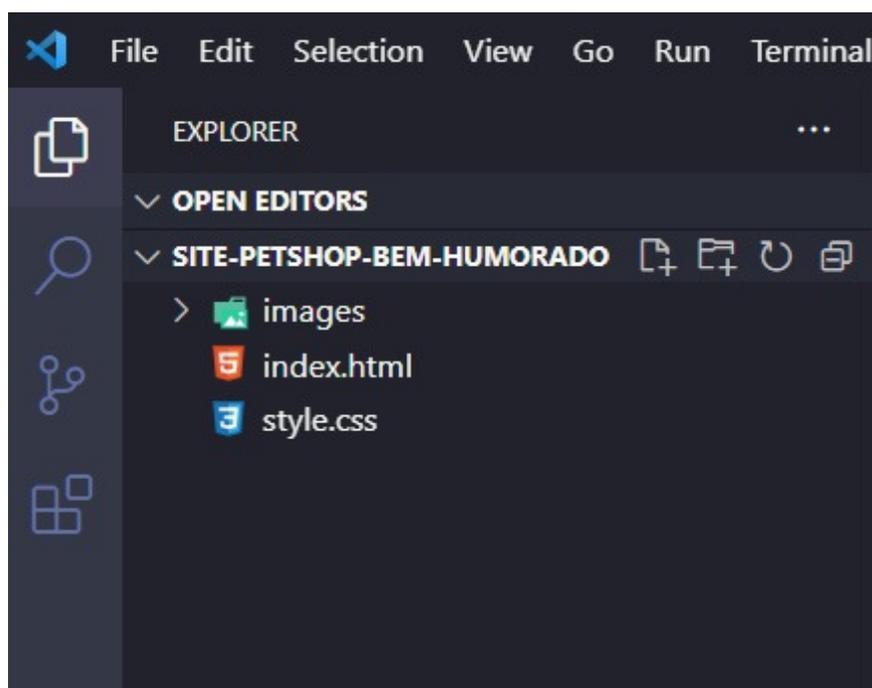


Vamos repetir o mesmo processo para criarmos o segundo arquivo do site o style.css (que será responsável pelos nossos estilos) Vamos clicar no desenho do arquivo sinalizado pelo número 3 e criaremos o arquivo style.css do site.



Estrutura de Arquivos e Pastas do Site

Agora temos toda a estrutura de pastas e arquivos criada dentro do Microsoft Visual Studio Code.



Na próxima página veremos como buscar as imagens que serão usadas no desenvolvimento do nosso site.

Buscando as Fotos as Quais Usaremos no Desenvolvimento do Site

Para buscar as imagens estou tomando como base que vocês já estão com o:

- 1 - Git instalado
- 2 - GitHub Instalado
- 3 - Perfil Criado no Github

Caso ainda não tenham feito esses 3 passos acima... vou deixar aqui abaixo o link da aula... a qual eu explico passo a passo como esses 3 passos são feitos. Link da aula (Totalmente gratuita):

https://www.linkedin.com/posts/igor-rebolla_aula11-activity-6897502324269662208-pSrE?utm_source=linkedin_share&utm_medium=member_desktop_web

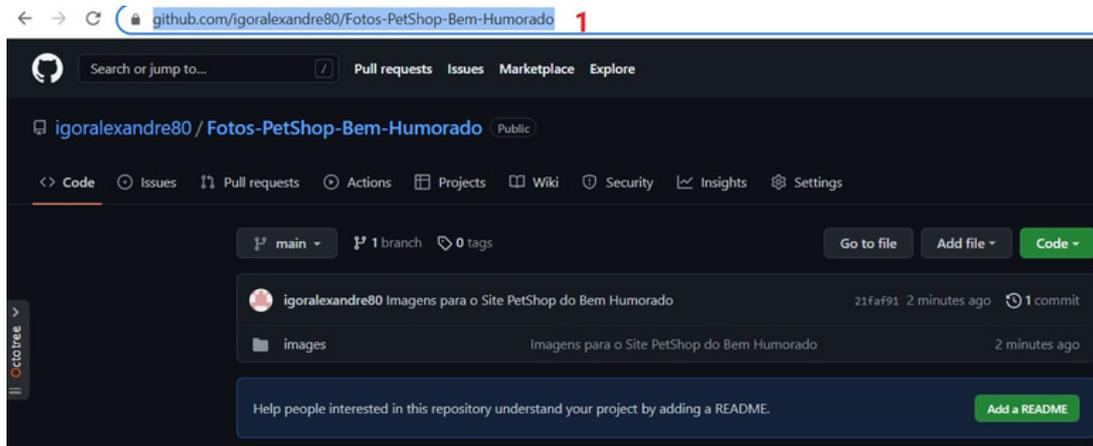
Agora que tá todo mundo com o Git e o GitHub (instalados) e o perfil dentro do Github criado. O primeiro passo que vocês vão fazer para buscar as imagens será:

- 1 - Clicar no link do github abaixo com o repositório das imagens:

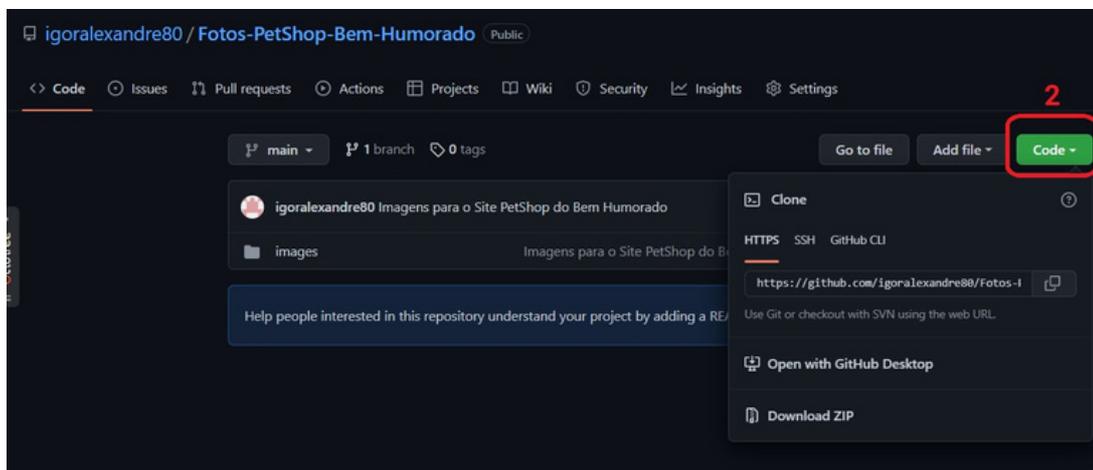
<https://github.com/igoralexandre80/Fotos-PetShop-Bem-Humorado>



Buscando as Fotos as Quais Usaremos no Desenvolvimento do Site

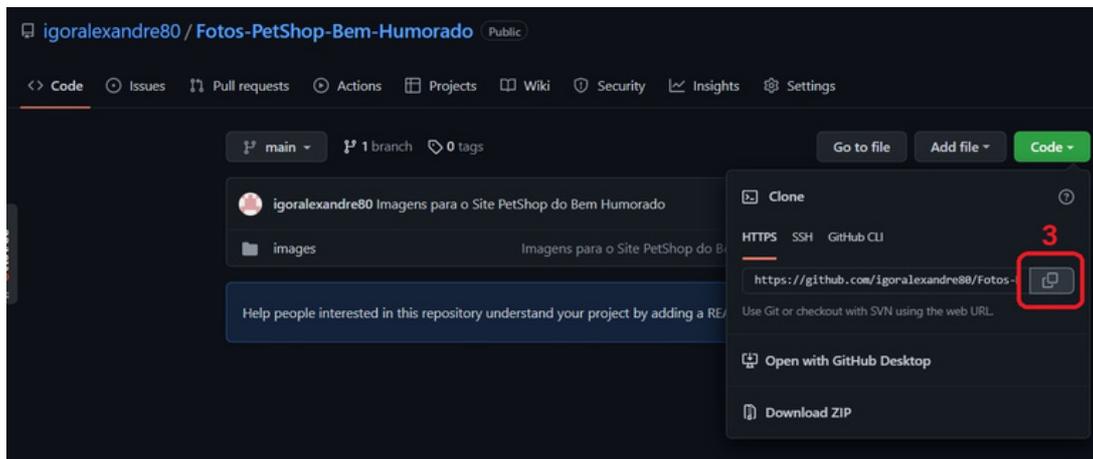


2 - Vamos clicar em Code



Buscando as Fotos as Quais Usaremos no Desenvolvimento do Site

3 - Vamos clicar no icone que representa 2 arquivos (assim copiaremos o link que está a sua esquerda)



4 - Agora vamos abrir o Prompt de Comando dentro do MS Windows (Selecionaremos um local para clonar esses arquivos para a nossa máquina local - Aqui eu escolhi o D:\)... e para clonarmos esses arquivos precisamos digitar git clone + (colar o link o qual acabamos de copiar dentro do GitHub). O comando ficará assim:

```
git clone https://github.com/igoralexandre80/Fotos-PetShop-Bem-Humorado.git
```

E daremos um ENTER



Buscando as Fotos as Quais Usaremos no Desenvolvimento do Site

```
cmd Prompt de Comando
D:\>git clone https://github.com/igoralexandre80/Fotos-PetShop-Bem-Humorado.git
Cloning into 'Fotos-PetShop-Bem-Humorado'...
remote: Enumerating objects: 18, done.
remote: Counting objects: 100% (18/18), done.
remote: Compressing objects: 100% (17/17), done.
remote: Total 18 (delta 0), reused 18 (delta 0), pack-reused 0
Receiving objects: 100% (18/18), 1.38 MiB | 6.40 MiB/s, done.
D:\>_
```

Imagens devida "clonadas" e baixadas para o computador local, agora é só pegar essas imagens (copia-las e cola-las) dentro da pasta do images do (Site-PetShop-Bem-Humorado).

Caso algum passo do Git Clone não tenha ficado claro, vou deixar aqui abaixo o link da aula a qual eu explico com mais detalhes o funcionamento do git clone. Link da aula (Totalmente gratuita):

https://www.linkedin.com/posts/igor-rebolla_aula12-activity-6900039147038617600-9dHv?utm_source=linkedin_share&utm_medium=member_desktop_web



Estrutura Básica do Documento HTML do Site

Antes de começarmos a colocar os códigos do nosso site dentro do arquivo index.html, precisamos definir a sua estrutura básica que é formada da seguinte maneira:

```
<!DOCTYPE html>
<html lang="pt-br">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>

<body>

  <!-- AQUI DENTRO É O LOCAL ONDE TODO O CÓDIGO HTML DO
  NOSSO SITE "PETSHOP DO BEM HUMORADO" SERÁ INSERIDO -->

</body>

</html>
```

Obs.: Notem que dentro da Tag <body> foi inserido um texto aqui representado pela cor vermelha entre essa estrutura **<!-- Texto -->**. Essa estrutura significa que inserimos um comentário para identificar onde começa e termina por exemplo uma seção a qual estamos desenvolvendo dentro do arquivo index.html. **O comentário não é renderizado no navegador.**



Estrutura Básica do Documento HTML do Site

A estrutura básica do Documento HTML dentro do Microsoft Visual Studio Code ficará conforme imagem abaixo (com o comentário inserido)

```
index.html X
1 <!DOCTYPE html>
2 <html lang="pt-br">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Document</title>
8 </head>
9
10 <body>
11
12   <!-- AQUI DENTRO É O LOCAL ONDE TODO O CÓDIGO HTML DO NOSSO SITE SERÁ INSERIDO -->
13
14 </body>
15
16 </html>
```

Na imagem abaixo podemos ver o resultado dessa estrutura básica com o comentário renderizado no navegador.



*** Conforme resultado esperado: Nada foi mostrado(renderizado) no navegador (nem mesmo o comentário).**

Estrutura Básica do Documento HTML do Site

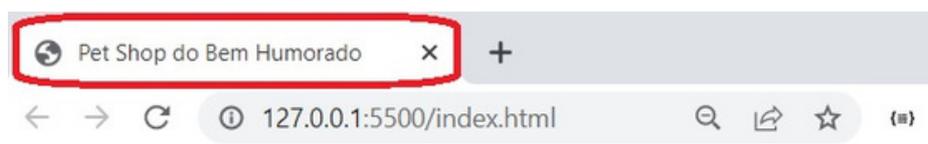
Dentro da estrutura básica precisamos alterar o Title (Título) O elemento `<title>` serve para definir o título de sua página. Portanto, é aqui que se define o título que é mostrado na guia do navegador (Google Chrome).

`<title>Pet Shop do Bem Humorado</title>`.

O `<title>Pet Shop do Bem Humorado</title>` dentro do Visual Studio Code ficará assim:

```
index.html x
1 <!DOCTYPE html>
2 <html lang="pt-br">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Pet Shop do Bem Humorado</title>
8 </head>
9
10 <body>
11
12   <!-- AQUI DENTRO É O LOCAL ONDE TODO O CÓDIGO HTML DO NOSSO SITE SERÁ INSERIDO -->
13
14 </body>
15
16 </html>
```

Na imagem abaixo podemos ver o resultado do `<title></title>` sendo mostrado na guia do navegador (Google Chrome)



Código HTML do Site (NavBar)

Aqui é onde vamos iniciar toda a estrutura de códigos HTML que compõem a NAVBAR. O objetivo final depois de estilizarmos (style.css) a NAVBAR é esse abaixo:



```
<body>

<!-- INICIO DO CÓDIGO HTML - NAVBAR -->
<nav id="navbar">
  <div class="navbar-container">
    <h1 class="logo">P  t Shop do Bem Humorado</h1>
    <ul class="navbar-items">
      <li><a href="#banner">Inicio</a></li>
      <li><a href="#clients">Clientes</a></li>
      <li><a href="#photos">Fotos</a></li>
    </ul>
  </div>
</nav>
</body>
</html>

<!-- FINAL DO CÓDIGO HTML - NAVBAR -->
```

Na imagem abaixo podemos ver o resultado do <title></title> sendo mostrado na guia do navegador (Google Chrome)



Explicação de Cada Elemento(Tag) do Código HTML do Site (NavBar)

```
<!-- INICIO DO CÓDIGO HTML - NAVBAR -->
```

Comentário indicando onde é o início do Código HTML da NAVBAR (Só reforçando... esse comentário não é renderizado pelo navegador.

```
<nav id="navbar">
```

Aqui criamos uma nav com o id que recebe "navbar". A nav terá um identificador(id) único e a "navbar" será utilizada para estilizarmos essa nav quando formos trabalhar dentro do arquivo style.css

```
<div class="navbar-container">
```

Dentro da navbar nós temos um container, esse container ficará responsável por agrupar os itens dentro do navbar-container.

```
<h1 class="logo">P  t Shop do Bem Humorado</h1>
```

Dentro dessa div nós teremos um <h1> com uma class "logo". O nome aqui foi definido como Pet Shop do Bem Humorado e a letra 'e' da palavra Pet foi substituída por uma imagem de uma patinha.

```
<ul class="navbar-items">
```

Aqui vamos definir uma ul(lista) com um class "navbar-items" e dentro dessa ul teremos três

continua na próxima página...



/igor-rebolla

Explicação de Cada Elemento(Tag) do Código HTML do Site (NavBar)

... continuação da página anterior

```
<li><a href="#banner">Inicio</a></li>
```

```
<li><a href="#clients">Clientes</a></li>
```

```
<li><a href="#photos">Fotos</a></li>
```

Aqui definimos 3 (li's - Inicio, Clientes e Fotos), onde cada li foi colocada dentro de uma tag <a> com um atributo href e dentro desse atributo href definimos cada um com uma # (#banner, #clients, #photos) tudo isso significa que ao clicarmos em Clientes na nossa NAVBAR automaticamente ele vai para Section (Seção) Clientes do nosso site e o mesmo acontece ao clicarmos em Inicio ele volta automaticamente para o topo da página.

```
</ul>
```

Tag de fechamento dessa ul aqui <ul class="navbar-items">

```
</div>
```

Tag de fechamento dessa div aqui <div class="navbar-container">

```
</nav>
```

Tag de fechamento dessa nav aqui <nav id="navbar">

```
</body>
```

Tag de fechamento desse <body>

```
</html>
```

Tag de fechamento de toda estrutura HTML



/igor-rebolla

Explicação de Cada Elemento(Tag) do Código HTML do Site (NavBar)

<!-- FINAL DO CÓDIGO HTML - NAVBAR -->

Comentário indicando onde é o final do Código HTML da NAVBAR (Só reforçando... esse comentário não é renderizado pelo navegador.

Agora vamos ver como toda essa estrutura do NAVBAR ficará dentro do VsCode.

```
index.html X
1 <!DOCTYPE html>
2 <html lang="pt-br">
3
4 <head>
5 <meta charset="UTF-8">
6 <meta name="viewport" content="width=device-width, initial-scale=1.0">
7 <title>Pet Shop do Bem Humorado</title>
8 </head>
9
10 <body>
11
12 <body>
13 <!-- INICIO DO CÓDIGO HTML - NAVBAR -->
14 <nav id="navbar">
15 <div class="navbar-container">
16 <h1 class="logo">P  t Shop do Bem Humorado</h1>
17 <ul class="navbar-items">
18 <li><a href="#banner">Inicio</a></li>
19 <li><a href="#clients">Clientes</a></li>
20 <li><a href="#photos">Fotos</a></li>
21 </ul>
22 </div>
23 </nav>
24 <!-- FINAL DO CÓDIGO HTML - NAVBAR -->
25
26 </body>
27
28 </html>
```

Na imagem da próxima página podemos ver o resultado do estrutura da NAVBAR sendo renderizada (mostrada) no navegador (Google Chrome)

Explicação de Cada Elemento(Tag) do Código HTML do Site (NavBar)

← → ↻ 127.0.0.1:5500/index.html

 t Shop do Bem Humorado

- [Inicio](#)
- [Clientes](#)
- [Fotos](#)

Como nós estruturamos "apenas" o código HTML até o momento sem nenhum estilo (style.css) o resultado é esse acima (não muito simpático).

Por esse ser um e-book didático (eu optei por estruturar todo o HTML antes de partirmos para as estilizações)

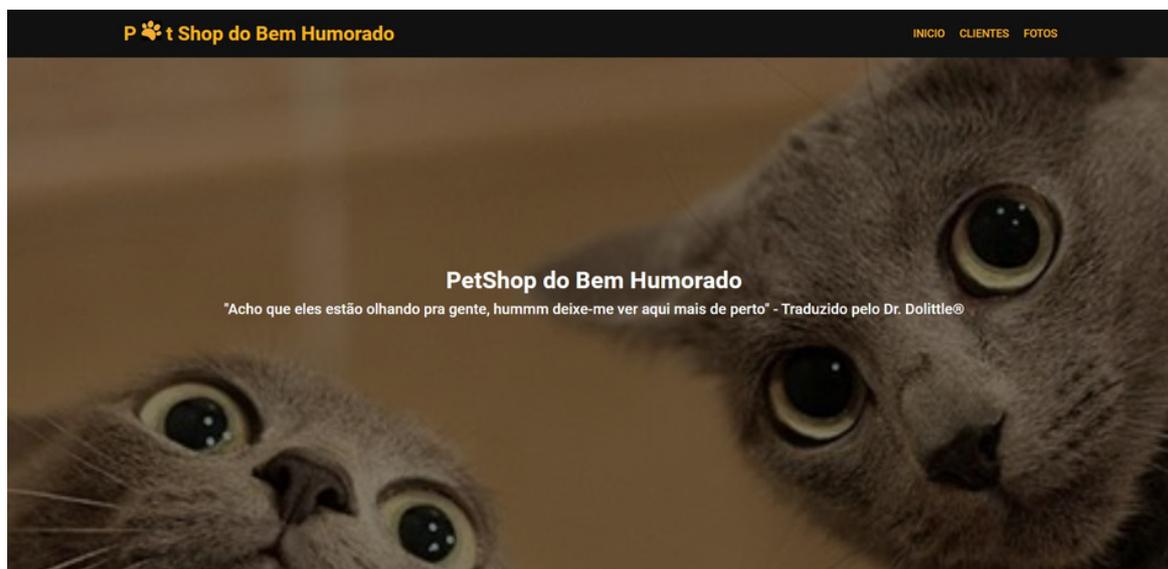
Então sabendo disso e como já estruturamos toda a NAVBAR do nosso site, vamos partir para a segunda estrutura do nosso site que será:

2° - HEADER (BANNER + Parágrafos)

Então bora ver esse tal de HEADER já na próxima página....

Código HTML do Site (Header)

Aqui é onde vamos iniciar toda a estrutura de códigos HTML que compõem o HEADER. O objetivo final depois de estilizarmos (style.css) a HEADER é esse abaixo:



Notem que acima a Navbar também está sendo mostrada (Sim, e está correto) foi definido dentro do style.css que a Navbar está com position (fixed) ou seja ela vai aparecer em todas as etapas (Até no footer) que é a última estrutura HTML a qual iremos ver aqui.

Não se preocupem com esse position fixed agora (Veremos no momento de estilizar o site) eu comentei aqui para que fique transparente o porque da NAVBAR estar aparecendo aqui.

Código HTML do Site (Header)

O início do código do Header será feito logo abaixo do fechamento do comentário da NAVBAR.

```
<!-- FINAL DO CÓDIGO HTML - NAVBAR -->

<!-- INICIO DO CÓDIGO HTML - HEADER (Banner) -->
<header id="banner">
  <div class="banner-container">
    <h2>PetShop do Bem Humorado</h2>
    <p>"Acho que eles estão olhando pra gente, hummm deixe-me ver aqui mais de perto" - Traduzido pelo Dr. Dolittle®</p>
  </div>
</header>
<!-- FINAL DO CÓDIGO HTML - HEADER (Banner) -->
```



/igor-rebolla

Explicação de Cada Elemento(Tag) do Código HTML do Site (Header)

```
<!-- INICIO DO CÓDIGO HTML - HEADER (Banner) -->
```

Comentário indicando onde é o início do Código HTML do HEADER (Só reforçando... esse comentário não é renderizado pelo navegador).

```
<header id="banner">
```

Aqui criamos um header com o id que recebe "banner". O header terá um identificador(id) único e o "banner" será utilizada para estilizarmos esse header quando formos trabalhar dentro do arquivo style.css

```
<div class="banner-container">
```

Dentro da div nós temos um container, esse container ficará responsável por agrupar os itens dentro do banner-container.

```
<h2>PetShop do Bem Humorado</h2>
```

```
<p>"Acho que eles estão olhando pra gente, hummm deixe-me ver aqui mais de perto" - Traduzido pelo Dr. Dolittle®</p>
```

Dentro dessa div nós teremos um `<h2>` e um `<p>` com os respectivos textos acima.

```
</div>
```

Tag de fechamento dessa div aqui `<div class="banner-container">`

```
</header>
```

Tag de fechamento desse header aqui `<header id="banner">`



/igor-rebolla

Explicação de Cada Elemento(Tag) do Código HTML do Site (Header)

<!-- FINAL DO CÓDIGO HTML - HEADER (Banner) -->

Comentário indicando onde é o final do Código HTML do HEADER (Só reforçando... esse comentário não é renderizado pelo navegador).

Agora vamos ver como toda essa estrutura do HEADER ficará dentro do VsCode.

```
<!-- FINAL DO CÓDIGO HTML - NAVBAR -->

<!-- INICIO DO CÓDIGO HTML - HEADER (Banner) -->
<header id="banner">
  <div class="banner-container">
    <h2>PetShop do Bem Humorado</h2>
    <p>"Acho que eles estão olhando pra gente, hummm deixe-me ver aqui mais de perto" - Traduzido pelo Dr. Dolittle*</p>
  </div>
</header>
<!-- FINAL DO CÓDIGO HTML - HEADER (Banner) -->
```

Na imagem abaixo podemos ver o resultado do estrutura do HEADER sendo renderizado(mostrado) no navegador (Google Chrome)



Explicação de Cada Elemento(Tag) do Código HTML do Site (Header)

Na figura anterior - Notem que o Header aparece logo abaixo da NavBar dentro do Navegador (isso ocorre devido a estrutura a qual definimos dentro do arquivo index.html).

Tanto que a próxima estrutura que vamos desenvolver será dos CLIENTES e essa estrutura ficará logo abaixo do Header e quando for renderizada no navegador seguirá essa mesma ordem e será mostrada abaixo do Header.

Então sabendo disso e como já estruturamos a NAVBAR e o HEADER do nosso site, vamos partir para a terceira estrutura do nosso site que será:

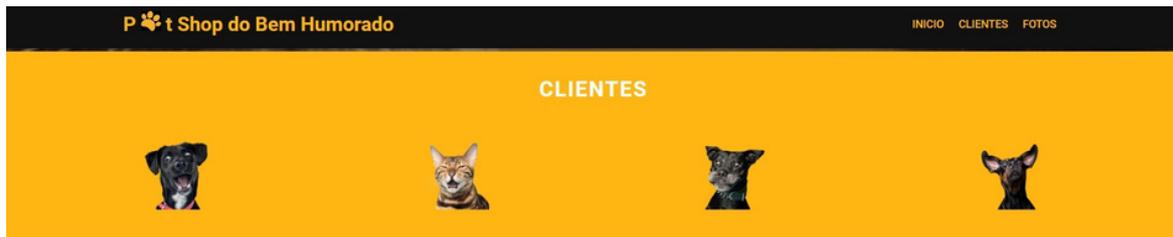
3° - SECTION (CLIENTS)

Então bora ver esse tal de SECTION (Clients) já na próxima página....



Código HTML do Site SECTION (Clients)

Aqui é onde vamos iniciar toda a estrutura de códigos HTML que compõem a SECTION (Clients). O objetivo final depois de estilizarmos (style.css) a Section (Clients) é esse abaixo:



Código HTML do Site SECTION (Clients)

O início do código da SECTION (Clients) será feito logo abaixo do fechamento do comentário do HEADER (Banner).

```
<!-- FINAL DO CÓDIGO HTML - HEADER (Banner) -->  
  
<!-- INICIO DO CÓDIGO HTML - SECTION (Clients) -->  
<section id="clients">  
  <div class="headtext">Clientes</div>  
  <div class="clients-container">  
      
      
      
      
  </div>  
</section>  
<!-- FINAL DO CÓDIGO HTML - SECTION (Clients) -->
```



/igor-rebolla

Explicação de Cada Elemento(Tag) do Código HTML do Site (SECTION - Clients)

`</div>`

Tag de fechamento dessa div aqui `<div class="clients-container">`

`</section>`

Tag de fechamento dessa section aqui `<section id="clients">`

`<!--FINAL DO CÓDIGO HTML - SECTION (Clients) -->`

Comentário indicando onde é o final do Código HTML da SECTION Clients (Só reforçando... esse comentário não é renderizado pelo navegador).

Agora vamos ver como toda essa estrutura da SECTION (Clients) ficará dentro do VsCode.

```
<!-- FINAL DO CÓDIGO HTML - HEADER (Banner) -->

<!-- INICIO DO CÓDIGO HTML - SECTION (Clients) -->
<section id="clients">
  <div class="headtext">Clientes</div>
  <div class="clients-container">
    
    
    
    
  </div>
</section>
<!-- FINAL DO CÓDIGO HTML - SECTION (Clients) -->
```

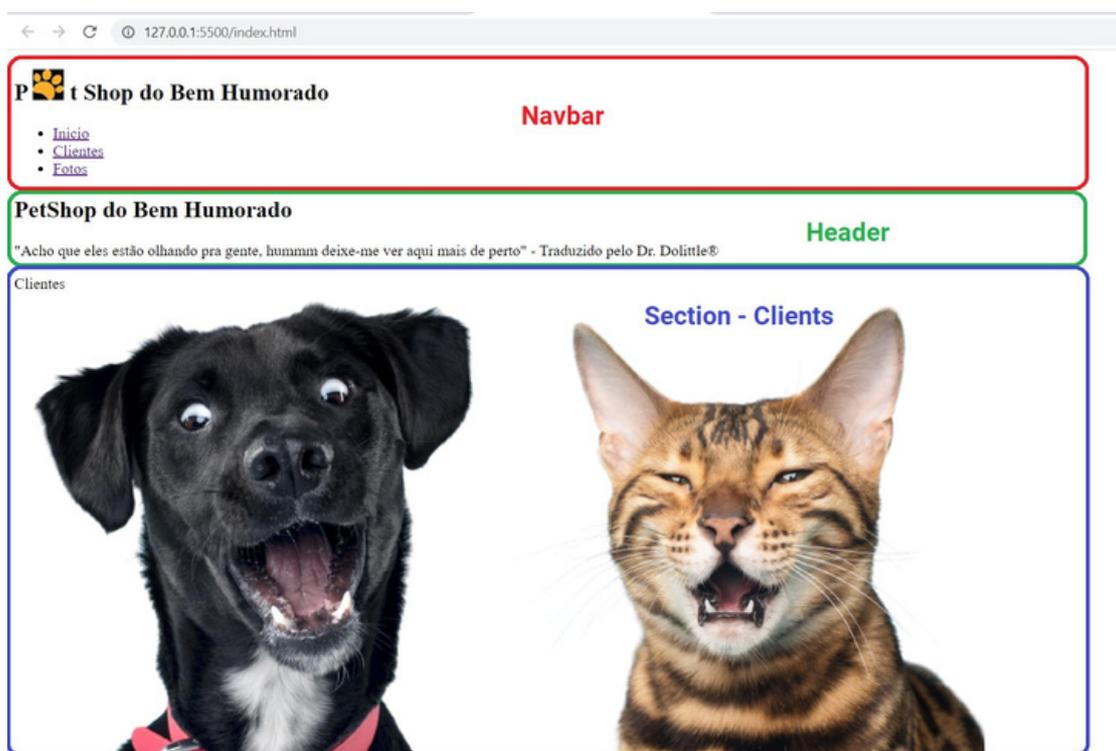


[/igor-rebolla](#)

Explicação de Cada Elemento(Tag) do Código HTML do Site (SECTION - Clients)

Nas 2 próximas imagens podemos ver o resultado da estrutura da SECTION - Clients sendo renderizada (mostrado) no navegador (Google Chrome)

Imagem 1:



Como nós inserimos 4 imagens dentro da estrutura do index.html e ainda não definimos nenhum estilo para essas imagens, quando elas são renderizadas pelo navegador e elas não tem mais espaço naquela elas automaticamente procuram "mais espaço" onde possam se encaixar.

Explicação de Cada Elemento(Tag) do Código HTML do Site (SECTION - Clients)

Imagem 2:



Até aqui estruturamos a NAVBAR, o HEADER e a Section Clients do nosso site, vamos partir para a quarta estrutura do nosso site que será:

4° - SECTION (PRODUTOS)

Então bora ver esse tal de SECTION (Produtos) já na próxima página....

Código HTML do Site SECTION (Products)

Aqui é onde vamos iniciar toda a estrutura de códigos HTML que compõem a SECTION (Products). O objetivo final depois de estilizarmos (style.css) a Section (Products) é esse abaixo:



Notem que na imagem acima existem 3 ícones (ossinho, bolinha e remédios) esses ícones eu busquei de um SITE EXTERNO chamado FONT AWESOME. O objetivo de colocar esses ícones aqui é mostrar para vocês como buscar elementos externos que facilitam as coisas quando estamos desenvolvendo o(s) nosso(s) site(s).

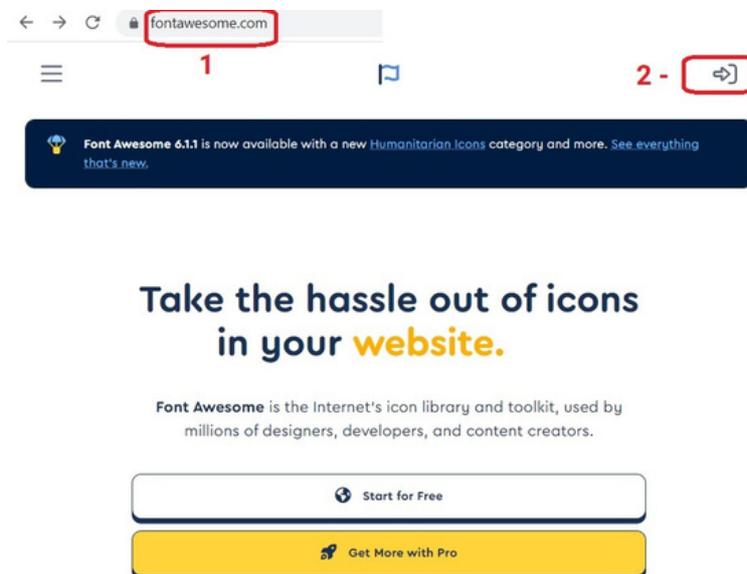
OBS.: O FONT AWESOME tem a sua versão gratuita que é a qual nós vamos utilizar aqui e supre bem as necessidades do nosso dia a dia.

Como acessar o Font Awesome para termos acesso aos ícones do nosso site:

- 1 - Digitar no navegador: <https://fontawesome.com/>
- 2 - Clicar em Sign In

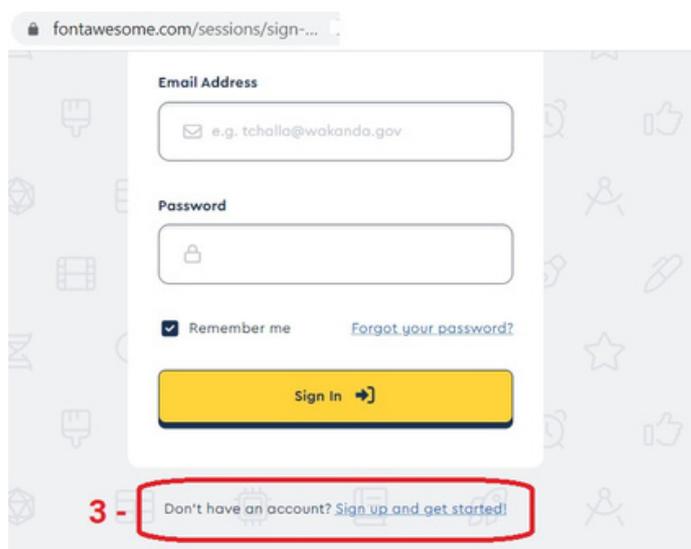
Podemos entender melhor os Passos 1 e 2 - Na imagem da próxima página.

Criando uma Conta no Font Awesome



Agora é necessário efetuamos um cadastro (só reforçando: é gratuito).

3 - Rolar a página até encontrar: Don't have an account? Sign up and get started! (Não tem uma conta? Assine e vamos iniciar!)



Criando uma Conta no Font Awesome

- 4 - Colocar o seu e-mail para receber o kit do Font Awesome
- 5 - Clicar em Send Kit Code (Enviar o código do Kit)

Get Started with Font Awesome

The easiest way to get icons on your website is with a Kit. It's your very own custom version of Font Awesome, all bundled up with only the icons, tools, and settings you need.



Enter your email to get started with a free Kit! ?

4

5

Send Kit Code 📧

Powered by Cloudflare

Already have an account? Sign In →

- 6 - Você vai receber uma mensagem Check Your Email (Verifique o seu Email)

6

Check Your Email

We've just sent a link to Verify your address and we'll get you all set up!

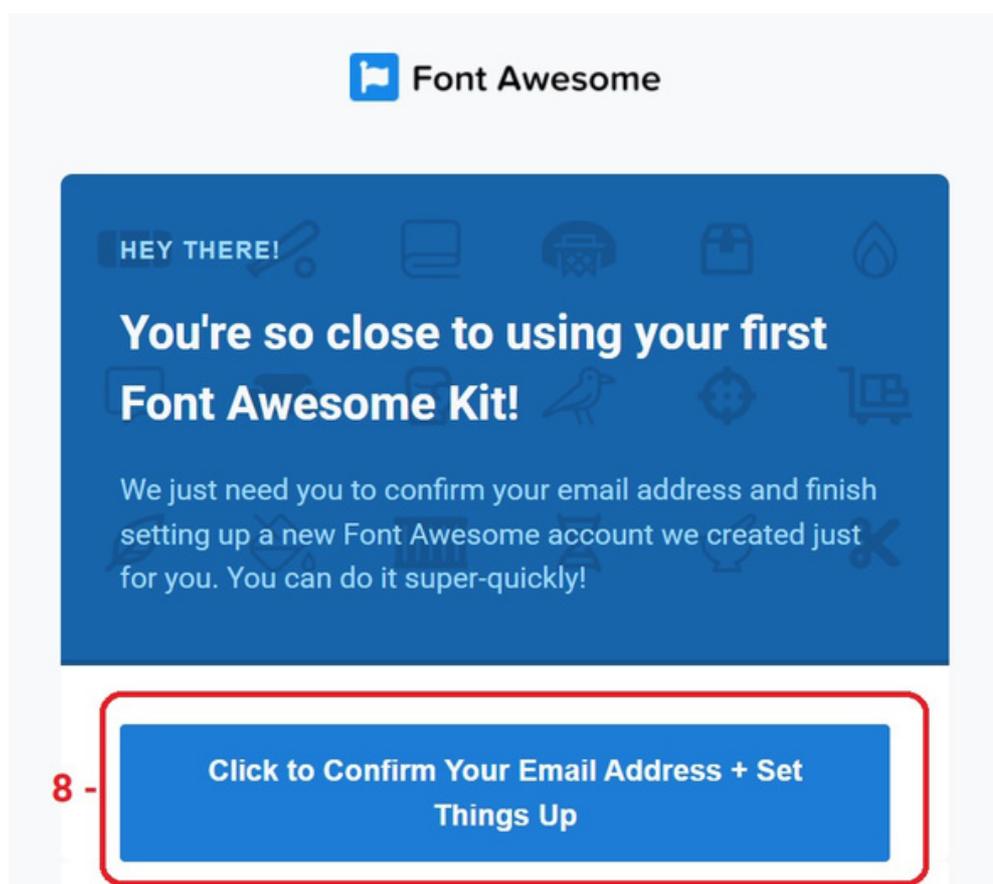
Resend Confirmation Email →

Criando uma Conta no Font Awesome

7 - Clicar 2x no e-mail recebido (Font Awesome)



8 - Clicar em Click to Confirm Your Email Address + Set Things Up (Clique para confirmar o seu endereço de e-mail + definir alguns ajustes).



Criando uma Conta no Font Awesome

9 - No campo New Password (Nova Senha) digitar uma senha de Vossa Preferência

10 - No campo Confirm New Password (Confirmar Nova Senha) repetir a mesma senha digitada no passo 9 (acima).

11 - Clicar em Set Password e Continue (Definir Senha e Continuar)

LET'S FINISH SETTING UPI



Choose a Password

Hi Choose a password for your new Font Awesome account.

New Password

9

Make it a good one!

Confirm New Password

10

Make it match the good one!

11

Set Password & Continue →

Criando uma Conta no Font Awesome

12 - No formulário Tell Us About Yourself (Fale sobre você) não precisa colocar as informações solicitadas para dar sequência no processo de cadastro, sabendo disso rolar a página do navegador e clicar em No, thanks. Let's skip this step for now (Não obrigado. Vamos pular essa etapa por agora).

LET'S FINISH SETTING UP!



Tell Us About Yourself

Mind sharing a few personal bits,
allexandreigor@gmail.com?

What's your first name?

How about your last name?

You first used Font Awesome in...

Got a favorite Font Awesome icon?

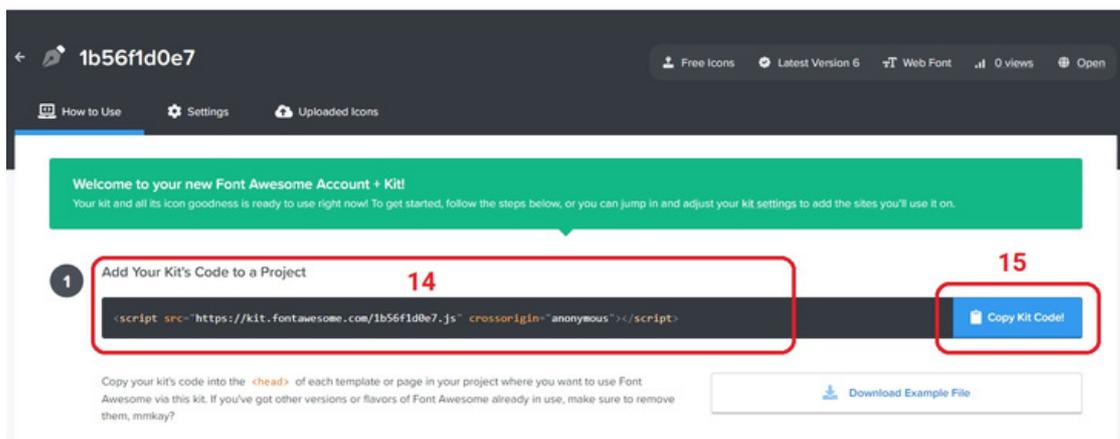
All set. Let's go! →

12 [No, thanks. Let's skip this step for now.](#)

Criando uma Conta no Font Awesome

14 - Nessa tela foi disponibilizado o código para usarmos em nosso projeto (esse código foi gerado para o e-mail o qual eu cadastrei para você será gerado um código diferente).

15 - Vamos clicar em Copy Kit Code (Copiar o Kit do Código)

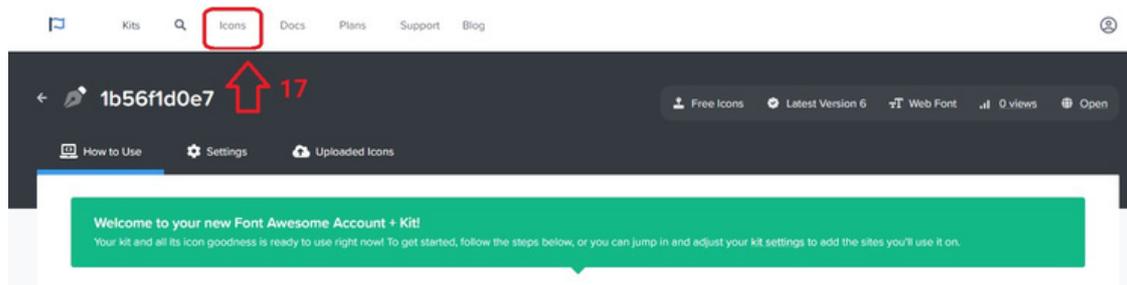


16 - Agora que o código foi copiado, vamos até o arquivo index.html do site Pet Shop do Bem Humorado (o qual estamos desenvolvendo) e dentro do <head> logo abaixo do <title> vamos copiar esse código. Ao fazermos a inserção desse código aqui ele vai habilitar o uso dos ícones do FONT AWESOME em nosso site.



Buscando os Ícones no Font Awesome - SECTION (Products)

17 - Vamos procurar dentro do site Font Awesome os ícones que queremos utilizar dentro do nosso site. Para isso clicaremos em Icons:



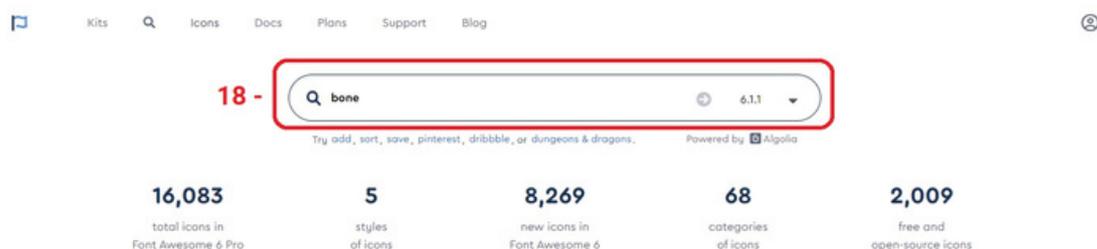
18 - Ao clicarmos em icons ele vai abrir uma nova página com milhares de ícones.

Precisamos de 3 ícones nesse momento para usarmos na Section (Products) do nosso site.

- Ossinho (que vai representar os Petiscos)
- Bolinha (que vai representar os Brinquedos)
- Comprimidos (que representarão os Remédios)

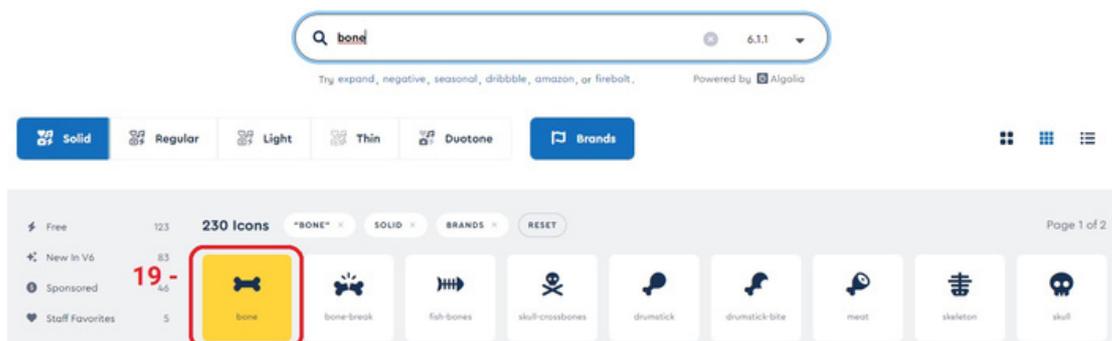
Para procurarmos o Ossinho dentro do Campo Search (Pesquisa) e vamos digitar:

bone (osso) e pressionaremos a tecla Enter

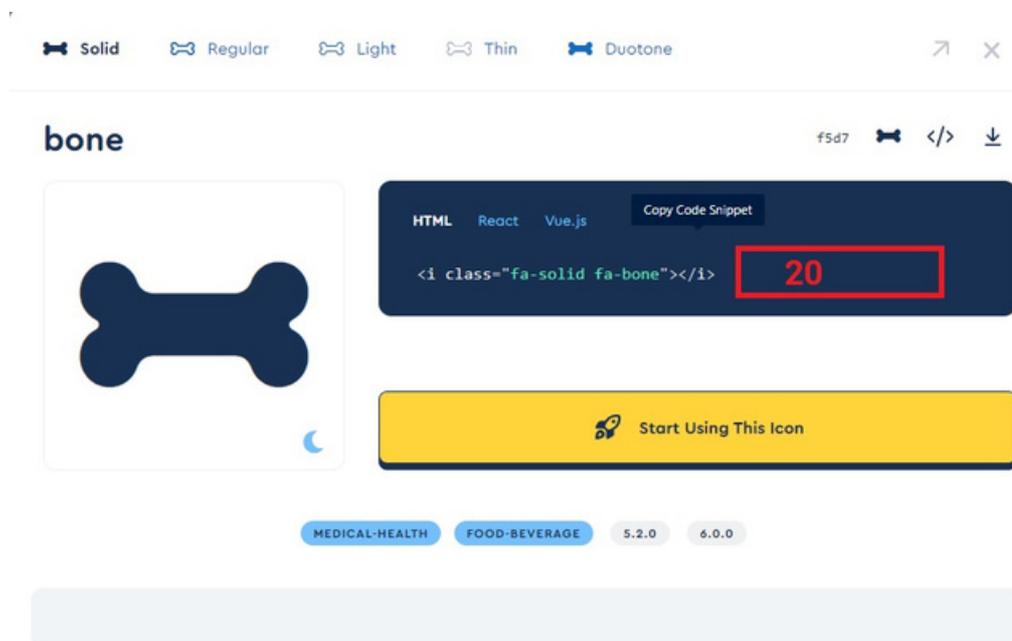


Buscando os Ícones no Font Awesome - SECTION (Products)

19 - Quando efetuamos a pesquisa alguns bones (ossos) vão aparecer aqui vamos usar a primeira opção (quando o mouse é colocado sobre o icone o qual queremos ele muda de cor)

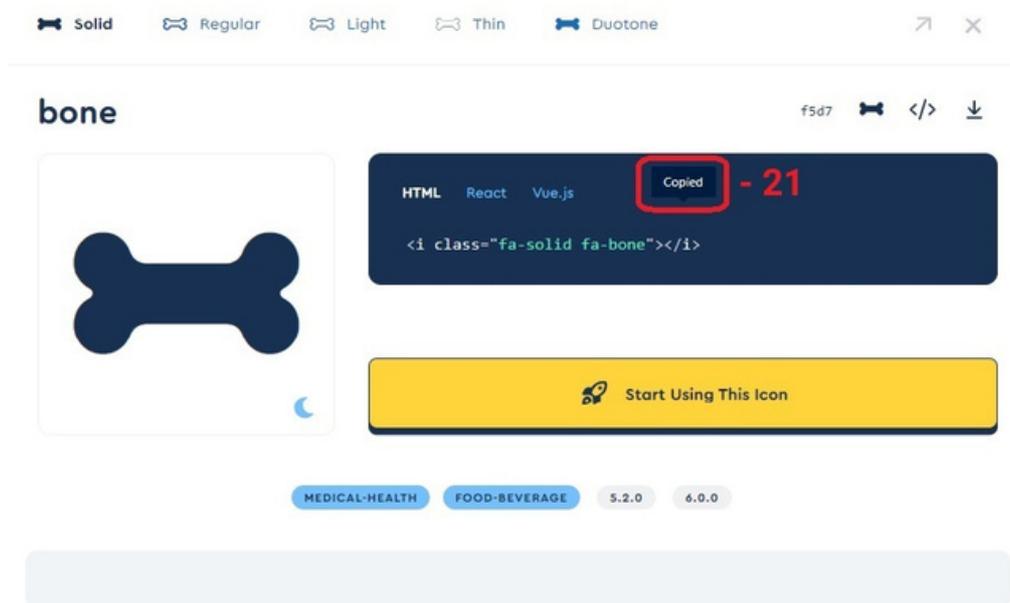


20 - Nessa tela vamos colocar o cursor do mouse próximo a linha a qual queremos copiar até aparecer uma mensagem Copy Code Snippet (Copiar o Código Selecionado)



Buscando os Ícones no Font Awesome - SECTION (Products)

21 - Depois que a mensagem mensagem Copy Code Snippet (Copiar o Código Selecionado) apareceu vamos clicar 1x na área do retângulo vermelho do Passo 20 (anterior) e ao clicarmos 1x a mensagem Copied (Copiado) é mostrada na tela.



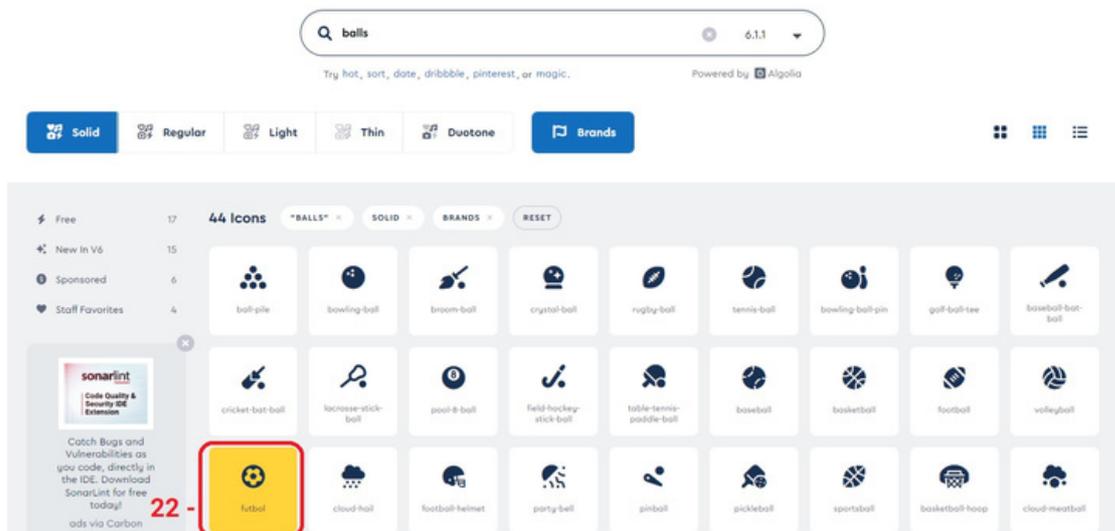
Isso significa que a linha `<i class="fa-solid fa-bone"></i>` foi copiada para uma área de transferência e podemos colocá-la dentro do (bloco de notas, do word e etc...) aqui o nosso objetivo é colocarmos essa linha dentro do arquivo index.html. Só que como precisamos pegar mais 2 ícones (a bolinha e as capsulas) dentro do Font Awesome e ainda não temos o código da SECTION (Products) definido... vamos deixar essa linha salva aqui abaixo para daqui a pouco colocá-la em nosso código.

```
<i class="fa-solid fa-bone"></i>
```

Buscando os Ícones no Font Awesome - SECTION (Products)

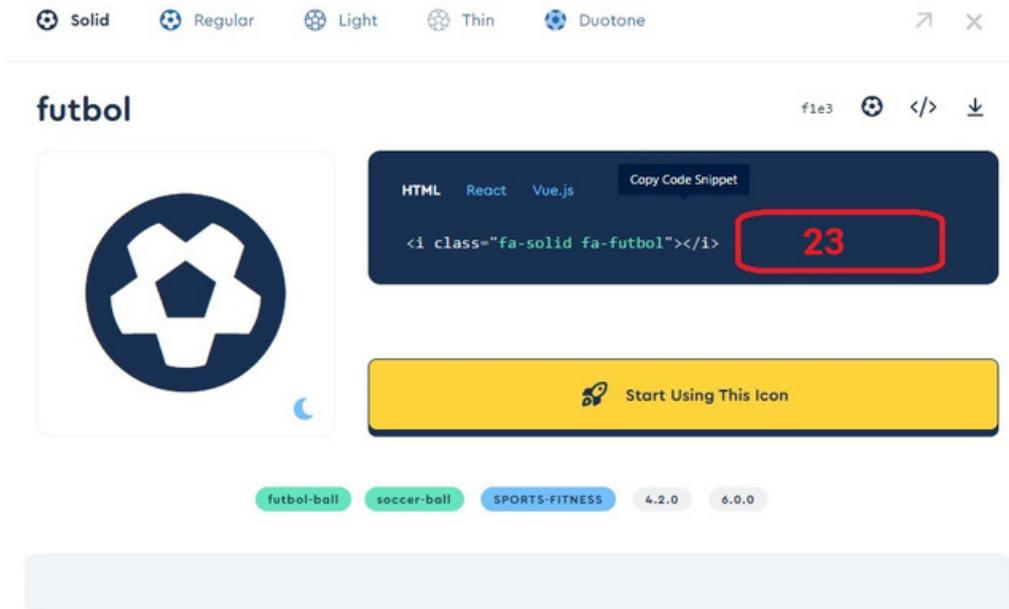
Agora vamos repetir o mesmo processo dos passos 19, 20 e 21 para pegar dentro do site do FONT AWESOME a linha de código que faz referência ao ícone da bolinha do nosso site.

22 - Quando efetuamos a pesquisa algumas balls (bolas) vão aparecer aqui vamos selecionar a primeira opção (quando o mouse é colocado sobre o ícone o qual queremos ele muda de cor)

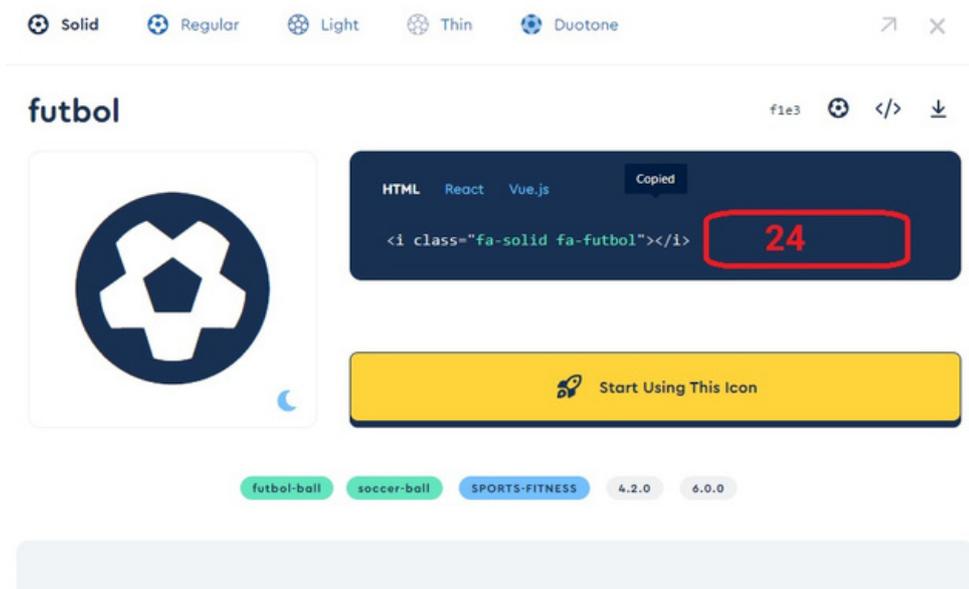


23 - Nessa tela vamos colocar o cursor do mouse próximo a linha a qual queremos copiar até aparecer uma mensagem Copy Code Snippet (Copiar o Código Seleccionado)

Buscando os Ícones no Font Awesome - SECTION (Products)



24 - Depois que a mensagem mensagem Copy Code Snippet (Copiar o Código Seleccionado) apareceu vamos clicar 1x na área do retangulo vermelho do Passo 23 (anterior) e ao clicarmos 1x a mensagem Copied (Copiado) é mostrada na tela.



Buscando os Ícones no Font Awesome - SECTION (Products)

Isso significa que a linha `<i class="fa-solid fa-futbol"></i>` foi copiada para uma área de transferência e podemos coloca-la dentro do (bloco de notas, do word e etc...) aqui o nosso objetivo é colocarmos essa linha também dentro do arquivo index.html. Só que como precisamos pegar mais 1 ícone (as capsulas) dentro do Font Awesome e ainda não temos o código da SECTION (Products) definido... vamos deixar essa linha salva aqui abaixo para daqui a pouco coloca-la em nosso código.

```
<i class="fa-solid fa-futbol"></i>
```

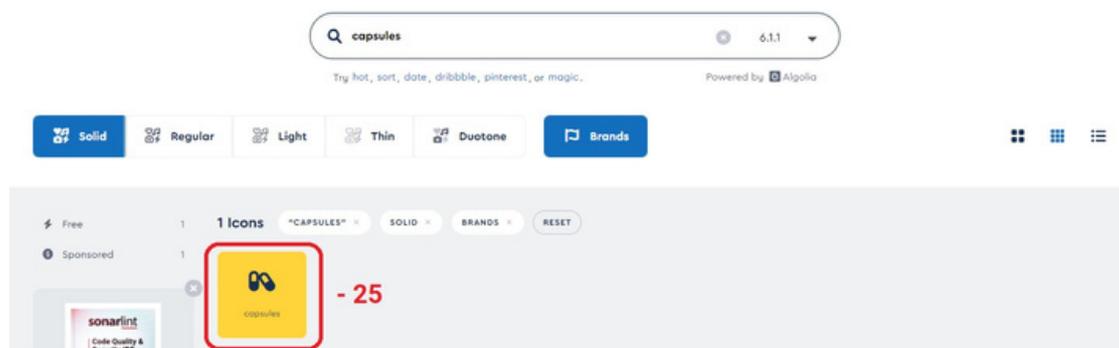


/igor-rebolla

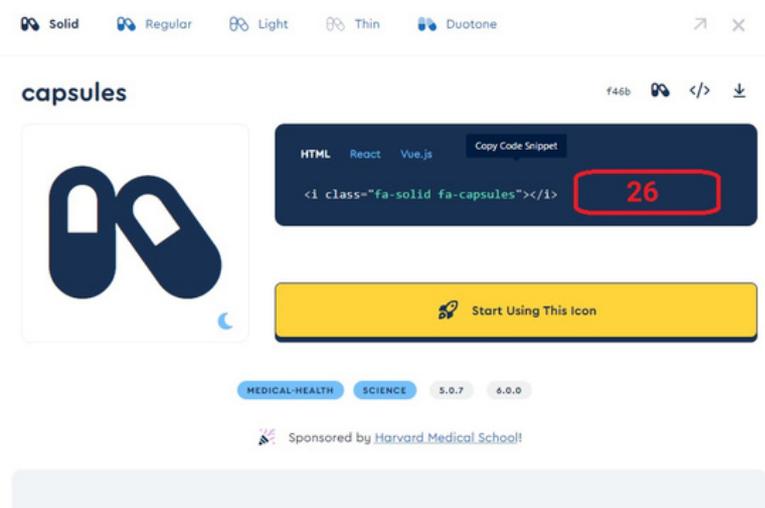
Buscando os Ícones no Font Awesome - SECTION (Products)

Agora vamos repetir o mesmo processo dos passos 22, 23 e 24 para pegar dentro do site do FONT AWESOME a linha de código que faz referência ao ícone da bolinha do nosso site.

25 - Quando efetuamos a pesquisa apenas uma busca por capsules (capsulas) irá aparecer aqui vamos selecionar a única opção (quando o mouse é colocado sobre o ícone o qual queremos ele muda de cor)

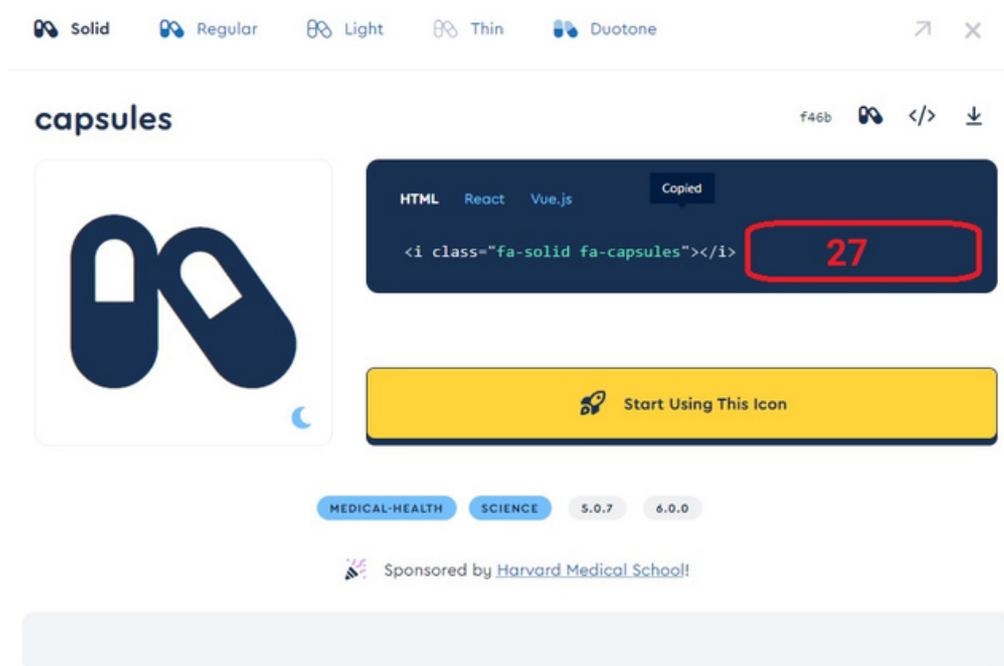


26 - Nessa tela vamos colocar o cursor do mouse próximo a linha a qual queremos copiar até aparecer uma mensagem Copy Code Snippet (Copiar o Código Seleccionado)



Buscando os Ícones no Font Awesome - SECTION (Products)

27 - Depois que a mensagem mensagem Copy Code Snippet (Copiar o Código Selecionado) apareceu vamos clicar 1x na área do retângulo vermelho do Passo 26 (anterior) e ao clicarmos 1x a mensagem Copied (Copiado) é mostrada na tela.



Buscando os Ícones no Font Awesome - SECTION (Products)

Isso significa que a linha `<i class="fa-solid fa-capsules"></i>` foi copiada para uma área de transferência e podemos colocá-la dentro do (bloco de notas, do word e etc...) aqui o nosso objetivo é colocarmos essa linha também dentro do arquivo index.html. Ainda não temos o código da SECTION (Products) definido... vamos deixar essa linha salva aqui abaixo junto com as outras 2 para daqui a pouco colocá-la em nosso código.

Ossinho - `<i class="fa-solid fa-bone"></i>`

Bola - `<i class="fa-solid fa-futbol"></i>`

Cápsulas - `<i class="fa-solid fa-capsules"></i>`

Agora que temos os três ícones definidos



[/igor-rebolla](#)

Código HTML do Site SECTION (Products)

O início do código da SECTION (Products) será feito logo abaixo do fechamento do comentário da Section (Clients).

```
<!-- INICIO DO CÓDIGO HTML - SECTION (Products) -->
<section id="products">
  <div class="headtext">Produtos</div>
  <div class="products-container">
    <div class="product">
      <i class="fa-solid fa-bone"></i>
      <span class="product-title">Petiscos</span>
      <p>Aqui você encontra Petiscos dos mais variados para o seu Pet. Até
em formato de ossinho.</p>
    </div>
    <div class="product">
      <i class="fa-solid fa-futbol"></i>
      <span class="product-title">Brinquedos</span>
      <p>Temos todos os tipos de Brinquedos para "cansar a beleza" do seu
amigo Pet.</p>
    </div>
    <div class="product">
      <i class="fa-solid fa-capsules"></i>
      <span class="product-title">Remédios</span>
      <p>Sabemos que os nossos amigos pets não gostam muito de tomar
remédios, mas caso eles fiquem "dodoi" temos uma farmácia completa
por aqui.</p>
    </div>
  </div>
</section>
<!-- FINAL DO CÓDIGO HTML - SECTION (Products) -->
```



/igor-rebolla

Explicação de Cada Elemento(Tag) do Código HTML do Site (SECTION - Products)

```
<!-- INICIO DO CÓDIGO HTML - SECTION (Products) -->
```

Comentário indicando onde é o início do Código HTML da SECTION Products (Só reforçando... esse comentário não é renderizado pelo navegador).

```
<section id="products">
```

Aqui criamos uma section com o id que recebe "products". A section terá um identificador(id) único e o "products" será utilizada para estilizar essa section quando formos trabalhar dentro do arquivo style.css

```
<div class="headtext">Produtos</div>
```

Aqui criamos uma div com a class "headtext". Essa div será a responsável por mostrar o "titulo Produtos" logo acima das 3 produtos (Petiscos, Brinquedos e Remédios) dentro da Section products.

```
<div class="products-container">
```

Dentro da div nós temos um container, esse container ficará responsável por agrupar os 3 produtos (Petiscos, Brinquedos e Remédios) dentro do products-container.

```
<div class="product">
```

Dentro de "products-container" temos uma div com a class "product". Essa div terá dentro dela o primeiro icone que pegamos dentro do Font Awesome (Ossinho - Bone) essa div ainda terá em dentro dela um span com a class "product-title" que recebe um subtítulo "Petiscos" e ainda conta com um parágrafo que faz a descrição do Produto (Petisco)



Explicação de Cada Elemento(Tag) do Código HTML do Site (SECTION - Products)

`</div>`

Essa div logo abaixo do fechamento da tag `</p>` indica o fechamento da tag `<div class="product">` desse primeiro produto (Petiscos)

Vamos repetir mais 1x a `<div class="product">` agora para os Brinquedos

`<div class="product">`

Dentro de "products-container" temos uma div com a class "product". Essa div terá dentro dela o segundo icone que pegamos dentro do Font Awesome (Bola - Balls) essa div ainda terá dentro dela um span com a class "product-title" que recebe um subtítulo "Brinquedos" e ainda conta com um parágrafo que faz a descrição do Produto (Brinquedos)

`</div>`

Essa div logo abaixo do fechamento da tag `</p>` indica o fechamento da tag `<div class="product">` desse segundo produto (Brinquedos)

Vamos repetir mais 1x a `<div class="product">` agora para os Remédios.

`<div class="product">`

Dentro de "products-container" temos uma div com a class "product". Essa div terá dentro dela o segundo icone que pegamos dentro do Font Awesome (Cápsulas - Capsules) essa div ainda terá dentro dela um span com a class "product-title" que recebe um subtítulo "Remédios" e ainda conta com um parágrafo que faz a descrição do Produto (Remédios)



[/igor-rebolla](#)

Explicação de Cada Elemento(Tag) do Código HTML do Site (SECTION - Products)

`</div>`

Essa div logo abaixo do fechamento da tag `</p>` indica o fechamento da tag `div` `<div class="product">` desse terceiro e último produto (Remédios).

`</div>`

Tag de fechamento dessa div aqui `<div class="products-container">`

`</section>`

Tag de fechamento dessa section aqui `<section id="products">`

`<!--FINAL DO CÓDIGO HTML - SECTION (Products) -->`

Comentário indicando onde é o final do Código HTML da SECTION Products (Só reforçando... esse comentário não é renderizado pelo navegador).

Agora vamos ver na imagem da próxima página como toda essa estrutura da SECTION (Products) ficará dentro do VsCode.



/igor-rebolla

Explicação de Cada Elemento(Tag) do Código HTML do Site (SECTION - Products)

```
<!-- FINAL DO CÓDIGO HTML - SECTION (Clients) -->
<!-- INICIO DO CÓDIGO HTML - SECTION (Products) -->
<section id="products">
  <div class="headtext">Produtos</div>
  <div class="products-container">
    <div class="product">
      <i class="fa-solid fa-bone"></i>
      <span class="product-title">Petiscos</span>
      <p>Aqui você encontra Petiscos dos mais variados para o seu Pet. Até em formato de ossinho.</p>
    </div>
    <div class="product">
      <i class="fa-solid fa-futbol"></i>
      <span class="product-title">Brinquedos</span>
      <p>Temos todos os tipos de Brinquedos para "cansar a beleza" do seu amigo Pet.</p>
    </div>
    <div class="product">
      <i class="fa-solid fa-capsules"></i>
      <span class="product-title">Remédios</span>
      <p>Sabemos que os nossos amigos pets não gostam muito de tomar remédios, mas caso eles fiquem "dodoi" temos uma farmácia completa por aqui.</p>
    </div>
  </div>
</section>
<!-- FINAL DO CÓDIGO HTML - SECTION (Products) -->
```

Na imagem abaixo podemos ver o resultado da estrutura da SECTION (Products) sendo renderizado(mostrado) no navegador (Google Chrome)



Explicação de Cada Elemento(Tag) do Código HTML do Site (SECTION - Products)

Até aqui estruturamos a NAVBAR, o HEADER, a SECTION CLIENTS e a SECTION PRODUCTS do nosso site, vamos partir para a quinta estrutura do nosso site que será:

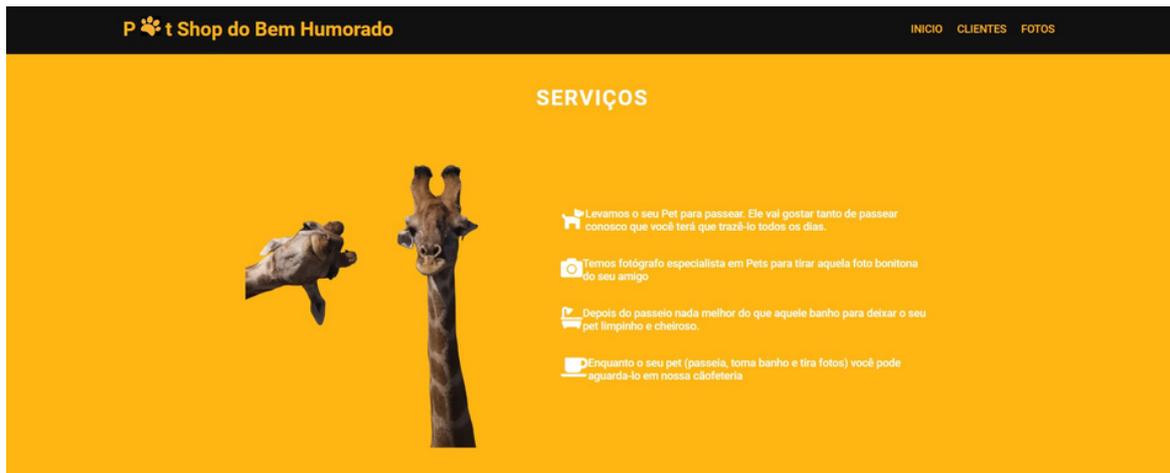
5° - SECTION (SERVIÇOS)

E aprendemos também como criar uma conta no site FONT AWESOME, criar um KIT para estar apto a procurar os ícones e por fim localizamos 3 ícones para 3 produtos do nosso site. Pois bem agora na SECTION (Serviços) vamos buscar mais 4 ícones para o nosso site.

Então bora ver esse tal de SECTION (Serviços) já na próxima página....

Código HTML do Site SECTION (Services)

Aqui é onde vamos iniciar toda a estrutura de códigos HTML que compõem a SECTION (Services). O objetivo final depois de estilizarmos (style.css) a Section (Services) é esse abaixo:



Notem que na imagem acima existem 4 ícones (Cachorrinho, Câmera, Banheira e o Café) esses ícones eu também busquei do FONT AWESOME.

Agora que já temos o nosso cadastro criado no Font Awesome, temos o nosso Kit que permite a utilização dos ícones do site e já temos esse Kit inserido no <head> dentro do arquivo index.html... só precisamos acessar a parte de pesquisa do site e buscar pelos ícones que vamos usar no site

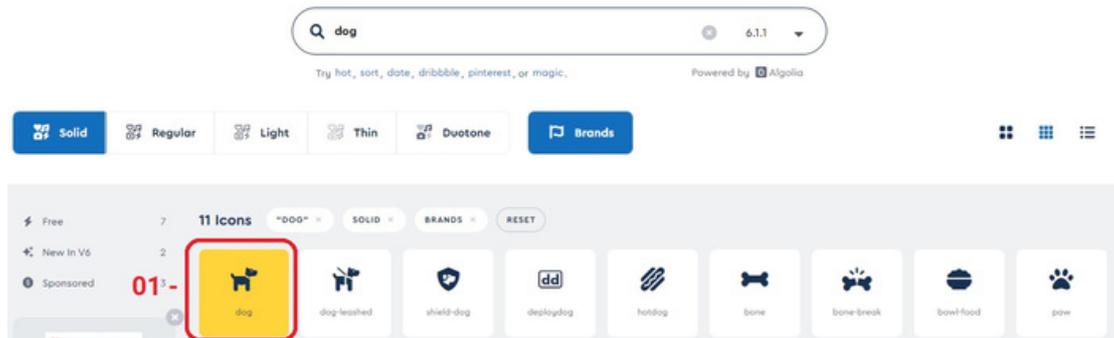
Então bora pegar esses 4 ícones dentro do Font Awesome....



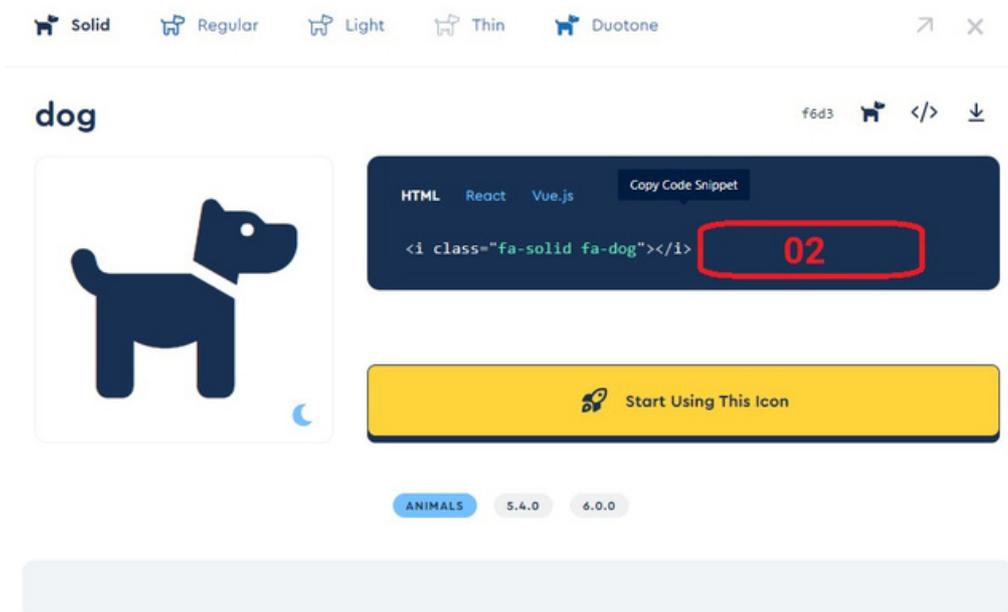
/igor-rebolla

Buscando os Ícones no Font Awesome - SECTION (Services)

01 - Quando efetuamos a pesquisa dois dogs (cachorros) vão aparecer aqui vamos selecionar a primeira opção (quando o mouse é colocado sobre o ícone o qual queremos ele muda de cor)

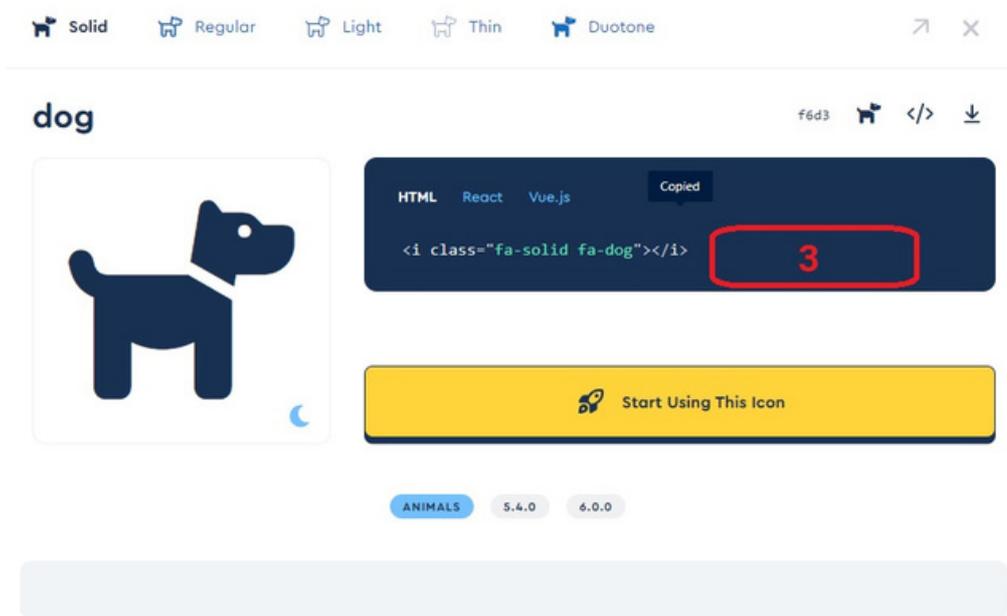


02 - Nessa tela vamos colocar o cursor do mouse próximo a linha a qual queremos copiar até aparecer uma mensagem Copy Code Snippet (Copiar o Código Selecionado)



Buscando os Ícones no Font Awesome - SECTION (Services)

03 - Depois que a mensagem mensagem Copy Code Snippet (Copiar o Código Selecionado) apareceu vamos clicar 1x na área do retângulo vermelho do Passo 02 (anterior) e ao clicarmos 1x a mensagem Copied (Copiado) é mostrada na tela.



Buscando os Ícones no Font Awesome - SECTION (Services)

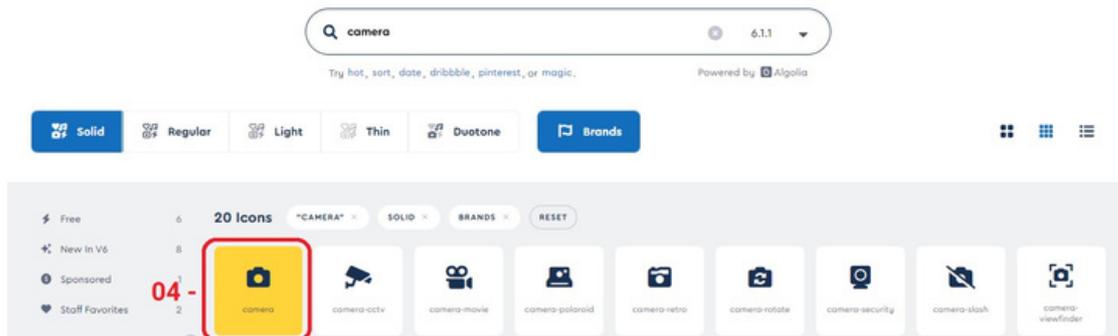
Isso significa que a linha `<i class="fa-solid fa-dog"></i>` foi copiada para uma área de transferência e podemos colocá-la dentro do (bloco de notas, do word e etc...) aqui o nosso objetivo é colocarmos essa linha também dentro do arquivo index.html. Só que como precisamos pegar mais 3 ícones (Câmera, Banheira e o Café) dentro do Font Awesome e ainda não temos o código da SECTION (Services) definido... vamos deixar essa linha salva aqui abaixo para daqui a pouco colocá-la em nosso código.

```
<i class="fa-solid fa-dog"></i>
```

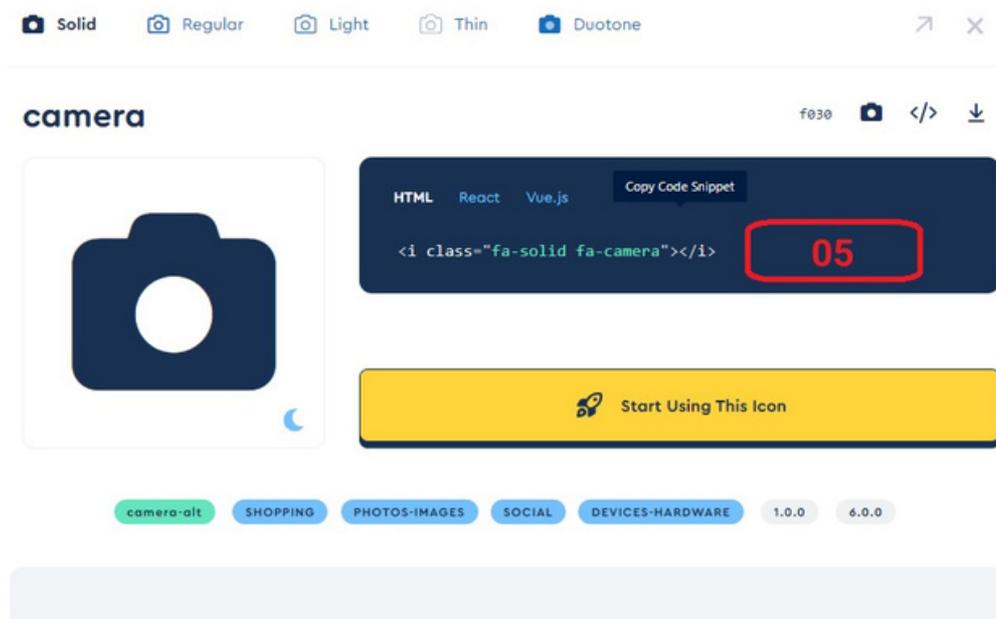


Buscando os Ícones no Font Awesome - SECTION (Services)

04 - Quando efetuamos a pesquisa algumas Cameras (Cameras) vão aparecer aqui vamos selecionar a primeira opção (quando o mouse é colocado sobre o ícone o qual queremos ele muda de cor)

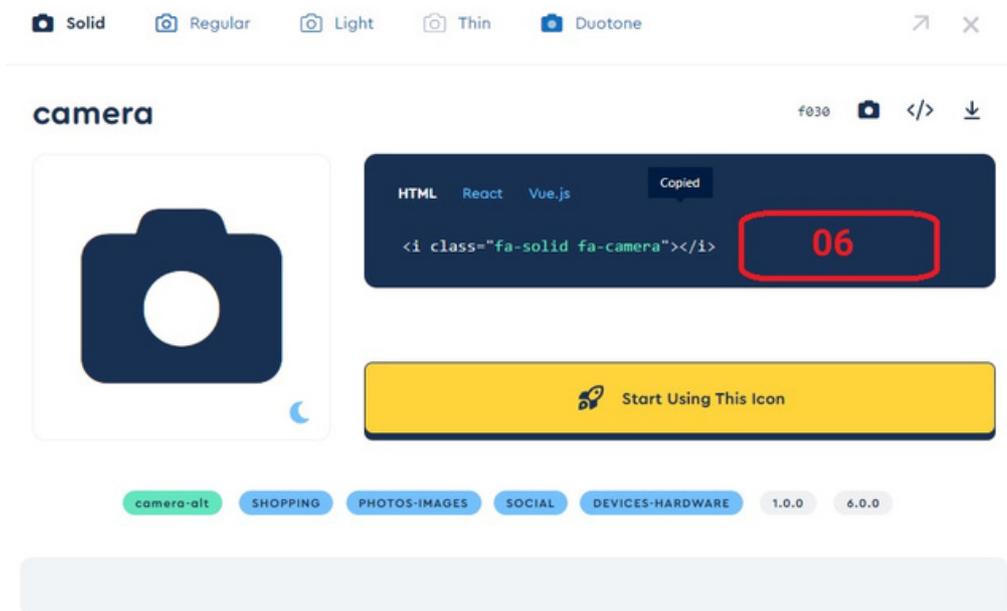


05 - Nessa tela vamos colocar o cursor do mouse próximo a linha a qual queremos copiar até aparecer uma mensagem Copy Code Snippet (Copiar o Código Seleccionado)



Buscando os Ícones no Font Awesome - SECTION (Services)

06 - Depois que a mensagem mensagem Copy Code Snippet (Copiar o Código Selecionado) apareceu vamos clicar 1x na área do retângulo vermelho do Passo 05 (anterior) e ao clicarmos 1x a mensagem Copied (Copiado) é mostrada na tela.



Buscando os Ícones no Font Awesome - SECTION (Services)

Isso significa que a linha `<i class="fa-solid fa-camera"></i>` foi copiada para uma área de transferência e podemos colocá-la dentro do (bloco de notas, do word e etc...) aqui o nosso objetivo é colocarmos essa linha também dentro do arquivo index.html. Só que como precisamos pegar mais 2 ícones (Banheira e o Café) dentro do Font Awesome e ainda não temos o código da SECTION (Services) definido... vamos deixar essa linha salva aqui abaixo para daqui a pouco colocá-la em nosso código.

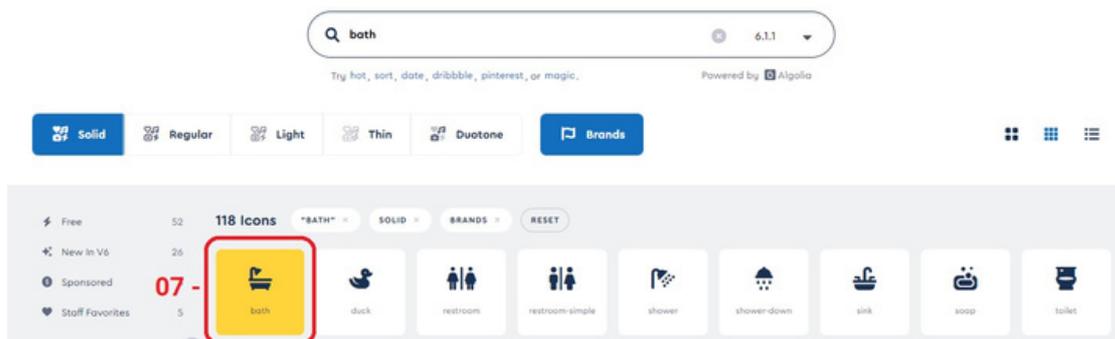
```
<i class="fa-solid fa-camera"></i>
```



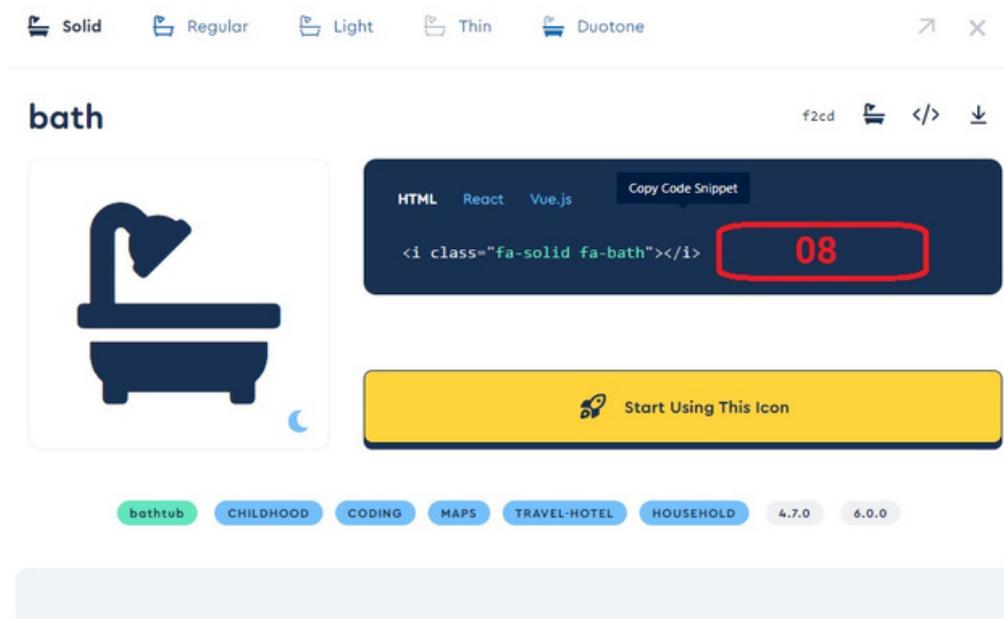
/igor-rebolla

Buscando os Ícones no Font Awesome - SECTION (Services)

07 - Quando efetuamos a pesquisa um único bath (Banho) irá aparecer aqui vamos selecionar essa opção (quando o mouse é colocado sobre o ícone o qual queremos ele muda de cor)

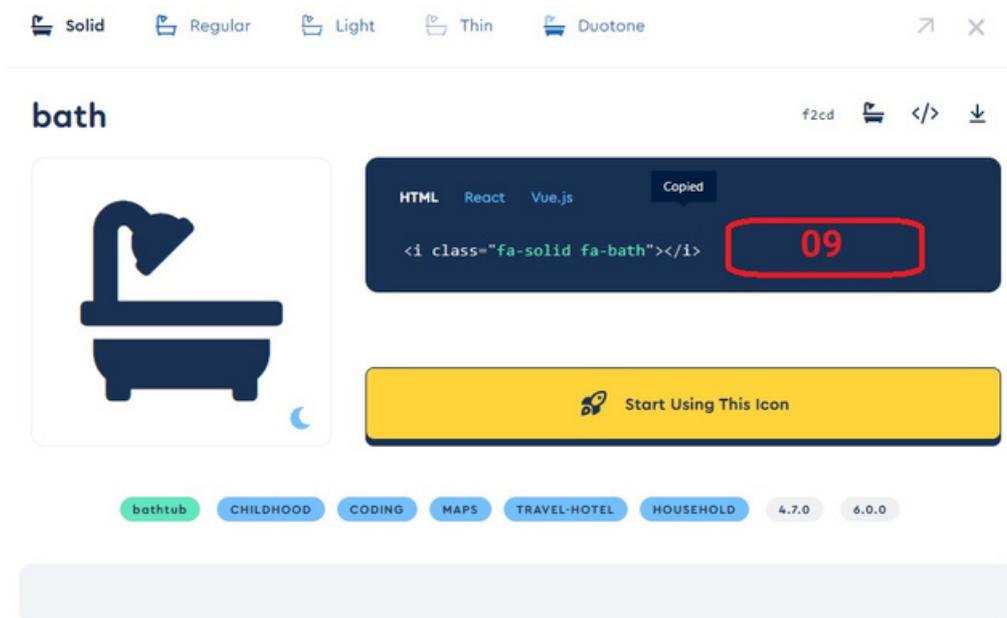


08 - Nessa tela vamos colocar o cursor do mouse próximo a linha a qual queremos copiar até aparecer uma mensagem Copy Code Snippet (Copiar o Código Selecionado)



Buscando os Ícones no Font Awesome - SECTION (Services)

09 - Depois que a mensagem mensagem Copy Code Snippet (Copiar o Código Selecionado) apareceu vamos clicar 1x na área do retângulo vermelho do Passo 08 (anterior) e ao clicarmos 1x a mensagem Copied (Copiado) é mostrada na tela.



Buscando os Ícones no Font Awesome - SECTION (Services)

Isso significa que a linha `<i class="fa-solid fa-bath"></i>` foi copiada para uma área de transferência e podemos colocá-la dentro do (bloco de notas, do word e etc...) aqui o nosso objetivo é colocarmos essa linha também dentro do arquivo index.html. Só que como precisamos pegar mais 1 ícone (Café) dentro do Font Awesome e ainda não temos o código da SECTION (Services) definido... vamos deixar essa linha salva aqui abaixo para daqui a pouco colocá-la em nosso código.

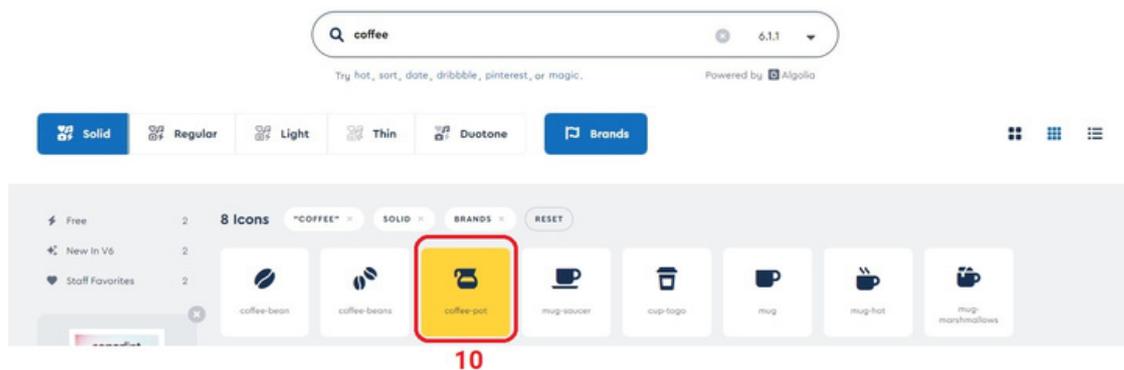
```
<i class="fa-solid fa-bath"></i>
```



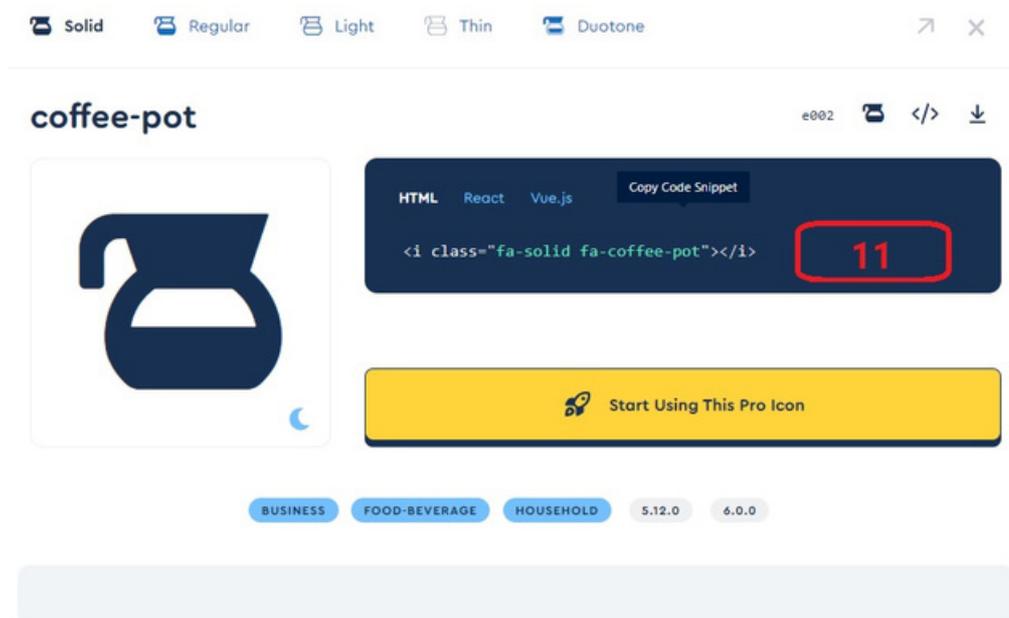
/igor-rebolla

Buscando os Ícones no Font Awesome - SECTION (Services)

10 - Quando efetuamos a pesquisa algumas opções sobre Coffee (Café) vão aparecer aqui vamos selecionar a terceira opção (quando o mouse é colocado sobre o ícone o qual queremos ele muda de cor)

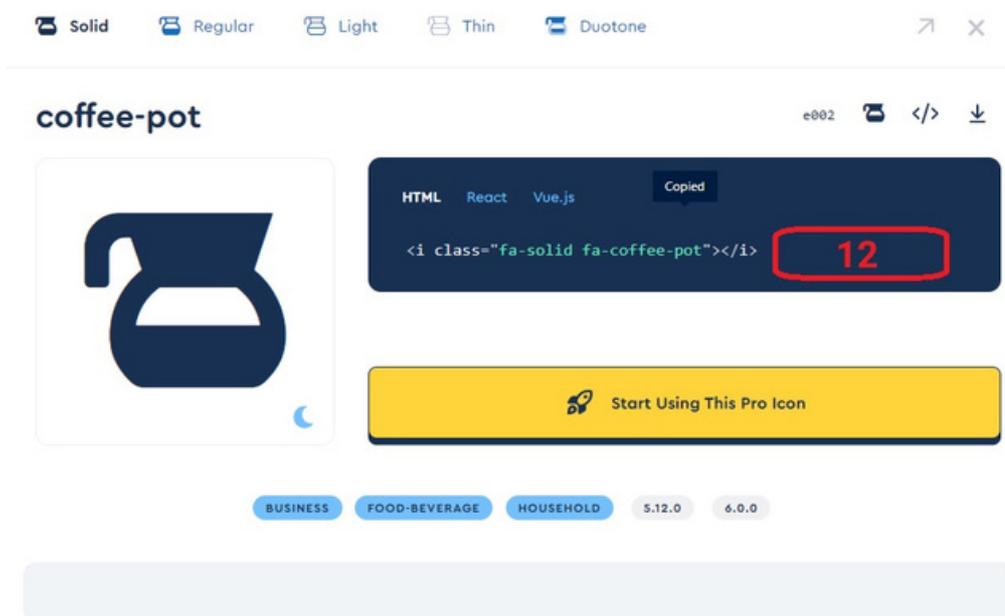


11 - Nessa tela vamos colocar o cursor do mouse próximo a linha a qual queremos copiar até aparecer uma mensagem Copy Code Snippet (Copiar o Código Seleccionado)



Buscando os Ícones no Font Awesome - SECTION (Services)

12 - Depois que a mensagem mensagem Copy Code Snippet (Copiar o Código Selecionado) apareceu vamos clicar 1x na área do retângulo vermelho do Passo 11 (anterior) e ao clicarmos 1x a mensagem Copied (Copiado) é mostrada na tela.



Buscando os Ícones no Font Awesome - SECTION (Services)

Isso significa que a linha `<i class="fa-solid fa-coffee-pot"></i>` foi copiada para uma área de transferência e podemos coloca-la dentro do (bloco de notas, do word e etc...) aqui o nosso objetivo é colocarmos essa linha também dentro do arquivo index.html. Ainda não temos o código da SECTION (Services) definido... vamos deixar essa linha salva aqui abaixo junto com as outras 3 para daqui a pouco coloca-la em nosso código.

Cachorro - `<i class="fa-solid fa-dog"></i>`

Camera - `<i class="fa-solid fa-camera"></i>`

Banheira - `<i class="fa-solid fa-bath"></i>`

Café - `<i class="fa-solid fa-coffee-pot"></i>`



/igor-rebolla

Código HTML do Site SECTION (Services)

O início do código da SECTION (Services) será feito logo abaixo do fechamento do comentário da SECTION (Products).

```
<!-- FINAL DO CÓDIGO HTML - SECTION (Products) -->

<!-- INICIO DO CÓDIGO HTML - SECTION (Services) -->
<section id="services">
  <div class="headtext">Serviços</div>
  <div class="services-container">
    
    <div class="items">
      <div class="item">
        <i class="fas fa-dog fa-2x"></i>
        <p>Levamos o seu Pet para passear. Ele vai gostar tanto de passear conosco que você terá que trazê-lo todos os dias.</p>
      </div>
      <div class="item">
        <i class="fas fa-camera fa-2x"></i>
        <p>Temos fotógrafo especialista em Pets para tirar aquela foto bonita do seu amigo</p>
      </div>
      <div class="item">
        <i class="fas fa-bath fa-2x"></i>
        <p>Depois do passeio nada melhor do que aquele banho para deixar o seu pet limpinho e cheiroso.</p>
      </div>
    </div>
  </div>
</section>
```

O código da SECTION (Services) continua na próxima página...



Código HTML do Site SECTION (Services)

```
<div class="item">  
  <i class="fas fa-coffee fa-2x"></i>  
  <p>Enquanto o seu pet (passeia, toma banho e tira fotos) você pode  
aguarda-lo em nossa cãofeteria</p>  
</div>  
</div>  
</div>  
</section>  
  
<!-- FINAL DO CÓDIGO HTML - SECTION (Services) -->
```



[/igor-rebolla](#)

Explicação de Cada Elemento(Tag) do Código HTML do Site (SECTION - Services)

```
<!-- INICIO DO CÓDIGO HTML - SECTION (Services) -->
```

Comentário indicando onde é o início do Código HTML da SECTION Services (Só reforçando... esse comentário não é renderizado pelo navegador).

```
<section id="services">
```

Aqui criamos uma section com o id que recebe "services". A section terá um identificador(id) único e o "services" será utilizada para estilizar essa section quando formos trabalhar dentro do arquivo style.css

```
<div class="headtext">Serviços</div>
```

Aqui criamos uma div com a class "headtext". Essa div será a responsável por mostrar o "titulo Serviços" logo acima da foto das 2 girafas e dos 4 serviços (Passeio, Foto, Banho e Café) dentro da Section services.

```
<div class="services-container">
```

Dentro da div nós temos um container, esse container ficará responsável por agrupar a foto das 2 girafas e dos 4 serviços (Passeio, Foto, Banho e Café) dentro de services-container.

```
<div class="services">
```

Dentro de "services-container" temos uma div com a class "items". Essa div terá dentro dela outra div com a class "item" que vai receber cada item dos nossos serviços. O primeiro ícone que pegamos dentro do Font Awesome (Cachorro - Dog) essa div ainda terá em dentro dela um parágrafo que faz a descrição do Serviço (Passeio)



Explicação de Cada Elemento(Tag) do Código HTML do Site (SECTION - Services)

`</div>`

Essa div logo abaixo do fechamento da tag `</p>` indica o fechamento da tag `div <div class="item">` desse primeiro Serviço (Passeio)

Vamos repetir mais 1x a `<div class="item">` agora para as Fotos

`<div class="item">`

Dentro de "services-container" temos uma div com a class "items". Essa div terá dentro dela outra div com a class "item" que vai receber cada item dos nossos serviços. O primeiro ícone que pegamos dentro do Font Awesome (Camera - Camera) essa div ainda terá dentro dela um parágrafo que faz a descrição do Serviço (Passeio)

`</div>`

Essa div logo abaixo do fechamento da tag `</p>` indica o fechamento da tag `div <div class="item">` desse segundo Serviço (Fotos)

Vamos repetir mais 1x a `<div class="">` agora para o Banho

`<div class="item">`

Dentro de "services-container" temos uma div com a class "items". Essa div terá dentro dela outra div com a class "item" que vai receber cada item dos nossos serviços. O primeiro ícone que pegamos dentro do Font Awesome (Banho - Bath) essa div ainda terá dentro dela um parágrafo que faz a descrição do Serviço (Banho)



/igor-rebolla

Explicação de Cada Elemento(Tag) do Código HTML do Site (SECTION - Services)

`</div>`

Essa div logo abaixo do fechamento da tag `</p>` indica o fechamento da tag `div <div class="item">` desse Terceiro Serviço (Banho)

Vamos repetir mais 1x a `<div class="">` agora para a Cafeteria

`<div class="item">`

Dentro de "services-container" temos uma div com a class "items". Essa div terá dentro dela outra div com a class "item" que vai receber cada item dos nossos serviços. O primeiro ícone que pegamos dentro do Font Awesome (Café - Coffee) essa div ainda terá dentro dela um parágrafo que faz a descrição do Serviço (Cafeteria)

`</div>`

Essa div logo abaixo do fechamento da tag `</p>` indica o fechamento da tag `div <div class="item">` desse quarto e último produto (Cafeteria).

`</div>`

Tag de fechamento dessa div aqui `<div class="items">`

`</div>`

Tag de fechamento dessa div aqui `<div class="services-container">`

`</section>`

Tag de fechamento dessa section aqui `<section id="services">`



/igor-rebolla

Explicação de Cada Elemento(Tag) do Código HTML do Site (SECTION - Services)

<!--FINAL DO CÓDIGO HTML - SECTION (Services) -->

Comentário indicando onde é o final do Código HTML da SECTION Services (Só reforçando... esse comentário não é renderizado pelo navegador).

Agora vamos ver na imagem abaixo como toda essa estrutura da SECTION (Services) ficará dentro do VsCode.

```
<!-- FINAL DO CÓDIGO HTML - SECTION (Products) -->

<!-- INICIO DO CÓDIGO HTML - SECTION (Services) -->
<section id="services">
  <div class="headtext">Serviços</div>
  <div class="services-container">
    
    <div class="items">
      <div class="item">
        <i class="fas fa-dog fa-2x"></i>
        <p>Levamos o seu Pet para passear. Ele vai gostar tanto de passear conosco que você terá que trazê-lo todos os dias.</p>
      </div>
      <div class="item">
        <i class="fas fa-camera fa-2x"></i>
        <p>Temos fotógrafo especialista em Pets para tirar aquela foto bonita do seu amigo</p>
      </div>
      <div class="item">
        <i class="fas fa-bath fa-2x"></i>
        <p>Depois do passeio nada melhor do que aquele banho para deixar o seu pet limpinho e cheiroso.</p>
      </div>
      <div class="item">
        <i class="fas fa-coffee fa-2x"></i>
        <p>Enquanto o seu pet (passeia, toma banho e tira fotos) você pode aguarda-lo em nossa cãofeteria</p>
      </div>
    </div>
  </div>
</section>

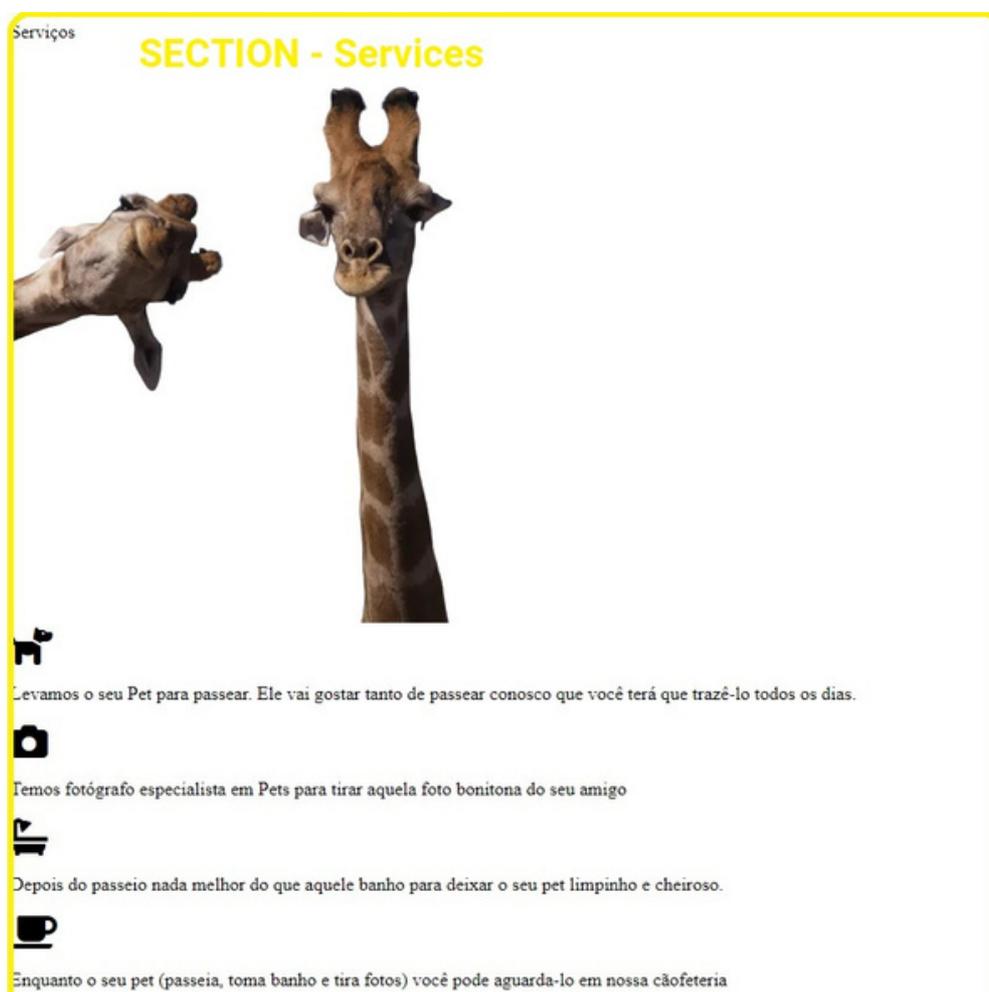
<!-- FINAL DO CÓDIGO HTML - SECTION (Services) -->
```



/igor-rebolla

Explicação de Cada Elemento(Tag) do Código HTML do Site (SECTION - Services)

Na imagem abaixo podemos ver o resultado do estrutura da SECTION (Services) sendo renderizada(mostrada) no navegador (Google Chrome)



Explicação de Cada Elemento(Tag) do Código HTML do Site (SECTION - Services)

Até aqui estruturamos a NAVBAR, o HEADER, a SECTION CLIENTS, a SECTION PRODUCTS e a SECTION SERVICES, do nosso site, vamos partir para a sexta estrutura do nosso site que será:

6° - SECTION (Depoimentos)

Então bora ver esse tal de SECTION (Depoimentos) já na próxima página....

Código HTML do Site SECTION (Depoimentos)

O início do código da SECTION (Depoimentos) será feito logo abaixo do fechamento do comentário da SECTION (Services).

```
<!-- FINAL DO CÓDIGO HTML - SERVICES -->

<!-- INICIO DO CÓDIGO HTML - DEPOIMENTOS -->
<section id="depoimentos">
  <div class="headtext">Depoimentos</div>
  <div class="depoimentos-container">
    <div class="depoimento">
      
      <p>"Não gostei!! Me falaram que era passeio só que era Vacina" -
Traduzido pelo Dr. Dolittle® </p>
    </div>
    <div class="depoimento">
      
      <p>"Vim só pelo petisco - Quero dizer Meowwwwwwwwwwwwwwwwwww" -
Traduzido pelo Dr. Dolittle® </p>
    </div>
  </div>
</section>

<!-- FINAL DO CÓDIGO HTML - DEPOIMENTOS -->
```

Explicação de Cada Elemento(Tag) do Código HTML Site (SECTION - Depoimentos)

```
<!-- INICIO DO CÓDIGO HTML - SECTION (Depoimentos) -->
```

Comentário indicando onde é o início do Código HTML da SECTION Depoimentos (Só reforçando... esse comentário não é renderizado pelo navegador).

```
<section id="depoimentos">
```

Aqui criamos uma section com o id que recebe "depoimentos". A section terá um identificador(id) único e o "depoimentos" será utilizado para estilizar essa section quando formos trabalhar dentro do arquivo style.css

```
<div class="headtext">Depoimentos</div>
```

Aqui criamos uma div com a class "headtext". Essa div será a responsável por mostrar o "titulo Depoimentos" logo acima da foto do cachorro e do seu respectivo paragrafo dentro da Section depoimentos.

```
<div class="depoimentos-container">
```

Dentro da div nós temos um container, esse container ficará responsável por agrupar as fotos (cachorro e do gato) junto com seus respectivos paragrafos dentro de depoimentos-container.

```
<div class="depoimento">
```

Dentro de "depoimentos-container" temos uma img(imagem) que recebe a classe "depoimento-image". Essa imagem faz referência ao depoimento do (Cachorro) e ainda terá dentro dela um parágrafo que faz a descrição do Depoimento do (Cachorro)



/igor-rebolla

Explicação de Cada Elemento(Tag) do Código HTML Site (SECTION - Depoimentos)

Vamos repetir mais 1x a `<div class="">` agora para a Depoimento do Gato

```
<div class="depoimento">
```

Dentro de "services-depoimento" temos uma `img`(imagem) que recebe a classe "depoimento-image". Essa imagem faz referência ao depoimento do (Gato) e ainda terá dentro dela um parágrafo que faz a descrição do Depoimento do (Gato)

```
</div>
```

Essa `div` logo abaixo do fechamento da tag `</p>` indica o fechamento da tag `div` `<div class="depoimento">` desse segundo e último depoimento (Gato).

```
</div>
```

Tag de fechamento dessa `div` aqui `<div class="depoimentos-container">`

```
</section>
```

Tag de fechamento dessa `section` aqui `<section id="depoimentos">`

```
<!--FINAL DO CÓDIGO HTML - SECTION (Depoimentos) -->
```

Comentário indicando onde é o final do Código HTML da SECTION Depoimentos (Só reforçando... esse comentário não é renderizado pelo navegador).

Agora vamos ver na imagem da próxima página como toda essa estrutura da SECTION (Depoimentos) ficará dentro do VsCode.



Explicação de Cada Elemento(Tag) do Código HTML Site (SECTION - Depoimentos)

Até aqui estruturamos a NAVBAR, o HEADER, a SECTION CLIENTS, a SECTION PRODUCTS, a SECTION SERVICES e a SECTION DEPOIMENTOS do nosso site, vamos partir para a sétima e penúltima estrutura do nosso site que será:

7° - SECTION (Photos)

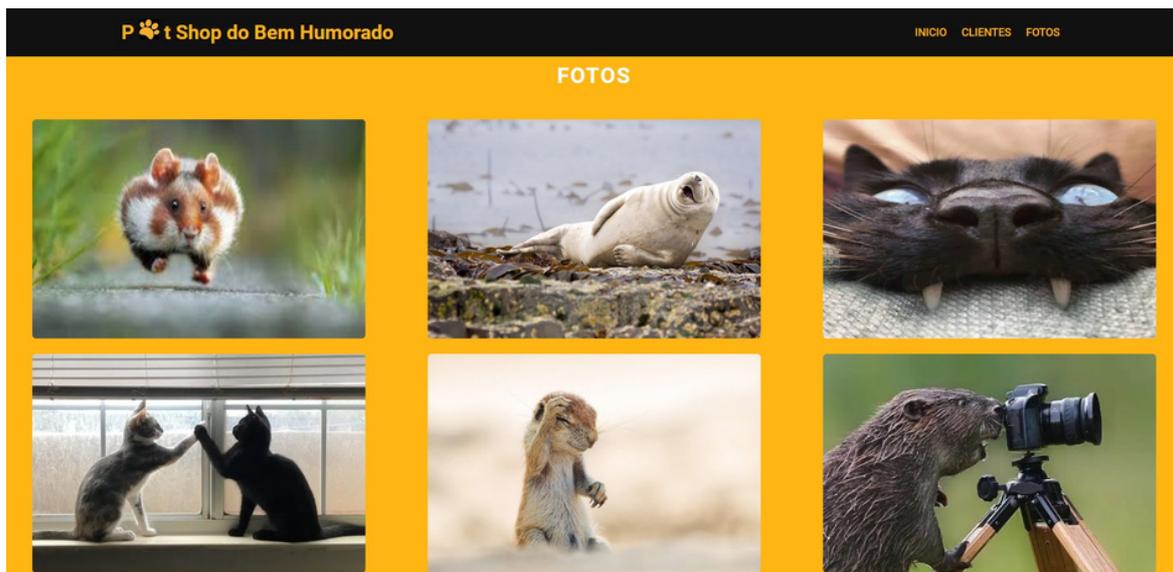
Então bora ver esse tal de SECTION (Photos) já na próxima página....



/igor-rebolla

Código HTML do Site SECTION (Photos)

Aqui é onde vamos iniciar toda a estrutura de códigos HTML que compõem a SECTION (Photos). O objetivo final depois de estilizarmos (style.css) a Section (Photos) é esse abaixo:



Código HTML do Site

SECTION (Photos)

O início do código da SECTION (Photos) será feito logo abaixo do fechamento do comentário da SECTION (Depoimentos).

```
<!-- FINAL DO CÓDIGO HTML - DEPOIMENTOS -->

<!-- INICIO DO CÓDIGO HTML - PHOTOS -->

<section id="fotos">
  <div class="headtext">Fotos</div>
  <div class="photos-container">
    
    
    
    
    
    
  </div>
</section>

<!-- FINAL DO CÓDIGO HTML - PHOTOS -->
```



/igor-rebolla

Explicação de Cada Elemento(Tag) do Código HTML do Site (SECTION - Photos)

```
<!-- INICIO DO CÓDIGO HTML - SECTION (Fotos) -->
```

Comentário indicando onde é o início do Código HTML da SECTION Depoimentos (Só reforçando... esse comentário não é renderizado pelo navegador).

```
<section id="photos">
```

Aqui criamos uma section com o id que recebe "photos". A section terá um identificador(id) único e o "photos" será utilizado para estilizar essa section quando formos trabalhar dentro do arquivo style.css

```
<div class="headtext">Fotos</div>
```

Aqui criamos uma div com a class "headtext". Essa div será a responsável por mostrar o "titulo Fotos" logo acima das 3 fotos que estão na primeira linha.

```
<div class="photos-container">
```

Dentro da div nós temos um container, esse container ficará responsável por agrupar as 6 fotos dentro de photos-container.

Dentro de "photos-container" temos nossa primeira img(imagem-1.jpg) que recebe a classe "photos-image".

Dentro de "photos-container" temos nossa segunda img(imagem-2.jpg) que recebe a classe "photos-image".



/igor-rebolla

Explicação de Cada Elemento(Tag) do Código HTML Site (SECTION - Depoimentos)

Dentro de "photos-container" temos nossa terceira img(imagem-3.jpg) que recebe a classe "photos-image".

Dentro de "photos-container" temos nossa quarta img(imagem-4.jpg) que recebe a classe "photos-image".

Dentro de "photos-container" temos nossa quinta img(imagem-5.jpg) que recebe a classe "photos-image".

Dentro de "photos-container" temos nossa sexta img(imagem-6.jpg) que recebe a classe "photos-image".

`</div>`

Tag de fechamento dessa div aqui `<div class="photos-container">`

`</section>`

Tag de fechamento dessa section aqui `<section id="photos">`

`<!-- FINAL DO CÓDIGO HTML - SECTION (Photos) -->`

Comentário indicando onde é o final do Código HTML da SECTION Photos (Só reforçando... esse comentário não é renderizado pelo navegador).



Explicação de Cada Elemento(Tag) do Código HTML Site (SECTION - Depoimentos)

Agora vamos ver na imagem abaixo como toda essa estrutura da SECTION (Photos) ficará dentro do VsCode.

```
<!-- FINAL DO CÓDIGO HTML - DEPOIMENTOS -->

<!-- INICIO DO CÓDIGO HTML - PHOTOS -->
<section id="photos">
  <div class="headtext">Fotos</div>
  <div class="photos-container">
    
    
    
    
    
    
  </div>
</section>
<!-- FINAL DO CÓDIGO HTML - PHOTOS -->
```

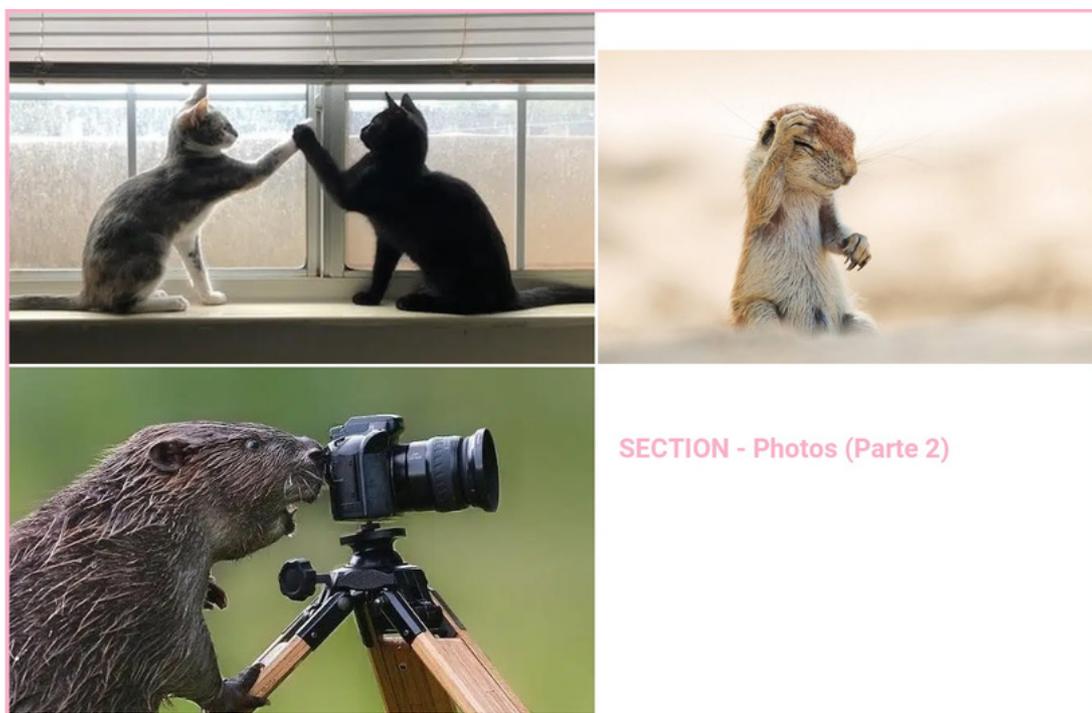
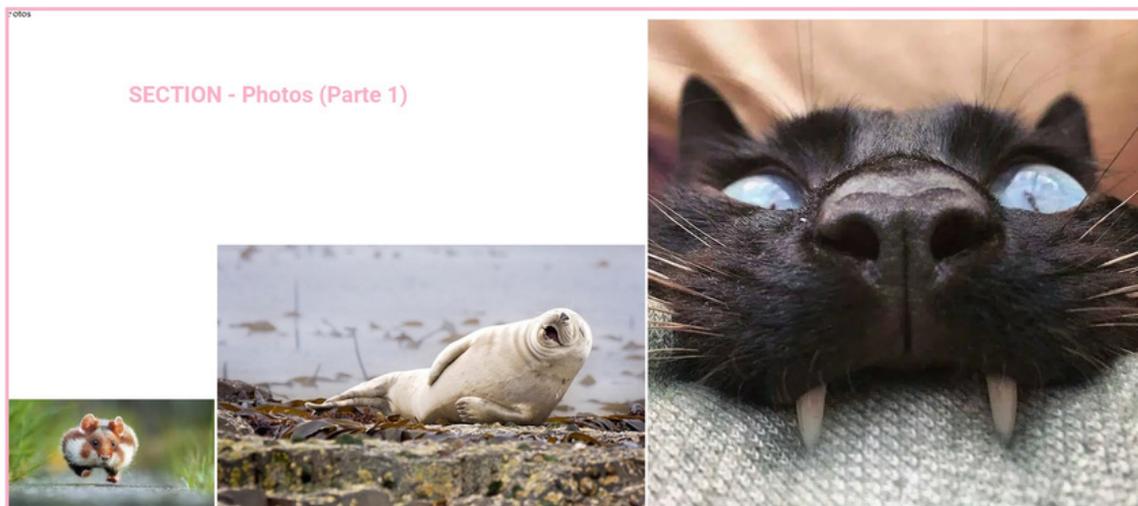
Nas 2 próximas imagens podemos ver o resultado da estrutura da SECTION (Photos) sendo renderizada (mostrada) no navegador (Google Chrome). Assim como aconteceu quando fizemos a SECTION (Clients) como temos 6 imagens e elas não cabem em uma única linha, automaticamente a primeira imagem que não cabe é colocada na linha abaixo e assim vai até a última imagem ser renderizada no navegador.

Notem também que nas 6 imagens eu coloquei a mesma classe com o nome `class="photos-image"` (isso aconteceu porque como cada imagem tem um tamanho diferente quando formos estilizar essa classe `"photos-image"`... todas as imagens receberão os valores desse estilo e ficarão em um tamanho padrão não sendo necessário estilizarmos uma por uma.



[/igor-rebolla](#)

Explicação de Cada Elemento(Tag) do Código HTML Site (SECTION - Depoimentos)



SECTION - Photos (Parte 2)

Explicação de Cada Elemento(Tag) do Código HTML do Site (SECTION - Photos)

Até aqui estruturamos a NAVBAR, o HEADER, a SECTION CLIENTS, a SECTION PRODUCTS, a SECTION SERVICES, a SECTION DEPOIMENTOS e a SECTION PHOTOS do nosso site, vamos partir para a oitava e penúltima estrutura do nosso site que será:

8° - FOOTER (Rodapé)

Então bora ver esse tal de FOOTER (Rodapé) já na próxima página....



/igor-rebolla

Código HTML do Site FOOTER (Rodapé)

Aqui é onde vamos iniciar toda a estrutura de códigos HTML que compõem o FOOTER (Rodapé). O objetivo final depois de estilizarmos (style.css) o Footer (Rodapé) é esse abaixo:

Copyright © 2022 - Todos os Direitos Reservados
Desenvolvido por um aluno do Curso FrontEnd do Bem



/igor-rebolla

Código HTML do Site (FOOTER - Rodapé)

O início do código do FOOTER (Rodapé) será feito logo abaixo do fechamento do comentário da SECTION (Photos).

```
<!-- FINAL DO CÓDIGO HTML - PHOTOS -->  
  
<!-- INICIO DO CÓDIGO HTML - FOOTER -->  
<footer id="rodape">  
  <p>Copyright © 2022 - Todos os Direitos Reservados</p>  
  <p>Desenvolvido por um aluno do Curso FrontEnd do Bem</p>  
</footer>  
  
<!-- FINAL DO CÓDIGO HTML - FOOTER -->
```



/igor-rebolla

Explicação de Cada Elemento(Tag) do Código HTML do Site (FOOTER - Rodapé)

```
<!-- INICIO DO CÓDIGO HTML - FOOTER (Rodapé) -->
```

Comentário indicando onde é o início do Código HTML do FOOTER (Rodapé) (Só reforçando... esse comentário não é renderizado pelo navegador).

```
<footer id="rodape">
```

Aqui criamos um footer com o id que recebe "rodape". O footer terá um identificador(id) único e o "rodape" será utilizado para estilizarmos essa section quando formos trabalhar dentro do arquivo style.css

O nosso footer está composto por apenas 2 parágrafos.

```
<p>Copyright © 2022 - Todos os Direitos Reservados</p>
```

```
<p>Desenvolvido por um aluno do Curso FrontEnd do Bem</p>
```

```
<!-- FINAL DO CÓDIGO HTML - FOOTER (Rodapé) -->
```

Comentário indicando onde é o final do Código HTML do FOOTER (Rodapé) (Só reforçando... esse comentário não é renderizado pelo navegador).



/igor-rebolla

Explicação de Cada Elemento(Tag) do Código HTML do Site (FOOTER - Rodapé)

Agora vamos ver na imagem abaixo como toda essa estrutura do FOOTER (rodapé) ficará dentro do VsCode.

```
<!-- FINAL DO CÓDIGO HTML - PHOTOS -->

<!-- INICIO DO CÓDIGO HTML - FOOTER (Rodapé) -->
<footer id="rodape">
  <p>Copyright © 2022 - Todos os Direitos Reservados</p>
  <p>Desenvolvido por um aluno do Curso FrontEnd do Bem</p>
</footer>
<!-- FINAL DO CÓDIGO HTML - FOOTER (Rodapé) -->
```

Agora vamos ver na imagem abaixo como toda essa estrutura do FOOTER (rodapé) ficará dentro do VsCode.

Copyright © 2022 - Todos os Direitos Reservados	FOOTER - Rodapé
Desenvolvido por um aluno do Curso FrontEnd do Bem	

UFA!!! Chegamos ao final da nossa estrutura HTML... agora vamos aplicar os estilos dessa estrutura através do arquivo style.css.

Então bora começar a estilizar o nosso site Pet Shop do Bem Humorado já na próxima página...

Estrutura Básica do Arquivo Style.css

Antes de iniciarmos precisamos fazer com o que o arquivo style.css faça a comunicação com o index.html e vice-versa.

Como no início desse e-book nós já criamos o nosso arquivo style.css o que precisamos fazer para que essa comunicação aconteça é irmos até o arquivo index.html e dentro da tag <head> colocarmos a seguinte linha de código:

```
<link rel="stylesheet" href="style.css">
```

Agora vamos ver na imagem abaixo como essa linha de código ficará dentro do VsCode.

```
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Pet Shop do Bem Humorado</title>
  <script src="https://kit.fontawesome.com/1b56f1d0e7.js" crossorigin="anonymous"></script>
  <link rel="stylesheet" href="style.css">
</head>
```

Outra coisa que vamos fazer aqui é buscar uma Fonte Externa para usarmos em nosso site.

Bora fazer isso já na próxima página...

Buscando uma Fonte Externa no Google Fonts

Assim quando usamos o FONT AWESOME para buscamos ícones para o nosso site, aqui vamos também até um site externo para buscar uma nova FONTE para usarmos em nosso site. Vou mencionar aqui duas das fontes mais conhecidas (ARIAL e TIMES NEW ROMAN).

"Tá Igor e qual é o objetivo de mostrar isso aqui?"

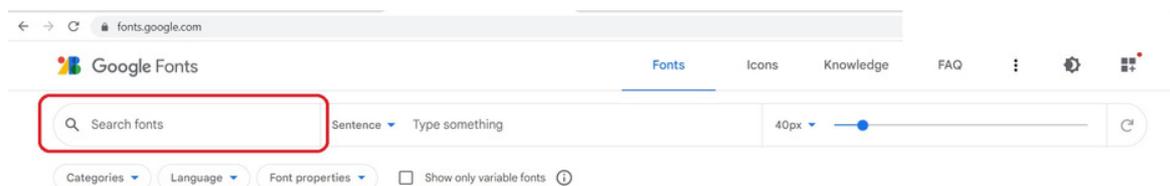
Eu respondo que são basicamente para 2 motivos:

- 1 - Mostrar para você como buscamos uma fonte externa para usarmos em um site.
- 2- Caso o cliente não tenha uma fonte específica e queira algo diferente você saiba onde ir procurar.

O site que vamos usar para buscar essa(s) fonte(s) é o GOOGLE FONTS:

<https://fonts.google.com/>

Ao digitarmos o site acima uma tela com um campo Search Fonts(Pesquisar fontes) irá aparecer.



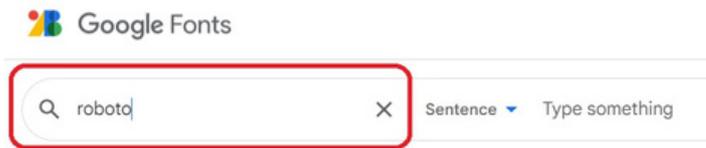
Vamos pesquisar pela fonte a qual queremos. Essa fonte é a Roboto.

Dentro do campo Search Fonts(Pesquisar fontes) digitaremos:

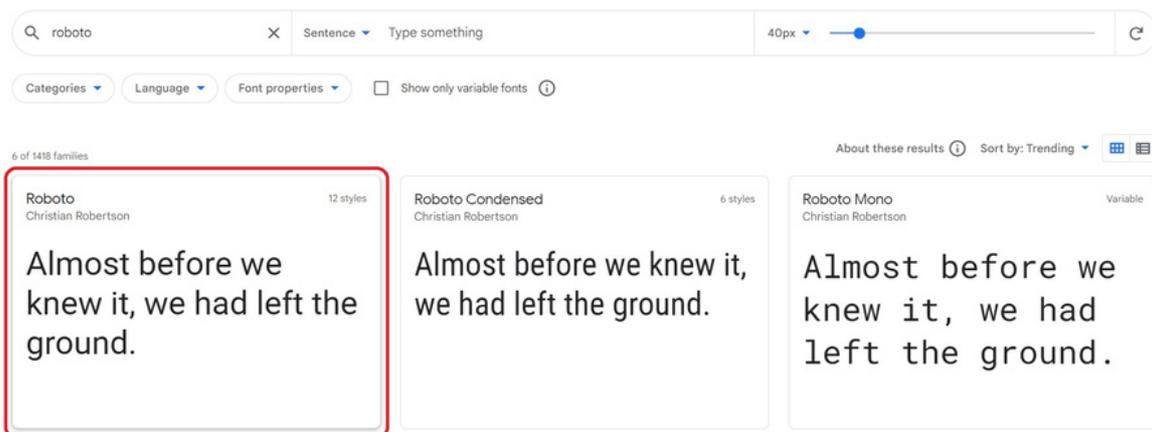
- Roboto e pressionaremos enter.



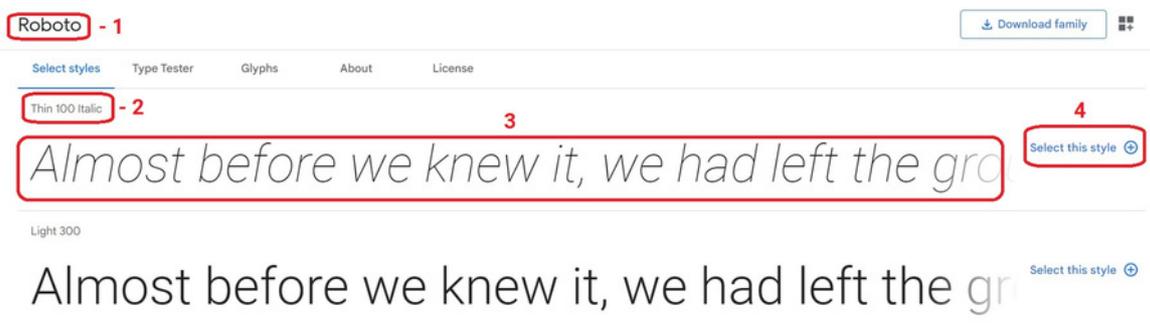
Buscando uma Fonte Externa no Google Fonts



Agora que efetuamos a pesquisa da fonte (algumas de opções da Fonte Roboto) foram mostradas na tela para escolhermos aquela que usaremos no desenvolvimento do nosso site. Aqui vamos escolher a primeira "Roboto". Notem que ao colocarmos a seta do mouse sobre essa primeira opção (surge um leve sombreamento).



Vamos clicar 1x sobre a caixa da fonte a qual selecionamos e ao clicarmos uma nova tela aparecerá com os estilos da Fonte Roboto.



Buscando uma Fonte Externa no Google Fonts

Notem que na figura acima temos 4 caixas que nos mostram algumas informações importantes. E são elas:

- 1 - A Fonte em si que é a Roboto;
- 2 - O Estilo da fonte aqui é a Thin 100 Italic (Ela é uma fonte mais fina que está inclinada por conta do itálico).
- 3 - Como a fonte será mostrada na tela
- 4 - Select this style (Selecione o seu estilo) é onde vamos clicar para selecionar os estilos das fontes os quais vamos usar em nosso site.

Agora vamos selecionar todos os estilos da Fonte Roboto que vamos usar em nosso site.

Light 300:

Light 300

Almost before we knew it, we had left the gr Remove this style

Medium 500:

Medium 500

Almost before we knew it, we had left the g Remove this style

Bold 700:

Bold 700

Almost before we knew it, we had left the g Remove this style



Buscando uma Fonte Externa no Google Fonts

Black 900:

Black 900

Almost before we knew it, we had left the

Remove this style

Agora que selecionamos os 4 estilos para serem usados em nosso site. Precisamos buscar a linha de código a qual usaremos dentro do arquivo style.css (é essa linha que irá "habilitar") o uso da Roboto.

Ao lado direito de Download Family (temos um ícone representado por três quadradinhos e um sinal de mais (Aqui eu estou usando a versão 11 do Windows - caso a sua versão seja diferente dessa aqui um sinal de + vai aparecer aqui no lugar). Vamos posicionar o ponteiro do mouse sobre esse ícone até aparecer View selected families (Ver as famílias selecionadas) conforme imagem abaixo

Roboto

Download family



View selected families

Select styles

Type Tester

Glyphs

About

License

Thin 100 Italic

Almost before we knew it, we had left the gro

Select this style

Vamos clicar sobre esse ícone e a próxima tela irá nos mostrar um resumo das Famílias da Fonte Roboto as quais selecionamos nos passos anteriores.

Buscando uma Fonte Externa no Google Fonts

Selected family  

Review

Roboto 

Light 300 

Medium 500 

Bold 700 

Black 900 

To embed a font, copy the code into the <head> of your html

<link> @import

```
<style>
@import url('https://fonts.googleapis.com/css2?family=Roboto:wght@300;500;700;900&display=swap');
</style>
```

Buscando uma Fonte Externa no Google Fonts

Agora que vimos o resumo dos estilos das fontes e está tudo OK. Conforme imagem abaixo, vamos selecionar o código o qual queremos para que possa ser usado no nosso arquivo style.css.

1 - Clicar em @import

2 - Copiar todo o código dentro da tag <style></style>.

Obs.: A Tag style não é copiada junto.

To embed a font, copy the code into the <head> of your html



<link>



@import

1 - Vamos clicar em @import

```
<style>
```

```
@import url('https://fonts.googleapis.com/css2?family=Roboto:wght@300;500;700;900&display=swap');
```

```
</style>
```

2 - Vamos copiar todo esse código entre a tag <style></style>

Essa é a linha a qual acabamos de copiar:

```
@import url('https://fonts.googleapis.com/css2?family=Roboto:wght@300;500;700;900&display=swap');
```

Agora vamos colar a linha copiada acima dentro do nosso arquivo style.css. A qual ficará assim dentro do VSCode.

```
style.css x
1 @import url('https://fonts.googleapis.com/css2?family=Roboto:wght@300;500;700;900&display=swap');
2
```

Definindo alguns Resets do Arquivo Style.css

Fonte selecionada e sua importação devidamente feita dentro do arquivo style.css. Vamos iniciar com alguns "Resets".

```
* {  
  margin: 0;  
  padding: 0;  
  box-sizing: border-box;  
}
```

Explicação do código acima:

*

Significa que o reset será aplicado de forma global (para todo o site)

margin: 0;

Iniciamos a margin com 0;

padding: 0;

Iniciamos o padding com 0;

box-sizing: border-box;

Ele desconsidera o margin e o padding no momento de contar a altura e/ou largura de um elemento

Agora veremos como o código acima ficará dentro do VSCode.



/igor-rebolla

Definindo alguns Resets do Arquivo Style.css

```
3 * {
4   margin: 0;
5   padding: 0;
6   box-sizing: border-box;
7 }
```

Agora vamos informar dentro do arquivo style.css (caso a font-family 'roboto') não seja renderizada no navegador uma fonte alternativa seja mostrada no lugar no caso uma padrão que é a sans-serif. Dentro da tag <body> que é o local onde digitamos toda nossa estrutura do código ficará assim:

```
body {
  font-family: 'Roboto', sans-serif;
}
```

Agora veremos como o código acima ficará dentro do VSCode.

```
body {
  font-family: 'Roboto', sans-serif;
}
```

E o nosso último reset será aplicado dentro da tag a (por padrão ela vem com um sublinhado e não queremos que esse sublinhado apareça em nosso site. Vamos aproveitar também para mudar a cor dos itens da . A estrutura do código ficará assim:



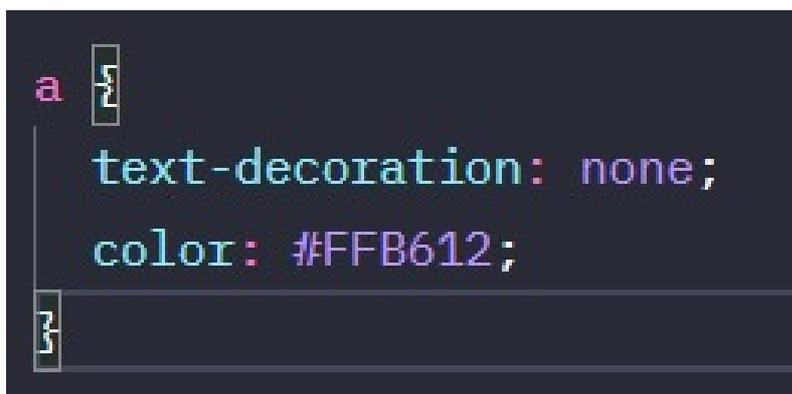
Definindo alguns Resets do Arquivo Style.css

```
a {  
  text-decoration: none;  
  color: #FFB612;  
}
```

text-decoration: none;
None faz com o sublinhado seja retirado abaixo do texto

color: #FFB612;
Foi definida uma cor que é a #FFB612

Agora veremos como o código acima ficará dentro do VSCode.



```
a {  
  text-decoration: none;  
  color: #FFB612;  
}
```

Resets OK - E agora chegou o momento de estilizarmos a NAVBAR que a qual foi estruturada primeiro dentro do nosso arquivo index.html.

Bora estilizar a NAVBAR

Aplicando Estilos - NAVBAR

navbar

Só lembrando que o objetivo final depois de aplicarmos todos os estilos da nossa NAVBAR será esse abaixo:



Para chegarmos nesse resultado temos que estilizar as seguintes estruturas da NAVBAR que estão vindo do arquivo index.html. E essas estruturas são:

1 - No index.html foi estruturado assim:

```
<nav id="navbar">
```

1 - No style.css vamos usar essa estrutura assim:

```
#navbar
```

2 - No index.html foi estruturado assim:

```
<div class="navbar-container">
```

2 - No style.css vamos usar essa estrutura assim:

```
.navbar-container
```

3 - No index.html foi estruturado assim:

```
<h1 class="logo">
```

3 - No style.css vamos usar essa estrutura assim:

```
.logo
```



Aplicando Estilos - NAVBAR

navbar

4 - No index.html foi estruturado assim:

```
<ul class="navbar-items">
```

4 - No style.css vamos usar essa estrutura assim:

```
.navbar-items
```

5 - No index.html foi estruturado assim:

```
<ul class="navbar-items"> <li>
```

5 - No style.css vamos usar essa estrutura assim:

```
.navbar-items li
```

Na próxima página vamos definir o estilo css para a nossa primeira estrutura da NAVBAR que é a:

1 - No index.html foi estruturado assim:

```
<nav id="navbar">
```

1 - No style.css vamos usar essa estrutura assim:

```
#navbar
```



Explicação de Cada Propriedade do Estilo CSS aplicada na NAVBAR (1 #navbar)

```
</* INICIO NAVBAR - LOGO + LINKS (INICIO, CLIENTES, FOTOS) */  
Comentário indicando onde é o início do Código CSS da NAVBAR  
(Não é renderizado pelo navegador)
```

```
<nav id="navbar"> Essa é a referência que veio do arquivo index.html  
para usarmos aqui no arquivo style.css temos que usar a #(que  
identifica um id) + o nome desse id(navbar). Portanto aqui no style.css  
vai ficar assim: #navbar
```

```
#navbar {
```

position: fixed;

Define que a nossa Navbar irá permanecer fixa na tela.

width: 100%;

Indica que a largura para esse elemento pai (no caso aqui o body) seja definida em 100% da largura da tela.

height: 80px;

Indica que a altura para esse elemento será de 80px

background: black;

Indica que a cor de fundo da nossa #navbar seja preta.

color: #FFB612;

indica que a cor da fonte dentro da nossa #navbar seja #FFB612

```
}
```



/igor-rebolla

Explicação de Cada Propriedade do Estilo CSS aplicada na NAVBAR (1 #navbar)

Agora veremos como o código da página anterior ficará dentro do VSCode.

```
/* INICIO NAVBAR - LOGO + LINKS (INICIO, CLIENTES, FOTOS) */  
#navbar {  
  position: fixed;  
  width: 100%;  
  height: 80px;  
  background: black;  
  color: #FFB612;  
}
```

Agora veremos como as 5 propriedades de estilo as quais aplicamos acima vão ficar quando são renderizadas no navegador



- 1 - Indica que a largura para esse elemento pai (no caso aqui o body) seja definida em 100% da largura da tela.
- 2 - Indica que a altura para esse elemento será de 80px
- 3 - Indica que a cor de fundo da nossa #navbar seja preta.
- 4 - Indica que a cor da fonte dentro da nossa #navbar seja #FFB612

Na próxima página vamos definir o estilo css para a nossa segunda estrutura da NAVBAR que é a:

2 - No index.html foi estruturado assim:

```
<div class="navbar-container">
```

2 - No style.css vamos usar essa estrutura assim:

```
.navbar-container
```

Explicação de Cada Propriedade do Estilo CSS aplicada na NAVBAR (2 .navbar-container)

`<div class="navbar-container">` Essa é a referência que veio do arquivo index.html para usarmos aqui no arquivo style.css temos que usar o ponto (.) que identifica uma classe + o nome dessa classe(navbar-container). Portanto aqui no style.css vai ficar assim:
`.navbar-container`

```
.navbar-container {
```

width: 80%;

Largura de 80%

height: 100%;

A Altura aqui foi definida em 100% para que ele seja 100% da Navbar

margin: auto;

Para alinhar automaticamente ao centro

display: flex;

Para que os elementos fiquem em linha

justify-content: space-between;

Aqui vai justificar o conteúdo com um espaço entre os elementos.

align-items: center;

Para alinharmos os itens ao centro

```
}
```



[/igor-rebolla](#)

Explicação de Cada Propriedade do Estilo CSS aplicada na NAVBAR (2 .navbar-container)

Agora veremos como o código da página anterior ficará dentro do VSCode.

```
.navbar-container {  
  width: 80%;  
  height: 100%;  
  margin: auto;  
  display: flex;  
  justify-content: space-between;  
  align-items: center;  
}
```

Agora veremos como as 6 propriedades de estilo as quais aplicamos acima vão ficar quando são renderizadas no navegador



Na próxima página vamos definir o estilo css para a nossa terceira estrutura da NAVBAR que é a:

3 - No index.html foi estruturado assim:

```
<h1 class="logo">
```

3 - No style.css vamos usar essa estrutura assim:

```
.logo
```



Explicação de Cada Propriedade do Estilo CSS aplicada na NAVBAR (3 .logo)

`<h1 class="logo">` Essa é a referência que veio do arquivo `index.html` para usarmos aqui no arquivo `style.css` temos que usar o ponto (.) que identifica uma classe + o nome dessa classe(`logo`). Portanto aqui no `style.css` vai ficar assim: `.logo`

```
.logo {
```

font-size: 2rem;

Aqui o tamanho da fonte foi definido em `2rem` (cada `rem` tem `16px` o equivalente a `32px`;

```
}
```

Agora veremos como o código da página anterior ficará dentro do VSCode.

```
.logo {  
  font-size: 2rem;  
}
```

Agora veremos na imagem da próxima página como essa propriedade de estilo a qual foi aplicada acima vai ficar quando renderizada no navegador.



[/igor-rebolla](#)

Explicação de Cada Propriedade do Estilo CSS aplicada na NAVBAR (3 .logo)



1 - O tamanho dessa fonte agora está em 2rem ou 32px

Na próxima página vamos definir o estilo css para a nossa quarta estrutura da NAVBAR que é a:

4 - No index.html foi estruturado assim:

```
<ul class="navbar-items">
```

4 - No style.css vamos usar essa estrutura assim:

```
.navbar-items
```



Explicação de Cada Propriedade do Estilo CSS aplicada na NAVBAR (4 .navbar-items)

`<ul class="navbar-items">` Essa é a referência que veio do arquivo `index.html` para usarmos aqui no arquivo `style.css` temos que usar o ponto (.) que identifica uma classe + o nome dessa classe (`navbar-items`). Portanto aqui no `style.css` vai ficar assim: `.navbar-items`

```
.navbar-items {
```

display: flex;

Para que os elementos fiquem em linha

list-style: none;

Para tirar aquelas bolinhas de marcação

text-transform: uppercase;

Transforma todo o texto em letras maiúsculas

```
}
```

Agora veremos como o código acima ficará dentro do VSCode.

```
.navbar-items {  
  display: flex;  
  list-style: none;  
  text-transform: uppercase;  
}
```

Explicação de Cada Propriedade do Estilo CSS aplicada na NAVBAR (4 .navbar-items)

Agora veremos como as 3 propriedades de estilo as quais aplicamos acima vão ficar quando são renderizadas no navegador

Paw Shop do Bem Humorado

INICIOCLIENTESFOTOS

Na próxima página vamos definir o estilo css para a nossa quinta e última estrutura da NAVBAR que é a:

5 - No index.html foi estruturado assim:

```
<ul class="navbar-items"> <li>
```

5 - No style.css vamos usar essa estrutura assim:

```
.navbar-items li
```



Explicação de Cada Propriedade do Estilo CSS aplicada na NAVBAR (5 .navbar-items li)

`<ul class="navbar-items"> ` Essa é a referência que veio do arquivo `index.html` para usarmos aqui no arquivo `style.css` temos que usar o ponto (`.`) que identifica uma classe + o nome dessa classe (`navbar-items`) só que aqui tem um "pequeno detalhe" queremos estilizar as `li`'s que estão dentro da `"navbar-items"`. E como faremos isso já que as `li`'s não tem uma classe definida para elas? Vamos pegar a classe que faz referencia a `navbar-items` e em seguida colocar a tag `li`. Portanto aqui no `style.css` vai ficar assim: `.navbar-items li`

```
.navbar-items li {
```

padding: 12px;

Vamos definir um padding (espaçamento interno) entre cada `li` de 12px

font-size: 1.1rem;

Vamos definir o tamanho da fonte para 1.1rem

```
}
```

```
/* FINAL NAVBAR - LOGO + LINKS (INICIO, CLIENTES, FOTOS) */
```

Comentário indicando onde é o final do Código CSS da NAVBAR (Não é renderizado pelo navegador)



/igor-rebolla

Explicação de Cada Propriedade do Estilo CSS aplicada na NAVBAR (5 .navbar-items li)

Agora veremos como o código acima ficará dentro do VSCode.

```
.navbar-items li {  
  padding: 12px;  
  font-size: 1.1rem;  
}
```

Agora veremos como as 3 propriedades de estilo as quais aplicamos acima vão ficar quando são renderizadas no navegador

Paw Shop do Bem Humorado

INICIO CLIENTES FOTOS

E aqui chegamos no final da estilização da estrutura completa da NAVBAR.

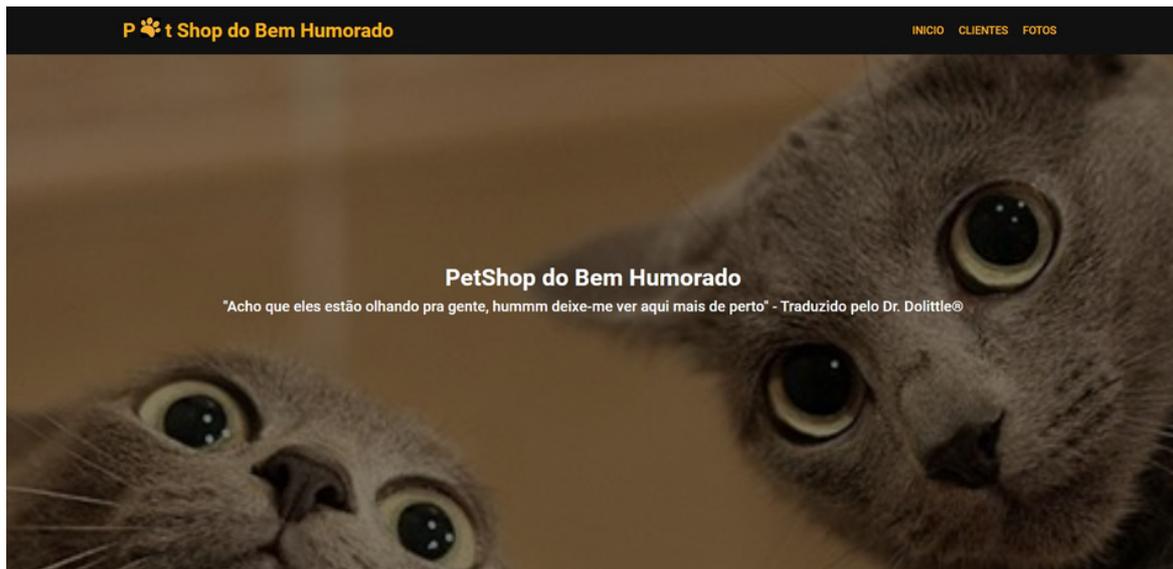
E agora chegou o momento de estilizarmos o HEADER (BANNER + h2 + Parágrafo)

Bora estilizar o HEADER (BANNER + h2 + Parágrafo) ...

Aplicando Estilos - HEADER

#banner

Só lembrando que o objetivo final depois de aplicarmos todos os estilos do nosso BANNER será esse abaixo:



Para chegarmos nesse resultado temos que estilizar as seguintes estruturas da NAVBAR que estão vindo do arquivo index.html. E essas estruturas são:

1 - No index.html foi estruturado assim:

```
<header id="banner">
```

1 - No style.css vamos usar essa estrutura assim:

```
#banner
```

2 - No index.html foi estruturado assim:

```
<div class="banner-container">
```

2 - No style.css vamos usar essa estrutura assim:

```
.banner-container
```



Aplicando Estilos - HEADER

banner

3 - No index.html foi estruturado assim:

```
<div class="banner-container"> <h2>
```

3 - No style.css vamos usar essa estrutura assim:

```
.banner-container h2
```

4 - No index.html foi estruturado assim:

```
<div class="banner-container"> <p>
```

4 - No style.css vamos usar essa estrutura assim:

```
.banner-container p
```

Na próxima página vamos definir o estilo css para a nossa primeira estrutura do BANNER que é a:

1 - No index.html foi estruturado assim:

```
<header id="banner">
```

1 - No style.css vamos usar essa estrutura assim:

```
#banner
```



Explicação de Cada Propriedade do Estilo CSS aplicada no HEADER (1 #banner)

```
/* INICIO DO BANNER - Imagem dos 2 Gatinhos olhando */  
Comentário indicando onde é o início do Código CSS do HEADER -  
Banner (Não é renderizado pelo navegador)
```

`<header id="banner">` Essa é a referência que veio do arquivo `index.html` para usarmos aqui no arquivo `style.css` temos que usar a `#` (que identifica um id) + o nome desse id(`navbar`). Portanto aqui no `style.css` vai ficar assim: `#banner`

```
#banner {
```

```
height: 100vh;
```

Estou definindo que o banner vai ocupar em 100vh a visão do usuário na altura.

```
display: flex;
```

Para que os elementos fiquem em linha

```
justify-content: center;
```

Vamos justificar o conteúdo ao centro

```
align-items: center;
```

Vamos alinhar o conteúdo ao centro

```
color: white;
```

Vamos definir a cor da fonte como branca

```
background: url('/images/banner.jpg') no-repeat center/cover;
```

A url aqui faz referência a imagem `banner.jpg` dentro da pasta `images`. No `repeat center/cover` (aqui estamos informando caso a imagem não caiba na tela que ela não se repita e que ao centro cobrindo toda a tela.

```
}
```



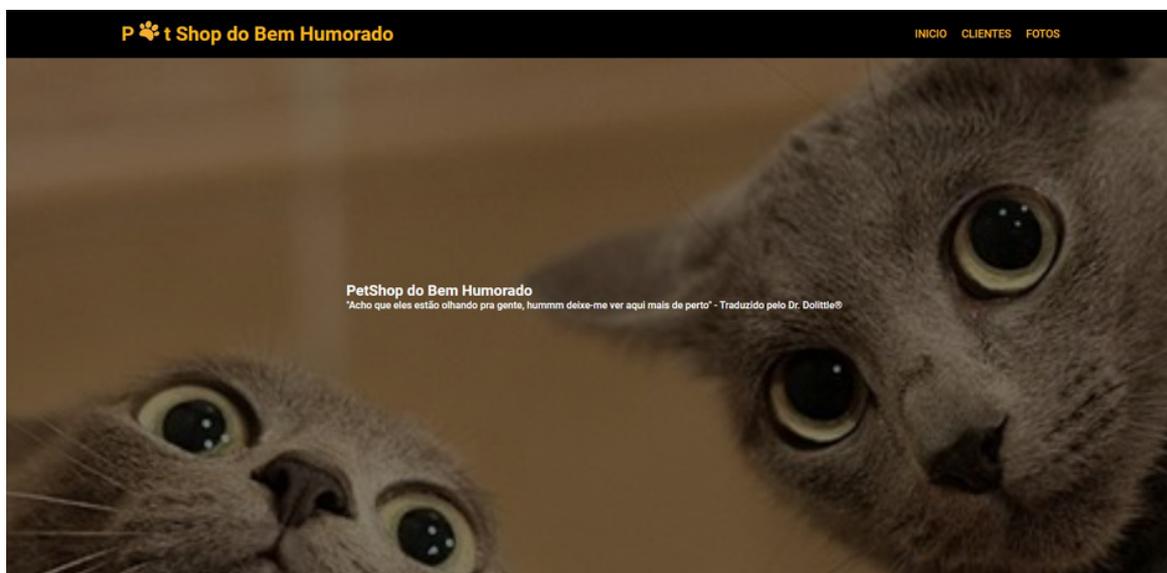
/igor-rebolla

Explicação de Cada Propriedade do Estilo CSS aplicada no HEADER (1 #banner)

Agora veremos como o código acima ficará dentro do VSCode.

```
#banner {  
  height: 100vh;  
  display: flex;  
  justify-content: center;  
  align-items: center;  
  color: white;  
  background: url('/images/banner.jpg') no-repeat center/cover;  
}
```

Agora veremos como as 6 propriedades de estilo as quais aplicamos acima vão ficar quando são renderizadas no navegador



Explicação de Cada Propriedade do Estilo CSS aplicada no HEADER (1 #banner)

Na próxima página vamos definir o estilo css para a nossa segunda estrutura do BANNER que é a:

2 - No index.html foi estruturado assim:

```
<div class="banner-container">
```

2 - No style.css vamos usar essa estrutura assim:

```
.banner-container
```



Explicação de Cada Propriedade do Estilo CSS aplicada no HEADER (2 .banner-container)

`<div class="banner-container">` Essa é a referência que veio do arquivo `index.html` para usarmos aqui no arquivo `style.css` temos que usar o ponto (.) que identifica uma classe + o nome dessa classe (`banner-container`). Portanto aqui no `style.css` vai ficar assim:
`.banner-container`

```
.banner-container {
```

text-align: center;

Aqui alinharemos o texto do `banner-container` ao Centro

```
}
```

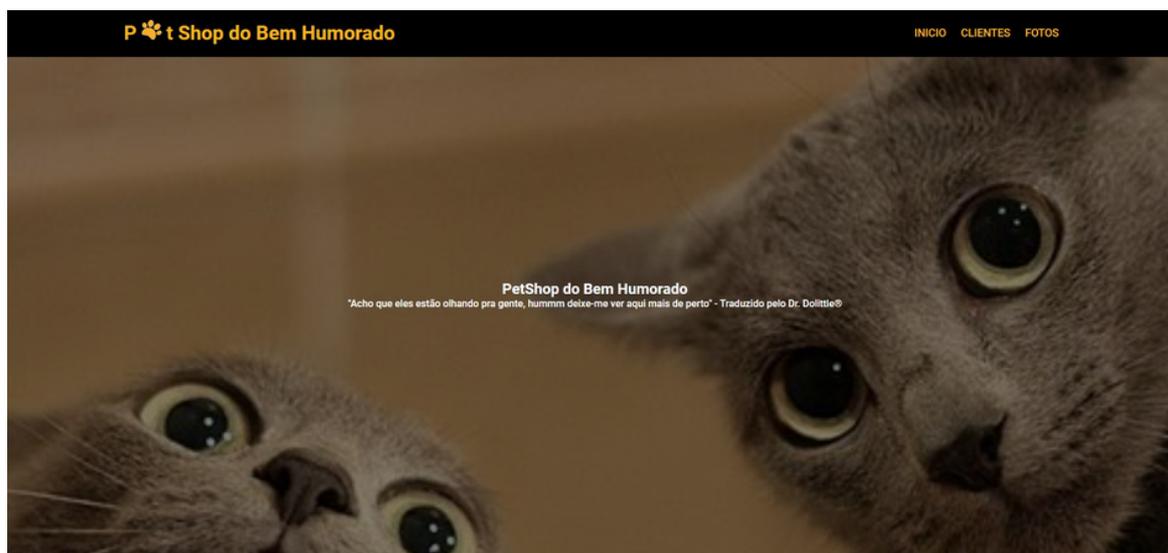
Agora veremos como o código acima ficará dentro do VSCode.

```
.banner-container {  
  text-align: center;  
}
```

Agora veremos como as 3 propriedades de estilo as quais aplicamos acima vão ficar quando são renderizadas no navegador

Explicação de Cada Propriedade do Estilo CSS aplicada no HEADER (2 .banner-container)

Agora veremos como as 3 propriedades de estilo as quais aplicamos acima vão ficar quando são renderizadas no navegador



Na próxima página vamos definir o estilo css para a nossa terceira estrutura do BANNER que é a:

3 - No index.html foi estruturado assim:

```
<div class="banner-container"> <h2>
```

3 - No style.css vamos usar essa estrutura assim:

```
.banner-container h2
```

Explicação de Cada Propriedade do Estilo CSS aplicada no HEADER

(2 .banner-container h2)

`<div class="banner-container"> <h2>` Essa é a referência que veio do arquivo `index.html` para usarmos aqui no arquivo `style.css` temos que usar o ponto (`.`) que identifica uma classe + o nome dessa classe (`banner-container`). só que aqui tem um "pequeno detalhe" queremos estilizar o `<h2>` que está dentro do "banner-container". E como faremos isso já que a `<h2>` não tem uma classe definida para elas? Vamos pegar a classe que faz referencia a `banner-container` e em seguida colocar a tag `h2`. Portanto aqui no `style.css` vai ficar assim: `.banner-container h2`

```
.banner-container h2 {
```

```
  font-size: 2.4rem;
```

```
  Definimos o tamanho do h2 para 2.4rem
```

```
}
```

Agora veremos como o código acima ficará dentro do VSCode.

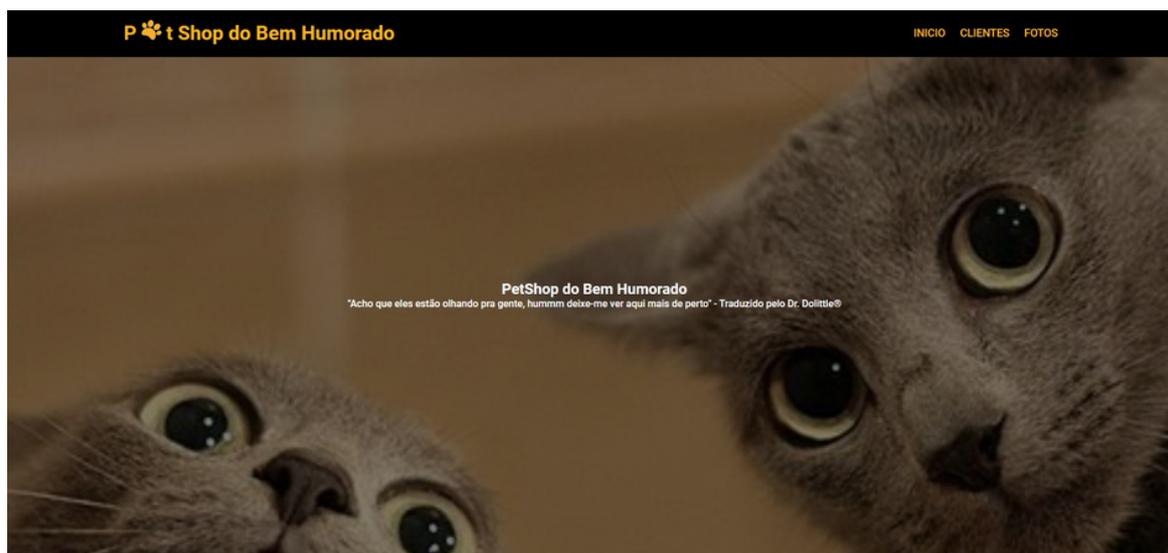


Explicação de Cada Propriedade do Estilo CSS aplicada no HEADER

(2 .banner-container h2)

```
.banner-container {  
  text-align: center;  
}
```

Agora veremos na imagem da próxima página como essa propriedade de estilo a qual foi aplicada acima vai ficar quando renderizada no navegador.



Na próxima página vamos definir o estilo css para a nossa quarta estrutura do BANNER que é a:

4 - No index.html foi estruturado assim:

```
<div class="banner-container"> <p>
```

4 - No style.css vamos usar essa estrutura assim:

```
.banner-container p
```



Explicação de Cada Propriedade do Estilo CSS aplicada no HEADER

(2 .banner-container p)

`<div class="banner-container"> <p>` Essa é a referência que veio do arquivo `index.html` para usarmos aqui no arquivo `style.css` temos que usar o ponto (`.`) que identifica uma classe + o nome dessa classe (`banner-container`). só que aqui tem um "pequeno detalhe" queremos estilizar o `<p>` que está dentro do "banner-container". E como faremos isso já que a `<p>` não tem uma classe definida para elas? Vamos pegar a classe que faz referencia a `banner-container` e em seguida colocar a tag `p`. Portanto aqui no `style.css` vai ficar assim:

```
.banner-container p {
```

font-size: 1.5rem;

Definimos o tamanho do `p` para `1.5rem`

padding: 9px 0;

Definimos um `padding` de `9px` (para cima e para baixo e `0` para esquerda e direita.

```
}
```

```
/* FINAL DO BANNER - Imagem dos 2 Gatinhos olhando */
```

```
Comentário indicando onde é o final do Código CSS do HEADER - Banner (Não é renderizado pelo navegador)
```



/igor-rebolla

Explicação de Cada Propriedade do Estilo CSS aplicada no HEADER

(2 .banner-container p)

Agora veremos como o código acima ficará dentro do VSCode.

```
.banner-container p {  
  font-size: 1.5rem;  
  padding: 9px 0;  
}
```

/* FINAL DO BANNER - Imagem dos 2 Gatinhos olhando */

Agora veremos na imagem abaixo como essas 2 propriedades de estilo as quais foi aplicadas acima vão ficar quando renderizadas no navegador.



Explicação de Cada Propriedade do Estilo CSS aplicada no HEADER

(2 .banner-container p)

E aqui chegamos no final da estilização da estrutura completa do HEADER (BANNER + h2 + Parágrafo) .

E agora chegou o momento de estilizarmos a SECTION (Clients).

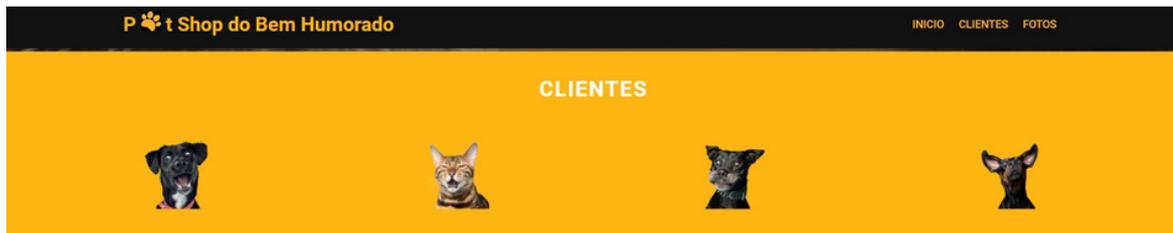
Bora estilizar a SECTION (Clients)



Aplicando Estilos - SECTION

#clients

Só lembrando que o objetivo final depois de aplicarmos todos os estilos da nossa SECTION (Clients) será esse abaixo:



Para chegarmos nesse resultado temos que estilizar as seguintes estruturas da SECTION (Clients) que estão vindo do arquivo index.html. E essas estruturas são:

1 - No index.html foi estruturado assim:

```
<section id="clients">
```

1 - No style.css vamos usar essa estrutura assim:

```
#banner
```

2 - No index.html foi estruturado assim:

```
<div class="headtext">Clientes</div>
```

2 - No style.css vamos usar essa estrutura assim:

```
.headtext
```

Aplicando Estilos - SECTION

#clients

3 - No index.html foi estruturado assim:

```
<div class="clients-container">
```

3 - No style.css vamos usar essa estrutura assim:

```
.clients-container
```

4 - No index.html foi estruturado assim:

```

```

```

```

```

```

```

```

4 - No style.css vamos usar essa estrutura assim:

```
.client
```

Na próxima página vamos definir o estilo css para a nossa primeira estrutura da SECTION (Clients) que é a:

1 - No index.html foi estruturado assim:

```
<section id="clients">
```

1 - No style.css vamos usar essa estrutura assim:

```
#clients
```



/igor-rebolla

Explicação de Cada Propriedade do Estilo CSS aplicada na SECTION (1 #clients)

`/* INICIO DOS CLIENTES - Imagem dos 3 Cachorros e 1 Gato */`
Comentário indicando onde é o início do Código CSS na SECTION - Clients (Não é renderizado pelo navegador)

`<section id="clients">` Essa é a referência que veio do arquivo `index.html` para usarmos aqui no arquivo `style.css` temos que usar a `#` (que identifica um id) + o nome desse id(`clients`). Portanto aqui no `style.css` vai ficar assim: `#clients`

```
#clients {
```

width: 100%;

Indica que a largura para esse elemento pai seja definida em 100% da largura da tela.

background: #FFB612;

indica que a cor de fundo dentro da Section `#clients` seja `#FFB612`

color: white;

Indica que a cor das fontes da nossa Section `#clients` seja branca.

padding: 40px;

Definimos um padding de 40px para (Top, Bottom, Right e Left)

```
}
```



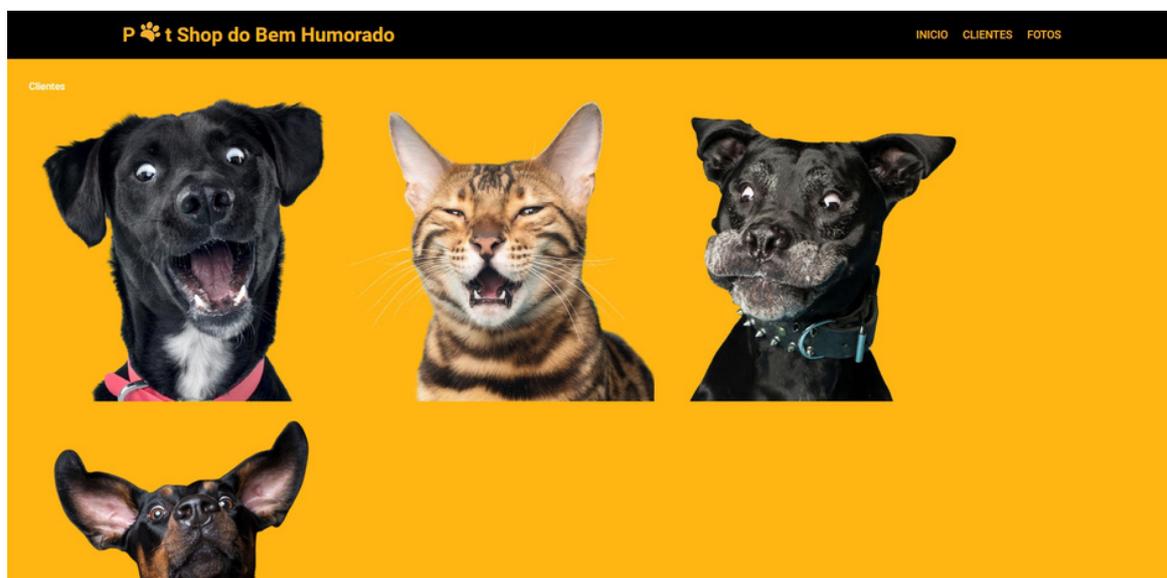
/igor-rebolla

Explicação de Cada Propriedade do Estilo CSS aplicada na SECTION Clients (1 #clients)

Agora veremos como o código acima ficará dentro do VSCode.

```
/* INICIO DOS CLIENTES - Imagem dos 3 Cachorros e 1 Gato */  
#clients {  
  width: 100%;  
  background: #FFB612;  
  color: white;  
  padding: 40px;  
}
```

Agora veremos na imagem abaixo como essas 4 propriedades de estilo as quais foram aplicadas acima vão ficar quando renderizadas no navegador.



Explicação de Cada Propriedade do Estilo CSS aplicada na SECTION Clients (1 #clients)

Na próxima página vamos definir o estilo css para a nossa estrutura da SECTION (Clients) que é a:

2 - No index.html foi estruturado assim:

```
<div class="headtext">Clientes</div>
```

2 - No style.css vamos usar essa estrutura assim:

```
.headtext
```

Explicação de Cada Propriedade do Estilo CSS aplicada na SECTION Clients (2 .headtext)

`<div class="headtext">Clientes</div>` Essa é a referência que veio do arquivo index.html para usarmos aqui no arquivo style.css temos que usar o ponto (.) que identifica uma classe + o nome dessa classe(headtext). Portanto aqui no style.css vai ficar assim: .headtext

```
.headtext {
```

font-size: 2.2rem;

Definimos o tamanho da fonte em 2.2rem

text-transform: uppercase;

Alteramos as letras para maiúsculas

font-weight: bold;

Negrito foi aplicado a fonte

text-align: center;

Alinhamos o texto no centro

padding-bottom: 50px;

Aplicamos um padding-bottom de 50px

letter-spacing: 2px;

Configuramos um espaçamento entre as letras de 2px.

```
}
```



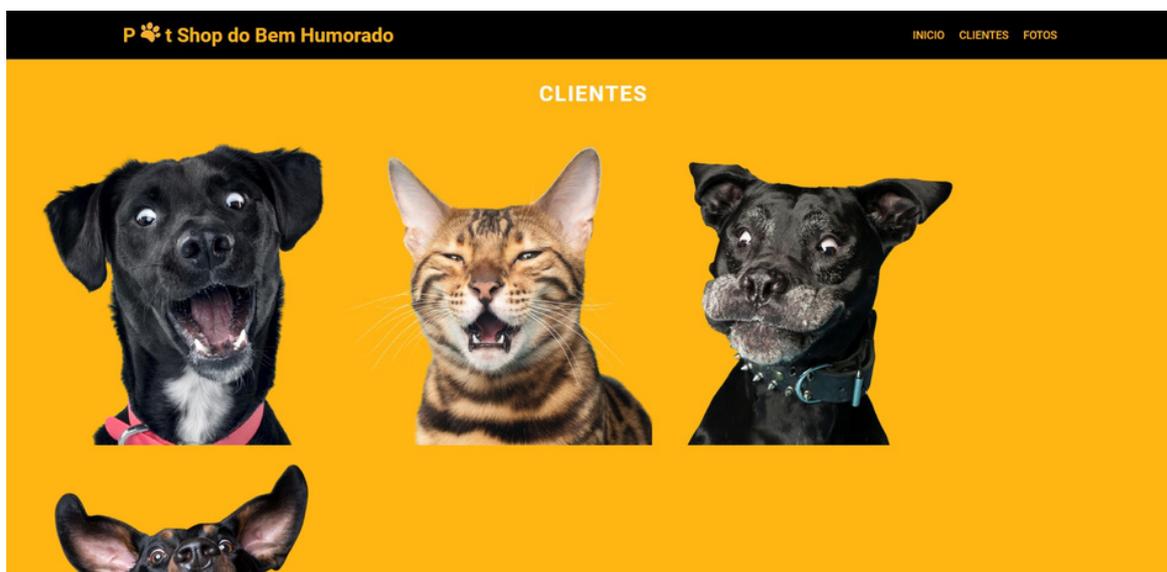
[/igor-rebolla](#)

Explicação de Cada Propriedade do Estilo CSS aplicada na SECTION Clients (2 .headtext)

Agora veremos como o código acima ficará dentro do VSCode.

```
.headtext {  
  font-size: 2.2rem;  
  text-transform: uppercase;  
  font-weight: bold;  
  text-align: center;  
  padding-bottom: 50px;  
  letter-spacing: 2px;  
}
```

Agora veremos na imagem abaixo como essas 6 propriedades de estilo as quais foram aplicadas acima vão ficar quando renderizadas no navegador.



Explicação de Cada Propriedade do Estilo CSS aplicada na SECTION Clients (3 .clients-container)

Na próxima página vamos definir o estilo css para a nossa terceira estrutura da SECTION (Clients) que é a:

3 - No index.html foi estruturado assim:

```
<div class="clients-container">
```

3 - No style.css vamos usar essa estrutura assim:

```
.clients-container
```

Explicação de Cada Propriedade do Estilo CSS aplicada na SECTION Clients (4 .client)

`` Essa é a referência que veio do arquivo `index.html` para usarmos aqui no arquivo `style.css` temos que usar o ponto (.) que identifica uma classe + o nome dessa classe (`client`) que é a classe que está segurando as 4 imagens dos nossos cliente. Portanto aqui no `style.css` vai ficar assim: `.client`.

```
.client {
```

display: inline-block;

Aqui queremos que a imagem seja considerada como um bloco

width: 25%;

Como temos 4 imagens (Aqui estamos falando que cada imagem será ajustada em 25% da largura da tela)

```
}
```

```
/* FINAL DOS CLIENTES - Imagem dos 3 Cachorros e 1 Gato */
```

Comentário indicando onde é o final do Código CSS na SECTION - Clients (Não é renderizado pelo navegador)

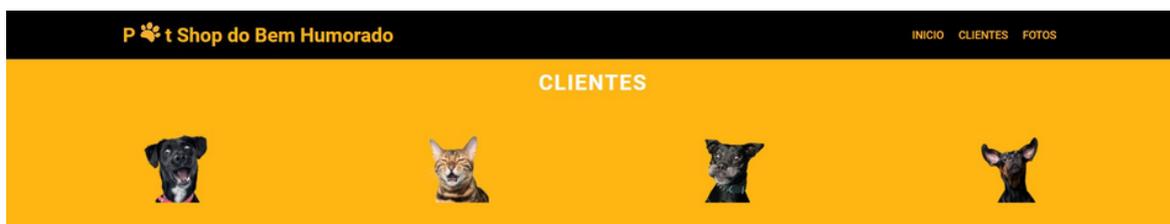


Explicação de Cada Propriedade do Estilo CSS aplicada na SECTION Clients (4 .client)

Agora veremos como o código acima ficará dentro do VSCode.

```
.clients-container {  
  display: grid;  
  grid-template-columns: repeat(4, 1fr);  
  justify-items: center;  
  padding: 12px;  
}
```

Agora veremos na imagem abaixo como essas 4 propriedades de estilo as quais foram aplicadas acima vão ficar quando renderizadas no navegador.



E aqui chegamos no final da estilização da estrutura completa do SECTION (Clients).

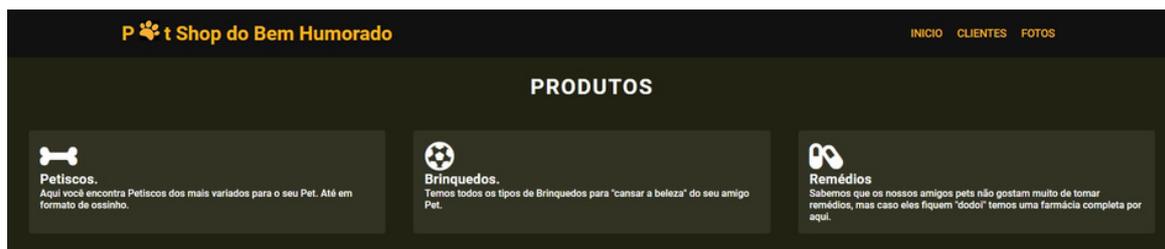
E agora chegou o momento de estilizarmos a SECTION (Products).

Bora estilizar a SECTION (Products)

Aplicando Estilos - SECTION

#products

Só lembrando que o objetivo final depois de aplicarmos todos os estilos da nossa SECTION (Products) será esse abaixo:



Para chegarmos nesse resultado temos que estilizar as seguintes estruturas da SECTION (Products) que estão vindo do arquivo index.html. E essas estruturas são:

1 - No index.html foi estruturado assim:

```
<section id="products">
```

1 - No style.css vamos usar essa estrutura assim:

```
#products
```

2 - No index.html foi estruturado assim:

```
<div class="headtext">Produtos</div>
```

2 - No style.css vamos usar essa estrutura assim:

```
.headtext
```

Aplicando Estilos - SECTION

#products

3 - No index.html foi estruturado assim:

```
<div class="products-container">
```

3 - No style.css vamos usar essa estrutura assim:

```
.products-container
```

4 - No index.html foi estruturado assim:

```
<div class="product">
```

4 - No style.css vamos usar essa estrutura assim:

```
.product
```

5 - No index.html foi estruturado assim:

```
<span class="product-title">Petiscos</span>
```

5 - No style.css vamos usar essa estrutura assim:

```
.product-title
```

Na próxima página vamos definir o estilo css para a nossa primeira estrutura da SECTION (Products) que é a:

1 - No index.html foi estruturado assim:

```
<section id="products">
```

1 - No style.css vamos usar essa estrutura assim:

```
#products
```



Explicação de Cada Propriedade do Estilo CSS aplicada na SECTION Products (1 #products)

```
/* INICIO DOS PRODUCTS - PRODUTOS */
```

Comentário indicando onde é o início do Código CSS na SECTION - Products (Não é renderizado pelo navegador)

`<section id="products">` Essa é a referência que veio do arquivo `index.html` para usarmos aqui no arquivo `style.css` temos que usar a `#` (que identifica um id) + o nome desse id(`products`). Portanto aqui no `style.css` vai ficar assim: `#products`

```
#products {
```

width: 100%;

Indica que a largura para esse elemento pai seja definida em 100% da largura da tela.

background: #221;

indica que a cor de fundo dentro da Section `#products` seja `#221`

color: white;

Indica que a cor das fontes da nossa Section `#products` seja branca.

padding: 40px;

Definimos um padding de 40px para (Top, Bottom, Right e Left)

```
}
```



[/igor-rebolla](#)

Explicação de Cada Propriedade do Estilo CSS aplicada na SECTION Products (1 #products)

Agora veremos como o código acima ficará dentro do VSCode.

```
/* INICIO DOS PRODUCTS - PRODUTOS */  
#products {  
  width: 100%;  
  background: #221;  
  color: white;  
  padding: 40px;  
}
```

Agora veremos na imagem abaixo como essas 4 propriedades de estilo as quais foram aplicadas acima vão ficar quando renderizadas no navegador.



Na próxima página vamos definir o estilo css para a nossa estrutura da SECTION (Produtos) que é a:

2 - No index.html foi estruturado assim:

```
<div class="headtext">Produtos</div>
```

2 - No style.css vamos usar essa estrutura assim:

```
.headtext
```



Explicação de Cada Propriedade do Estilo CSS aplicada na SECTION Products (2 .headtext)

`<div class="headtext">Produtos</div>` Essa é a referência que veio do arquivo `index.html` para usarmos aqui no arquivo `style.css` temos que usar o ponto (.) que identifica uma classe + o nome dessa classe(`headtext`). Portanto aqui no `style.css` vai ficar assim: `.headtext`

```
.headtext {
```

font-size: 2.2rem;

Definimos o tamanho da foto em 2.2rem

text-transform: uppercase;

Alteramos as letras para maiúsculas

font-weight: bold;

Negrito foi aplicado a fonte

text-align: center;

Alinhamos o texto no centro

padding-bottom: 50px;

Aplicamos um `padding-bottom` de 50px

letter-spacing: 2px;

Configuramos um espaçamento entre as letras de 2px.

```
}
```



[/igor-rebolla](#)

Explicação de Cada Propriedade do Estilo CSS aplicada na SECTION Products (2 .headtext)

Agora veremos como o código acima ficará dentro do VSCode.

```
.headtext {  
  font-size: 2.2rem;  
  text-transform: uppercase;  
  font-weight: bold;  
  text-align: center;  
  padding-bottom: 50px;  
  letter-spacing: 2px;  
}
```

Agora veremos na imagem abaixo como essas 6 propriedades de estilo as quais foram aplicadas acima vão ficar quando renderizadas no navegador.



Na próxima página vamos definir o estilo css para a nossa terceira estrutura da SECTION (Produtos) que é a:

3 - No index.html foi estruturado assim:

```
<div class="products-container">
```

3 - No style.css vamos usar essa estrutura assim:

```
.products-container
```



Explicação de Cada Propriedade do Estilo CSS aplicada na SECTION Products (3 .products-container)

`<div class="products-container">` Essa é a referência que veio do arquivo index.html para usarmos aqui no arquivo style.css temos que usar o ponto (.) que identifica uma classe + o nome dessa classe(products-container) que é a classe que está segurando os 3 ícones (Fonte Awesome) dos nossos Produtos, além de um spam e por último um parágrafo. Portanto aqui no style.css vai ficar assim:
.products-container

```
.products-container {
```

width: 100%;

Indica que a largura para esse elemento pai seja definida em 100%

margin: auto;

Para alinhar automaticamente ao centro

display: grid;

Aqui habilitamos para que o display seja grid

grid-template-columns: repeat(3, 1fr);

Aqui eu tenho 3 colunas que ocupam 1 fração do espaço

grid-column-gap: 45px;

Definimos um gap de 45px

```
}
```



/igor-rebolla

Explicação de Cada Propriedade do Estilo CSS aplicada na SECTION Products (3 .products-container)

Agora veremos como o código acima ficará dentro do VSCode.

```
.products-container {  
  width: 100%;  
  margin: auto;  
  display: grid;  
  grid-template-columns: repeat(3, 1fr);  
  grid-column-gap: 45px;  
}
```

Agora veremos na imagem abaixo como essas 5 propriedades de estilo as quais foram aplicadas acima vão ficar quando renderizadas no navegador.



Na próxima página vamos definir o estilo css para a nossa quarta estrutura da SECTION (Produtos) que é a:

4 - No index.html foi estruturado assim:

```
<div class="product">
```

4 - No style.css vamos usar essa estrutura assim:

```
.product
```



Explicação de Cada Propriedade do Estilo CSS aplicada na SECTION Products (4 .product)

`<div class="product">` Essa é a referência que veio do arquivo `index.html` para usarmos aqui no arquivo `style.css` temos que usar o ponto (.) que identifica uma classe + o nome dessa classe(`product`) que é a classe que está segurando os 3 ícones (Fonte Awesome) dos nossos Produtos, além de um `spam` e por último um parágrafo. Portanto aqui no `style.css` vai ficar assim: `.product`

```
.product {
```

display: flex;

para que os elementos fiquem um na frente do outro

flex-direction: column;

direção dos elementos estão em colunas

background: #332;

indica que a cor de fundo dentro de cada `product` seja `#332`

padding: 18px;

Definimos um `padding` de `18px` para (Top, Bottom, Right e Left)

border-radius: 4px;

Definimos uma borda arredondada em `4px`

```
}
```



/igor-rebolla

Explicação de Cada Propriedade do Estilo CSS aplicada na SECTION Products (4 .product)

Agora veremos como o código acima ficará dentro do VSCode.

```
.product {  
  display: flex;  
  flex-direction: column;  
  background: #332;  
  padding: 18px;  
  border-radius: 4px;  
}
```

Agora veremos na imagem abaixo como essas 5 propriedades de estilo as quais foram aplicadas acima vão ficar quando renderizadas no navegador.



Na próxima página vamos definir o estilo css para a nossa quinta estrutura da SECTION (Produtos) que é a:

5 - No index.html foi estruturado assim:

```
<span class="product-title">Remédios</span>
```

5 - No style.css vamos usar essa estrutura assim:

```
.product-title
```



Explicação de Cada Propriedade do Estilo CSS aplicada na SECTION Products (5 .product-title)

`<div class="product-title">` Essa é a referência que veio do arquivo `index.html` para usarmos aqui no arquivo `style.css` temos que usar o ponto (`.`) que identifica uma classe + o nome dessa classe (`product`) que é a classe que está segurando os 3 ícones (Fonte Awesome) dos nossos Produtos, além de um `spam` e por último um parágrafo. Portanto aqui no `style.css` vai ficar assim: `.product-title`

```
.product-title {
```

```
  font-size: 1.4rem;
```

Definimos o tamanho da fonte do título de cada produto em `1.4rem`

```
  font-weight: bold;
```

Definimos o negrito para o título de cada produto

```
}
```

```
/* FINAL DOS PRODUCTS - PRODUTOS */
```

Comentário indicando onde é o final do Código CSS na SECTION - Products (Não é renderizado pelo navegador)



/igor-rebolla

Explicação de Cada Propriedade do Estilo CSS aplicada na SECTION Products (5 .product-title)

Agora veremos como o código acima ficará dentro do VSCode.

```
.product-title {  
  font-size: 1.4rem;  
  font-weight: bold;  
}  
  
/* FINAL DOS PRODUCTS - PRODUTOS */
```

Agora veremos na imagem abaixo como essas 2 propriedades de estilo as quais foram aplicadas acima vão ficar quando renderizadas no navegador.



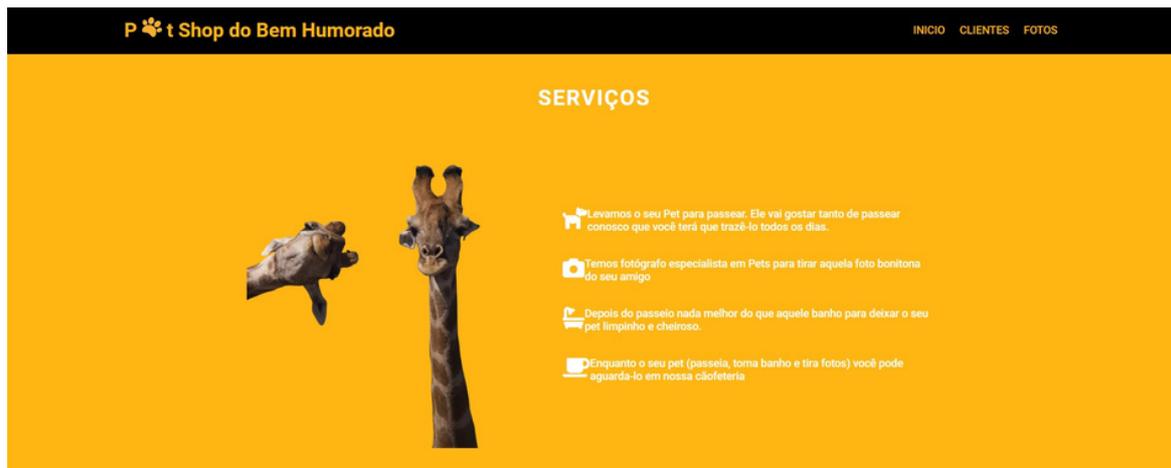
E aqui chegamos no final da estilização da estrutura completa do SECTION (Products).

E agora chegou o momento de estilizarmos a SECTION (Services)

Bora estilizar a SECTION (Services)

Aplicando Estilos - SECTION Services (1 #services)

Só lembrando que o objetivo final depois de aplicarmos todos os estilos da nossa SECTION (services) será esse abaixo:



Para chegarmos nesse resultado temos que estilizar as seguintes estruturas da SECTION (Services) que estão vindo do arquivo index.html. E essas estruturas são:

1 - No index.html foi estruturado assim:

```
<section id="services">
```

1 - No style.css vamos usar essa estrutura assim:

```
#services
```

2 - No index.html foi estruturado assim:

```
<div class="headtext">Serviços</div>
```

2 - No style.css vamos usar essa estrutura assim:

```
.headtext
```

Aplicando Estilos - SECTION Services (1 #services)

3 - No index.html foi estruturado assim:

```
<div class="services-container">
```

3 - No style.css vamos usar essa estrutura assim:

```
.services-container
```

4 - No index.html foi estruturado assim:

```
<div class="services">
```

4 - No style.css vamos usar essa estrutura assim:

```
.services
```

5 - No index.html foi estruturado assim:

```
<div class="service-item">
```

5 - No style.css vamos usar essa estrutura assim:

```
.services .service-item
```

Na próxima página vamos definir o estilo css para a nossa primeira estrutura da SECTION (Services) que é a:

1 - No index.html foi estruturado assim:

```
<section id="services">
```

1 - No style.css vamos usar essa estrutura assim:

```
#services
```

Explicação de Cada Propriedade do Estilo CSS aplicada na SECTION Services (1 #services)

```
/* INICIO DOS SERVICES - SERVIÇOS */
```

Comentário indicando onde é o início do Código CSS na SECTION - Services (Não é renderizado pelo navegador)

`<section id="services">` Essa é a referência que veio do arquivo index.html para usarmos aqui no arquivo style.css temos que usar a # (que identifica um id) + o nome desse id(services). Portanto aqui no style.css vai ficar assim: #services

```
#services {
```

width: 100%;

Indica que a largura para esse elemento pai seja definida em 100% da largura da tela.

background: #FFB612;

indica que a cor de fundo dentro da Section #services seja #FFB612

color: white;

Indica que a cor das fontes da nossa Section #services seja branca.

padding: 40px;

Definimos um padding de 40px para (Top, Bottom, Right e Left)

```
}
```



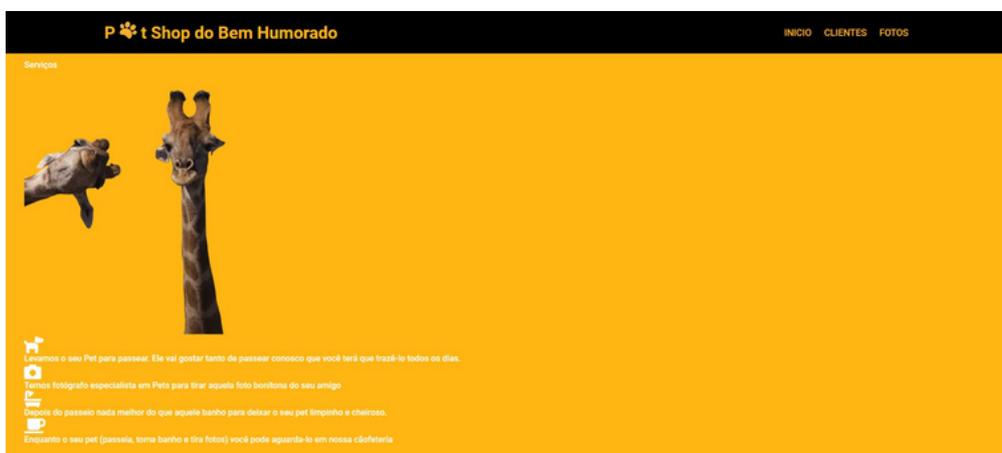
/igor-rebolla

Explicação de Cada Propriedade do Estilo CSS aplicada na SECTION Services (1 #services)

Agora veremos como o código acima ficará dentro do VSCode.

```
/* INICIO DOS SERVICES - SERVIÇOS - Imagem das 2 Girafas com 4 itens do lado direito */  
#services {  
  width: 100%;  
  background: #FFB612;  
  color: white;  
  padding: 40px;  
}
```

Agora veremos na imagem abaixo como essas 4 propriedades de estilo as quais foram aplicadas acima vão ficar quando renderizadas no navegador.



Na próxima página vamos definir o estilo css para a nossa segunda estrutura da SECTION (Services) que é a:

2 - No index.html foi estruturado assim:

```
<div class="headtext">Serviços</div>
```

2 - No style.css vamos usar essa estrutura assim:

```
.headtext
```



Explicação de Cada Propriedade do Estilo CSS aplicada na SECTION Services (2 .headtext)

`<div class="headtext">Produtos</div>` Essa é a referência que veio do arquivo index.html para usarmos aqui no arquivo style.css temos que usar o ponto (.) que identifica uma classe + o nome dessa classe(headtext). Portanto aqui no style.css vai ficar assim: .headtext

```
.headtext {
```

font-size: 2.2rem;

Definimos o tamanho da foto em 2.2rem

text-transform: uppercase;

Alteramos as letras para maiúsculas

font-weight: bold;

Negrito foi aplicado a fonte

text-align: center;

Alinhamos o texto no centro

padding-bottom: 50px;

Aplicamos um padding-bottom de 50px

letter-spacing: 2px;

Configuramos um espaçamento entre as letras de 2px.

```
}
```



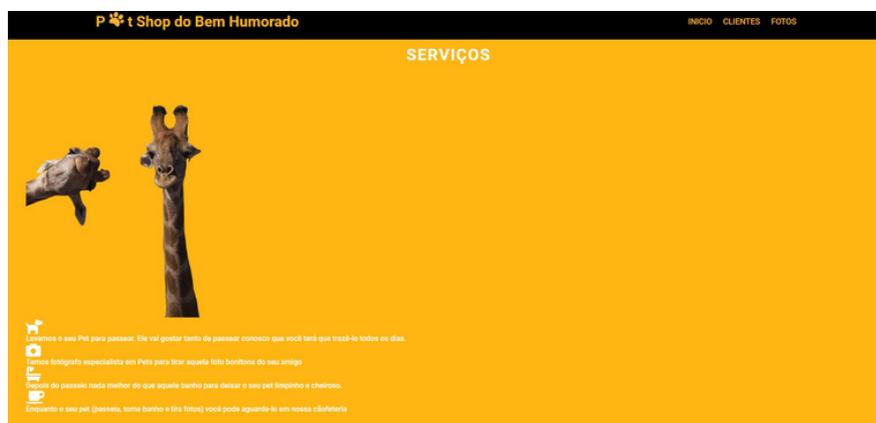
[/igor-rebolla](#)

Explicação de Cada Propriedade do Estilo CSS aplicada na SECTION Services (2 .headtext)

Agora veremos como o código acima ficará dentro do VSCode.

```
.headtext {  
  font-size: 2.2rem;  
  text-transform: uppercase;  
  font-weight: bold;  
  text-align: center;  
  padding-bottom: 50px;  
  letter-spacing: 2px;  
}
```

Agora veremos na imagem abaixo como essas 6 propriedades de estilo as quais foram aplicadas acima vão ficar quando renderizadas no navegador.



Na próxima página vamos definir o estilo css para a nossa terceira estrutura da SECTION (Services) que é a:

3 - No index.html foi estruturado assim:

```
<div class="services-container">
```

3 - No style.css vamos usar essa estrutura assim:

```
.services-container
```



Explicação de Cada Propriedade do Estilo CSS aplicada na SECTION Services (3 .services-container)

`<div class="services-container">` Essa é a referência que veio do arquivo index.html para usarmos aqui no arquivo style.css temos que usar o ponto (.) que identifica uma classe + o nome dessa classe (services-container) que é a classe que está segurando a image com as 2 girafinhas os 4 ícones (Fonte Awesome) dos nossos Serviços, além de um parágrafo. Portanto aqui no style.css vai ficar assim: .services-container

```
.services-container {
```

display: flex;

Colocamos um item ao lado do outro

justify-content: center;

Justificamos o conteúdo ao centro

align-items: center;

Alinhamos os itens ao centro

```
}
```



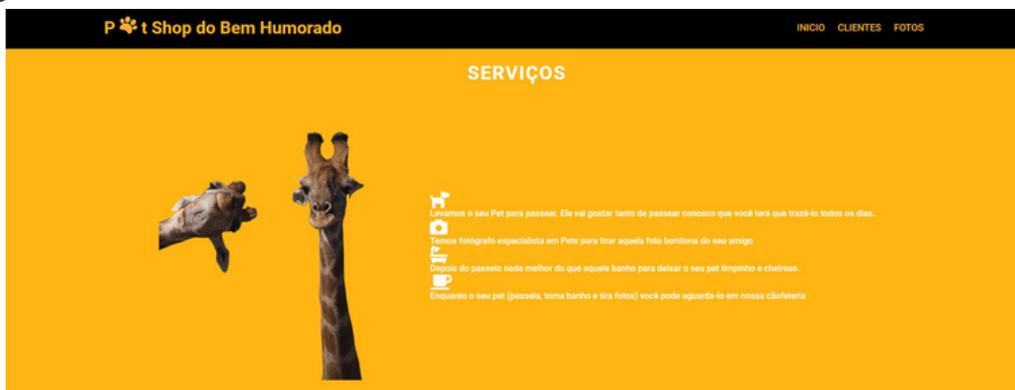
[/igor-rebolla](#)

Explicação de Cada Propriedade do Estilo CSS aplicada na SECTION Services (3 .services-container)

Agora veremos como o código acima ficará dentro do VSCode.

```
.services-container {  
  display: flex;  
  justify-content: center;  
  align-items: center;  
}
```

Agora veremos na imagem abaixo como essas 3 propriedades de estilo as quais foram aplicadas acima vão ficar quando renderizadas no navegador.



Na próxima página vamos definir o estilo css para a nossa quarta estrutura da SECTION (Services) que é a:

4 - No index.html foi estruturado assim:

```
<div class="services">
```

4 - No style.css vamos usar essa estrutura assim:

```
.services
```



Explicação de Cada Propriedade do Estilo CSS aplicada na SECTION Services (4 .services)

`<div class="services">` Essa é a referência que veio do arquivo `index.html` para usarmos aqui no arquivo `style.css` temos que usar o ponto (.) que identifica uma classe + o nome dessa classe(`services`). Portanto aqui no `style.css` vai ficar assim: `.services`

```
.services {
```

width: 625px;

Definimos uma largura de 625px

```
}
```

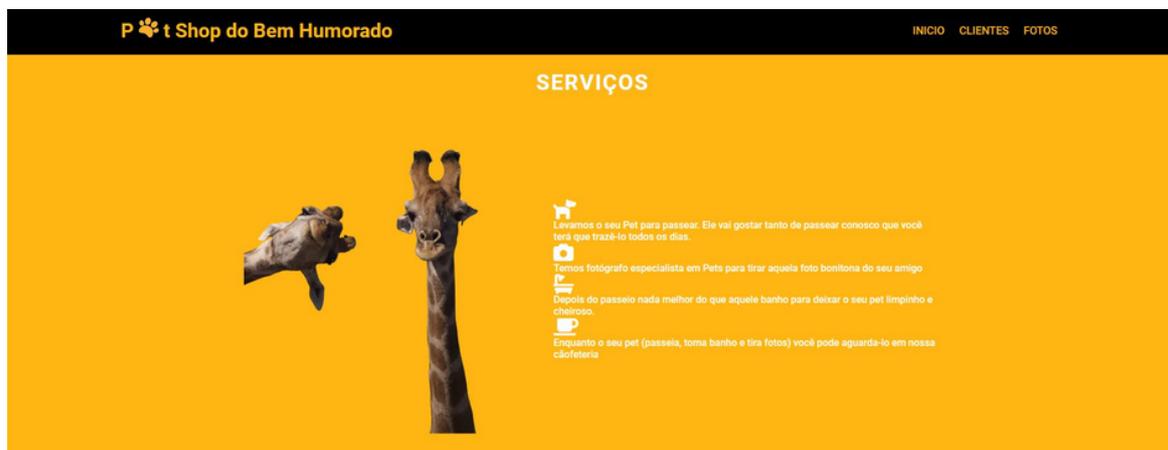


Explicação de Cada Propriedade do Estilo CSS aplicada na SECTION Services (4 .services)

Agora veremos como o código acima ficará dentro do VSCode.

```
.services {  
  width: 625px;  
}
```

Agora veremos na imagem abaixo como essa propriedade de estilo a qual foi aplicada acima vai ficar quando renderizada no navegador.



Na próxima página vamos definir o estilo css para a nossa quinta estrutura da SECTION (Services) que é a:

5 - No index.html foi estruturado assim:

```
<div class="service-item">
```

5 - No style.css vamos usar essa estrutura assim:

```
.services .service-item
```



Explicação de Cada Propriedade do Estilo CSS aplicada na SECTION Services (5 .services .service-item)

`<div class=".services .service-item ">` Essa é a referência que veio do arquivo index.html para usarmos aqui no arquivo style.css temos que usar o ponto (.) que identifica uma classe + o nome dessa classe (.services .service-item) . Portanto aqui no style.css vai ficar assim: `..services .service-item`

```
.services .service-item {
```

display: flex;

Colocamos os itens um do lado do outro

padding: 12px;

Definimos um padding de 12px para (Top, Bottom, Right e Left)

margin: 15px 0;

Definimos uma margem de 15px para (Top e Bottom) e 0 para (Right/Left)

font-size: 1.1rem;

Definimos o tamanho da fonte em 1.1rem

```
}
```

```
/* FINAL DOS SERVICES - SERVIÇOS - Imagem das 2 Girafas com 4 itens do lado direito */
```

Comentário indicando onde é o final do Código CSS na SECTION - Services (Não é renderizado pelo navegador)



Explicação de Cada Propriedade do Estilo CSS aplicada na SECTION Services (5 .services .service-item)

Agora veremos como o código acima ficará dentro do VSCode.

```
.services .service-item {  
  display: flex;  
  padding: 12px;  
  margin: 15px 0;  
  font-size: 1.1rem;  
}
```

Agora veremos na imagem abaixo como essas 4 propriedades de estilo as quais foram aplicadas acima vão ficar quando renderizadas no navegador.



E aqui chegamos no final da estilização da estrutura completa do SECTION (Services).

E agora chegou o momento de estilizarmos a SECTION (Depoimentos).

Bora estilizar a SECTION (Depoimentos)



Aplicando Estilos - SECTION

Depoimentos (#depoimentos)

3 - No index.html foi estruturado assim:

```
<div class="depoimentos-container">
```

3 - No style.css vamos usar essa estrutura assim:

```
.depoimentos-container
```

4 - No index.html foi estruturado assim:

```
<div class="depoimento">
```

4 - No style.css vamos usar essa estrutura assim:

```
.depoimento
```

5 - No index.html foi estruturado assim:

```
<div class="depoimento"> <class="depoimento-image">
```

5 - No style.css vamos usar essa estrutura assim:

```
.depoimento .depoimento-image
```

Na próxima página vamos definir o estilo css para a nossa primeira estrutura da SECTION (Depoimentos) que é a:

1 - No index.html foi estruturado assim:

```
<section id="depoimentos">
```

1 - No style.css vamos usar essa estrutura assim:

```
#depoimentos
```



Explicação de Cada Propriedade do Estilo CSS aplicada na SECTION Depoimentos (1 #depoimentos)

`/* INICIO DOS DEPOIMENTOS - 2 CARDS com 1 IMAGEM e um texto do lado direito */`

Comentário indicando onde é o início do Código CSS na SECTION - Depoimentos (Não é renderizado pelo navegador)

`<section id="depoimentos">` Essa é a referência que veio do arquivo `index.html` para usarmos aqui no arquivo `style.css` temos que usar a `#` (que identifica um id) + o nome desse id(`depoimentos`). Portanto aqui no `style.css` vai ficar assim: `#depoimentos`

```
#depoimentos {
```

width: 100%;

Indica que a largura para esse elemento pai seja definida em 100% da largura da tela.

background: #221;

indica que a cor de fundo dentro da Section `#depoimentos` seja `#221`

color: white;

Indica que a cor das fontes da nossa Section `#depoimentos` seja branca.

padding: 40px;

Definimos um padding de 40px para (Top, Bottom, Right e Left)

```
}
```



/igor-rebolla

Explicação de Cada Propriedade do Estilo CSS aplicada na SECTION Depoimentos (2 .headtext)

`<div class="headtext">Depoimentos</div>` Essa é a referência que veio do arquivo `index.html` para usarmos aqui no arquivo `style.css` temos que usar o ponto (.) que identifica uma classe + o nome dessa classe (`headtext`). Portanto aqui no `style.css` vai ficar assim: `.headtext`

```
.headtext {
```

font-size: 2.2rem;

Definimos o tamanho da foto em 2.2rem

text-transform: uppercase;

Alteramos as letras para maiúsculas

font-weight: bold;

Negrito foi aplicado a fonte

text-align: center;

Alinhamos o texto no centro

padding-bottom: 50px;

Aplicamos um `padding-bottom` de 50px

letter-spacing: 2px;

Configuramos um espaçamento entre as letras de 2px.

```
}
```



[/igor-rebolla](#)

Explicação de Cada Propriedade do Estilo CSS aplicada na SECTION Depoimentos (3 .depoimentos-container)

`<div class="depoimentos-container">` Essa é a referência que veio do arquivo index.html para usarmos aqui no arquivo style.css temos que usar o ponto (.) que identifica uma classe + o nome dessa classe(depoimentos-container). Portanto aqui no style.css vai ficar assim: `.depoimentos-container`

```
.depoimentos-container {
```

width: 80%;

Indica que a largura para esse elemento seja definida em 80% da largura da tela.

margin: auto;

Aqui a margem foi alinhada automaticamente ao meio

display: flex;

Colocamos um item ao lado do outro

flex-direction: column;

Indicamos que a direção será em colunas

align-items: center;

Alinhamos os itens ao centro

```
}
```



/igor-rebolla

Explicação de Cada Propriedade do Estilo CSS aplicada na SECTION Depoimentos (4 .depoimento)

`<div class="depoimento">` Essa é a referência que veio do arquivo `index.html` para usarmos aqui no arquivo `style.css` temos que usar o ponto (.) que identifica uma classe + o nome dessa classe(`depoimento`). Portanto aqui no `style.css` vai ficar assim:

```
.depoimento {
```

display: flex;

Colocamos os itens lado a lado

align-items: center;

Vamos alinhar os itens ao centro

font-size: 1.1rem;

O tamanho da fonte será de 1.1rem

background: #332;

indica que a cor de fundo seja #332;

padding: 15px;

Definimos um padding de 15px para (Top, Bottom, Right e Left)

margin: 15px 0;

Definimos uma margem de 15px para (Top e Bottom) e 0 para (Right/Left)

border-radius: 4px;

A borda foi levemente arredondada em 4px

```
}
```



[/igor-rebolla](#)

Explicação de Cada Propriedade do Estilo CSS aplicada na SECTION Depoimentos (4 .depoimento)

Agora veremos como o código acima ficará dentro do VSCode.

```
.depoimento {  
  display: flex;  
  align-items: center;  
  font-size: 1.1rem;  
  background: #332;  
  padding: 15px;  
  margin: 15px 0;  
  border-radius: 4px;  
}
```

Agora veremos na imagem abaixo como essas 7 propriedades de estilo as quais foram aplicadas acima vão ficar quando renderizadas no navegador.



Na próxima página vamos definir o estilo css para a nossa quinta e última estrutura da SECTION (Depoimentos) que é a:

5 - No index.html foi estruturado assim:

```
<div class="depoimento"> <class="depoimento-image">
```

5 - No style.css vamos usar essa estrutura assim:

```
.depoimento .depoimento-image
```



/igor-rebolla

Explicação de Cada Propriedade do Estilo CSS aplicada na SECTION Depoimentos (5 .depoimento .depoimento-image)

`<div class="depoimento">` `<class="depoimento-image">` Essa é a referência que veio do arquivo index.html para usarmos aqui no arquivo style.css temos que usar o ponto (.) que identifica uma classe + o nome dessa classe (depoimento .depoimento-image). Portanto aqui no style.css vai ficar assim: `.depoimento.depoimento-image`

```
.depoimento .depoimento-image {
```

```
height: 200px;
```

Definimos uma altura de 200px

```
width: auto;
```

Definimos uma largura automática

```
border-radius: 55%;
```

55% (deixa a imagem em círculo)

```
margin: 18px;
```

Definimos uma margem de 18px para (Top e Bottom) e 0 para (Right/Left)

```
}
```

```
/* FINAL DOS DEPOIMENTOS - 2 CARDS com 1 IMAGEM e um texto do lado direito */
```

Comentário indicando onde é o final do Código CSS na SECTION - Depoimentos (Não é renderizado pelo navegador)



Explicação de Cada Propriedade do Estilo CSS aplicada na SECTION Depoimentos (5 .depoimento .depoimento-image)

Agora veremos como o código acima ficará dentro do VSCode.

```
.depoimento .depoimento-image {  
  height: 200px;  
  width: auto;  
  border-radius: 55%;  
  margin: 18px;  
}
```

Agora veremos na imagem abaixo como essas 4 propriedades de estilo as quais foram aplicadas acima vão ficar quando renderizadas no navegador.



Explicação de Cada Propriedade do Estilo CSS aplicada na SECTION Depoimentos (5 .depoimento .depoimento-image)

E aqui chegamos no final da estilização da estrutura completa do SECTION (Depoimentos).

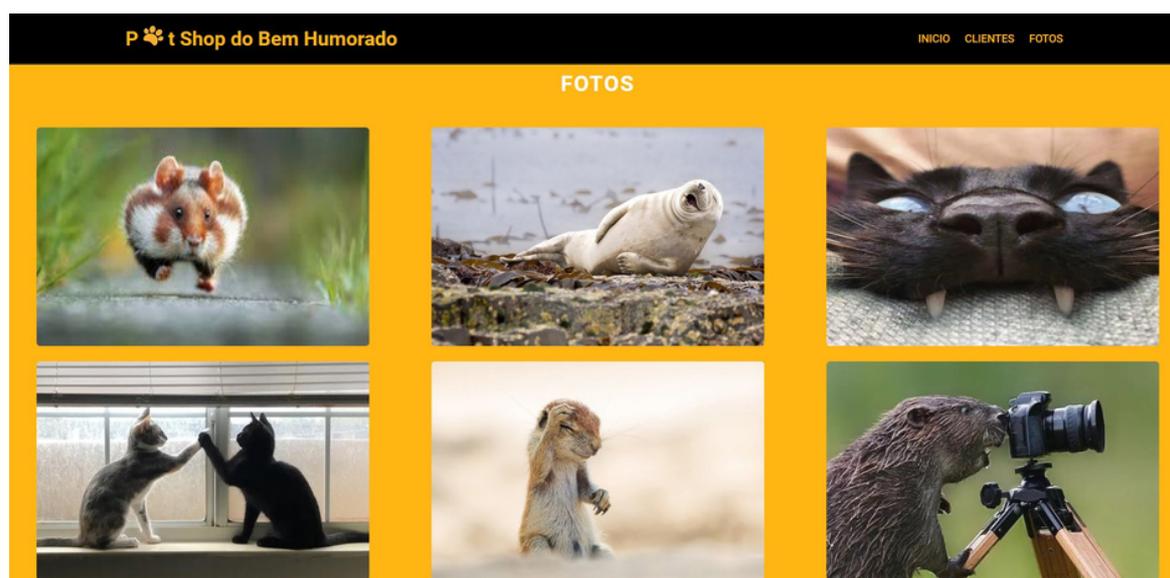
E agora chegou o momento de estilizarmos a SECTION (Photos).

Bora estilizar a SECTION (Photos)



Aplicando Estilos - SECTION Photos (#photos)

Só lembrando que o objetivo final depois de aplicarmos todos os estilos da nossa SECTION (photos) será esse abaixo:



Para chegarmos nesse resultado temos que estilizar as seguintes estruturas da SECTION (Photos) que estão vindo do arquivo index.html. E essas estruturas são:

1 - No index.html foi estruturado assim:

```
<section id="photos">
```

1 - No style.css vamos usar essa estrutura assim:

```
#photos
```

2 - No index.html foi estruturado assim:

```
<div class="headtext">Fotos</div>
```

2 - No style.css vamos usar essa estrutura assim:

```
.headtext
```

Aplicando Estilos - SECTION Photos (#photos)

3 - No index.html foi estruturado assim:

```
<div class="photos-container">
```

3 - No style.css vamos usar essa estrutura assim:

```
.photos-container
```

4 - No index.html foi estruturado assim:

```

```

4 - No style.css vamos usar essa estrutura assim:

```
.photos-image
```

Na próxima página vamos definir o estilo css para a nossa primeira estrutura da SECTION (Photos) que é a:

1 - No index.html foi estruturado assim:

```
<section id="photos">
```

1 - No style.css vamos usar essa estrutura assim:

```
#photos
```



Explicação de Cada Propriedade do Estilo CSS aplicada na SECTION Photos (1 #photos)

```
/* INICIO DOS PHOTOS - GALERIA DE FOTOS - 6 FOTOS */
```

Comentário indicando onde é o início do Código CSS na SECTION - Photos (Não é renderizado pelo navegador)

<section id="photos"> Essa é a referência que veio do arquivo index.html para usarmos aqui no arquivo style.css temos que usar a # (que identifica um id) + o nome desse id(photos). Portanto aqui no style.css vai ficar assim: #photos

```
#photos {
```

width: 90%;

Indica que a largura para esse elemento seja definida em 90% da largura da tela.

margin: auto;

Aqui a margem foi alinhada automaticamente ao meio

padding: 10px;

Definimos um padding de 10px para (Top, Bottom, Right e Left)

background: #221;

indica que a cor de fundo seja #221

color: white;

Indica que a cor das fontes seja branca.

```
}
```



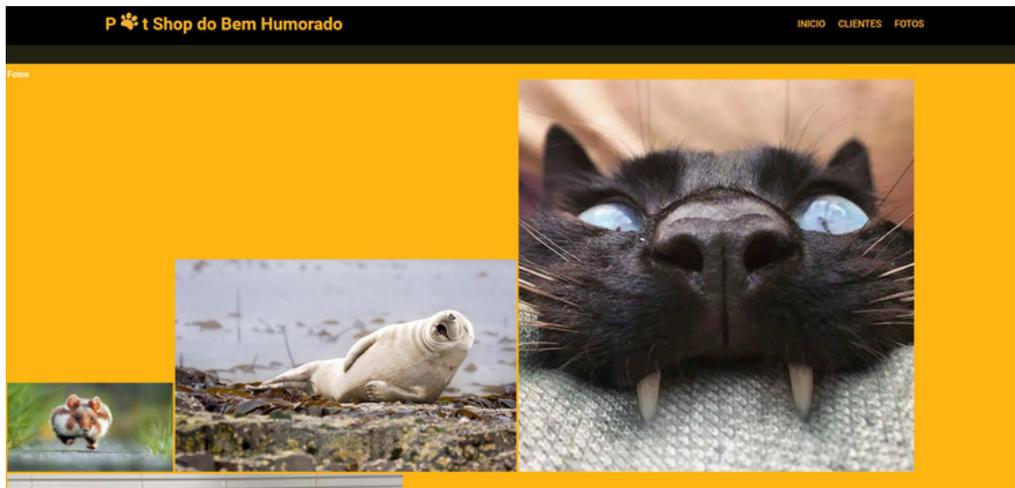
/igor-rebolla

Explicação de Cada Propriedade do Estilo CSS aplicada na SECTION Photos (1 #photos)

Agora veremos como o código acima ficará dentro do VSCode.

```
/* INICIO DOS PHOTOS - GALERIA DE FOTOS - 6 FOTOS */  
  
#photos {  
  height: 90%;  
  margin: auto;  
  padding: 10px;  
  background: #FFB612;  
  color: white;  
}
```

Agora veremos na imagem abaixo como essas 5 propriedades de estilo as quais foram aplicadas acima vão ficar quando renderizadas no navegador.



Na próxima página vamos definir o estilo css para a nossa segunda estrutura da SECTION (Photos) que é a:

2 - No index.html foi estruturado assim:

```
<div class="headtext">Fotos</div>
```

2 - No style.css vamos usar essa estrutura assim:

```
.headtext
```



/igor-rebolla

Explicação de Cada Propriedade do Estilo CSS aplicada na SECTION Photos (2 .headtext)

`<div class="headtext">Fotos</div>` Essa é a referência que veio do arquivo index.html para usarmos aqui no arquivo style.css temos que usar o ponto (.) que identifica uma classe + o nome dessa classe(headtext). Portanto aqui no style.css vai ficar assim: .headtext

```
.headtext {
```

font-size: 2.2rem;

Definimos o tamanho da foto em 2.2rem

text-transform: uppercase;

Alteramos as letras para maiúsculas

font-weight: bold;

Negrito foi aplicado a fonte

text-align: center;

Alinhamos o texto no centro

padding-bottom: 50px;

Aplicamos um padding-bottom de 50px

letter-spacing: 2px;

Configuramos um espaçamento entre as letras de 2px.

```
}
```



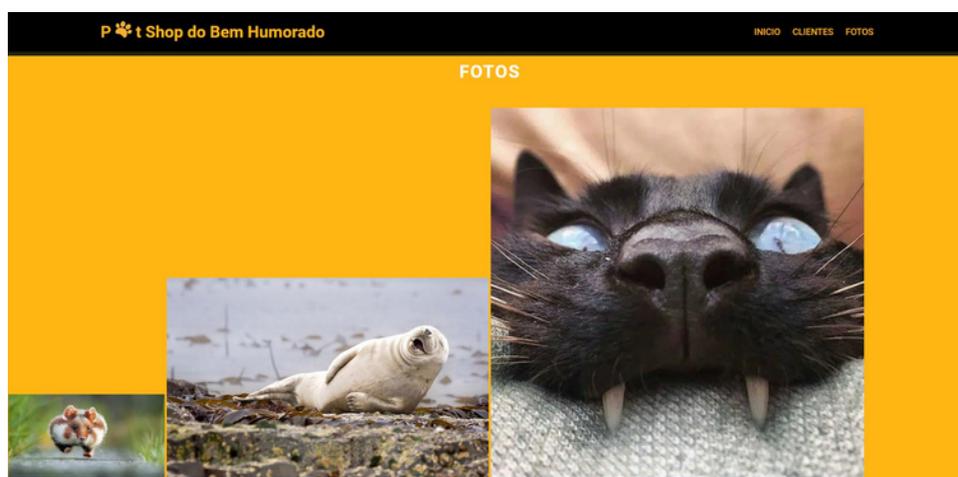
[/igor-rebolla](#)

Explicação de Cada Propriedade do Estilo CSS aplicada na SECTION Photos (2 .headtext)

Agora veremos como o código acima ficará dentro do VSCode.

```
.headtext {  
  font-size: 2.2rem;  
  text-transform: uppercase;  
  font-weight: bold;  
  text-align: center;  
  padding-bottom: 50px;  
  letter-spacing: 2px;  
}
```

Agora veremos na imagem abaixo como essas 6 propriedades de estilo as quais foram aplicadas acima vão ficar quando renderizadas no navegador.



Na próxima página vamos definir o estilo css para a nossa terceira estrutura da SECTION (Services) que é a:

3 - No index.html foi estruturado assim:

```
<div class="photos-container">
```

3 - No style.css vamos usar essa estrutura assim:

```
.photos-container
```



Explicação de Cada Propriedade do Estilo CSS aplicada na SECTION Photos (3 .photos-container)

`<div class="photos-container">` Essa é a referência que veio do arquivo `index.html` para usarmos aqui no arquivo `style.css` temos que usar o ponto (`.`) que identifica uma classe + o nome dessa classe (`photos-container`). Portanto aqui no `style.css` vai ficar assim: `.photos-container`

```
.photos-container {
```

display: grid;

Ativamos o uso do `display grid`

grid-template-columns: repeat(3, 1fr);

Queremos 3 imagens em cima e 3 imagens embaixo e que cada imagem ocupe 1 fração de espaço.

grid-gap: 25px;

Indica que criamos um espaço de 25 entre as imagens

justify-items: center;

Justificamos os itens ao centro

```
}
```



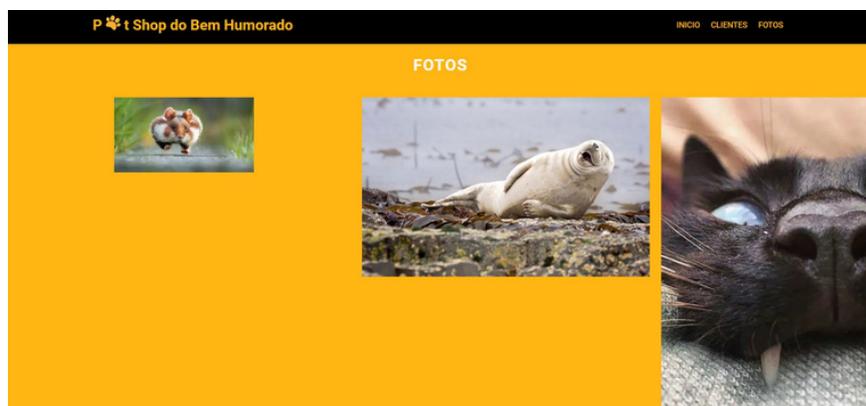
/igor-rebolla

Explicação de Cada Propriedade do Estilo CSS aplicada na SECTION Photos (3 .photos-container)

Agora veremos como o código acima ficará dentro do VSCode.

```
.photos-container {  
  display: grid;  
  grid-template-columns: repeat(3, 1fr);  
  grid-gap: 25px;  
  justify-items: center;  
}
```

Agora veremos na imagem abaixo como essas 4 propriedades de estilo as quais foram aplicadas acima vão ficar quando renderizadas no navegador.



Na próxima página vamos definir o estilo css para a nossa quarta e última estrutura da SECTION (Photos) que é a:

4 - No index.html foi estruturado assim:

```

```

4 - No style.css vamos usar essa estrutura assim:

```
.photos-image
```



/igor-rebolla

Explicação de Cada Propriedade do Estilo CSS aplicada na SECTION Photos (4 .photos-image)

```

```

Essa é a referência que veio do arquivo index.html para usarmos aqui no arquivo style.css temos que usar o ponto (.) que identifica uma classe + o nome dessa classe (photos-image). Portanto aqui no style.css vai ficar assim: .photos-image

```
.photos-image {
```

display: inline-block;

inline-block permite definir uma largura (width) e altura (height) no elemento

width: 535px;

Definimos uma largura de 535px

height: 355px;

Definimos uma altura de 355px

border-radius: 6px;

A borda de cada imagem foi levemente arredondada em 6px.

```
}
```

```
/* FINAL DOS PHOTOS - GALERIA DE FOTOS - 6 FOTOS */
```

Comentário indicando onde é o final do Código CSS na SECTION - Photos (Não é renderizado pelo navegador)

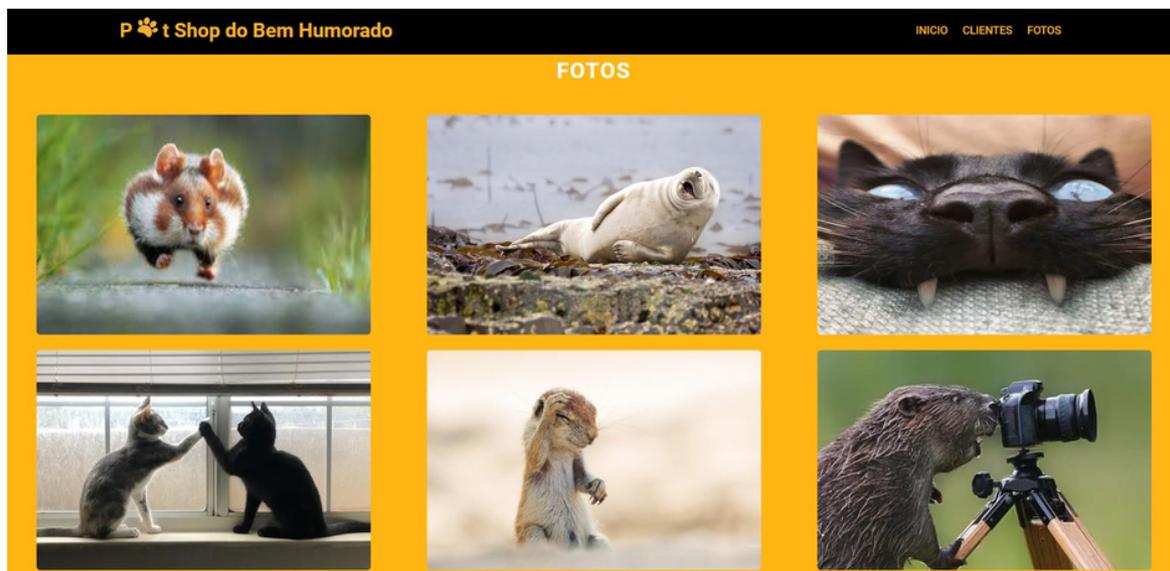


Explicação de Cada Propriedade do Estilo CSS aplicada na SECTION Photos (4 .photos-image)

Agora veremos como o código acima ficará dentro do VSCode.

```
.photos-image {  
  display: inline-block;  
  width: 535px;  
  height: 355px;  
  border-radius: 6px;  
}  
  
/* FINAL DAS PHOTOS - GALERIA DE FOTOS - 6 FOTOS */
```

Agora veremos na imagem abaixo como essas 4 propriedades de estilo as quais foram aplicadas acima vão ficar quando renderizadas no navegador.



Explicação de Cada Propriedade do Estilo CSS aplicada na SECTION Photos (4 .photos-image)

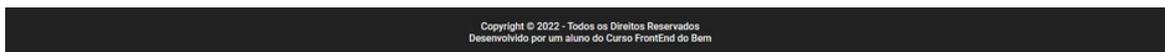
E aqui chegamos no final da estilização da estrutura completa do SECTION (Photos).

E agora chegou o momento de estilizarmos o Footer (Rodapé).

Bora estilizar o Footer (Rodapé)

Aplicando Estilos - FOOTER (#rodape)

Só relembrando que o objetivo final depois de aplicarmos todos os estilos da nossa Footer (rodape) será esse abaixo:



Para chegarmos nesse resultado temos que estilizar a seguinte estrutura do FOOTER (Rodape) que está vindo do arquivo index.html. E essa estrutura é:

1 - No index.html foi estruturado assim:

```
<footer id="rodape">
```

1 - No style.css vamos usar essa estrutura assim:

```
#rodape
```

E como essa é a única estrutura a qual foi definida no Footer (Rodape) bora estilizar na próxima página.



Explicação de Cada Propriedade do Estilo CSS aplicada no FOOTER Rodape (1 #rodape)

```
/* INICIO DO FOOTER - RODAPÉ */
```

Comentário indicando onde é o início do Código CSS do FOOTER - Rodapé (Não é renderizado pelo navegador)

`<footer id="rodape">` Essa é a referência que veio do arquivo `index.html` para usarmos aqui no arquivo `style.css` temos que usar a `#` (que identifica um id) + o nome desse id(rodape). Portanto aqui no `style.css` vai ficar assim: `#rodape`

```
#rodape {
```

background: #221;

indica que a cor de fundo seja #221

color: white;

Indica que a cor das fontes seja branca.

padding: 28px;

Definimos um padding de 28px para (Top, Bottom, Right e Left)

text-align: center;

O texto foi alinhado ao centro

```
}
```

```
/* FINAL DO FOOTER - RODAPÉ */
```

Comentário indicando onde é o final do Código CSS do FOOTER - Rodapé (Não é renderizado pelo navegador)



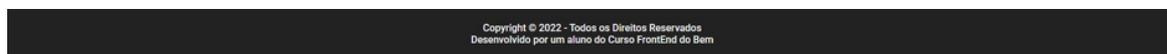
/igor-rebolla

Explicação de Cada Propriedade do Estilo CSS aplicada no FOOTER Rodape (1 #rodape)

Agora veremos como o código acima ficará dentro do VSCode.

```
/* FINAL DAS PHOTOS - GALERIA DE FOTOS - 6 FOTOS */  
  
/* INICIO DO FOOTER - RODAPÉ */  
#rodape {  
  background: #221;  
  color: white;  
  padding: 28px;  
  text-align: center;  
}  
  
/* FINAL DO FOOTER - RODAPÉ */
```

Agora veremos na imagem abaixo como essas 4 propriedades de estilo as quais foram aplicadas acima vão ficar quando renderizadas no navegador.



Aqui chegamos ao final da estilização do FOOTER e também do nosso site todo.

Só que falta mais uma "coisinha" para terminarmos a proposta desse ebook que é entregar um site responsivo.

É o que faremos já na próxima página (Vamos dentro do arquivo style.css) definir a media query que irá tornar o site responsivo.

E bora lá...



Aplicando Estilos Responsivos ao Site

Vamos primeiro definir o valor de uma media query que aqui será de 767px

```
@media screen and (max-width: 767px) {  
// AQUI DENTRO VAI TODO O NOSSO CÓDIGO PARA ESSE  
TAMANHO DE TELA O QUAL DEFINIMOS (max-width: 767px)  
}
```

Aqui nos estilos do media query eu vou utilizar uma abordagem diferente do que foi utilizada quando fizemos a estilização do site (sem a responsividade). Vou colocar todo o código responsivo já pronto e virei explicando.



[/igor-rebolla](#)

Explicação de Cada Propriedade do Estilo Responsivo do Site

```
/* INICIO DO CÓDIGO PARA TORNAR O SITE RESPONSIVO */  
Comentário indicando onde é o início do Código CSS do Site -  
Responsivo (Não é renderizado pelo navegador)
```

```
@media screen and (max-width: 767px) {
```

```
/* INICIO - APLICANDO ESTILO RESPONSIVO PARA A NAVBAR */
```

```
#navbar {
```

```
height: 235px;
```

Alteramos a altura da #navbar para 235px

```
}
```

```
.navbar-container {
```

```
flex-direction: column;
```

Vamos mudar a direção da .navbar-container para column (onde cada item vai ficar um abaixo do outro)

```
padding: 18px 0;
```

Definimos o aumento do padding de 18px para (Top e Bottom) e 0 para (Right e Left)

```
}
```



/igor-rebolla

Explicação de Cada Propriedade do Estilo Responsivo do Site

```
.navbar-items {
```

```
  flex-direction: column;
```

Vamos mudar também a direção da `.navbar-items` para `column` (onde cada item vai ficar um abaixo do outro)

```
  text-align: center;
```

Vamos alinhar o texto ao centro

```
  width: 60%;
```

Vamos atribuir uma largura de 60%

```
}
```

```
.navbar-items li:nth-child(2) {
```

Aqui eu estou definindo que a segunda `li`'s dentro do `.navbar-items` aplique esses 2 estilos abaixo

```
  border-top: 2px solid #332;
```

Vai aplicar uma linha acima de 2px (sólida) na cor #332

```
  border-bottom: 2px solid #332;
```

Vai aplicar uma linha abaixo de 2px (sólida) na cor #332

```
}
```

```
/* FINAL - APLICANDO ESTILO RESPONSIVO PARA A NAVBAR */
```



[/igor-rebolla](#)

Explicação de Cada Propriedade do Estilo Responsivo do Site

Antes de aplicarmos os estilos @media screen and (max-width: 767px) para a NAVBAR ela ficava assim quando renderizada no navegador



Depois de aplicarmos os estilos @media screen and (max-width: 767px) para a NAVBAR ela vai ficar assim quando renderizada no navegador



Explicação de Cada Propriedade do Estilo Responsivo do Site

```
/* INICIO - APLICANDO ESTILO RESPONSIVO PARA O BANNER */
```

```
.banner-container {
```

```
  padding: 35px;
```

Vamos aplicar um padding de 35px (Top, Bottom, Right e Left)

```
}
```

```
.banner-container h2 {
```

```
  padding-top: 180px;
```

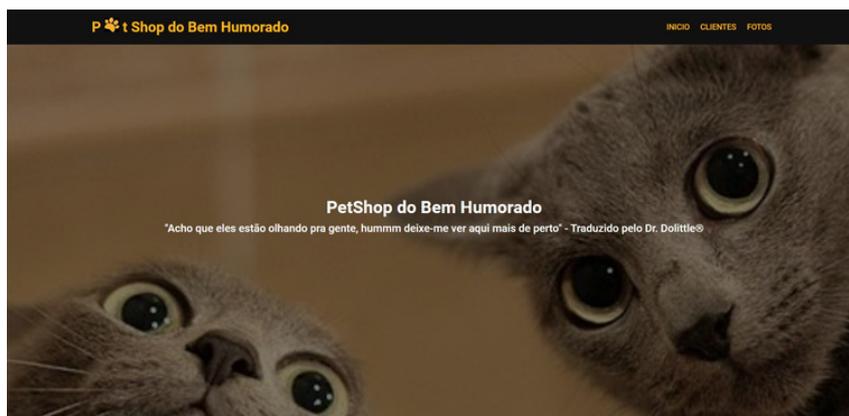
Vamos aplicar um padding no topo de 180px

```
}
```

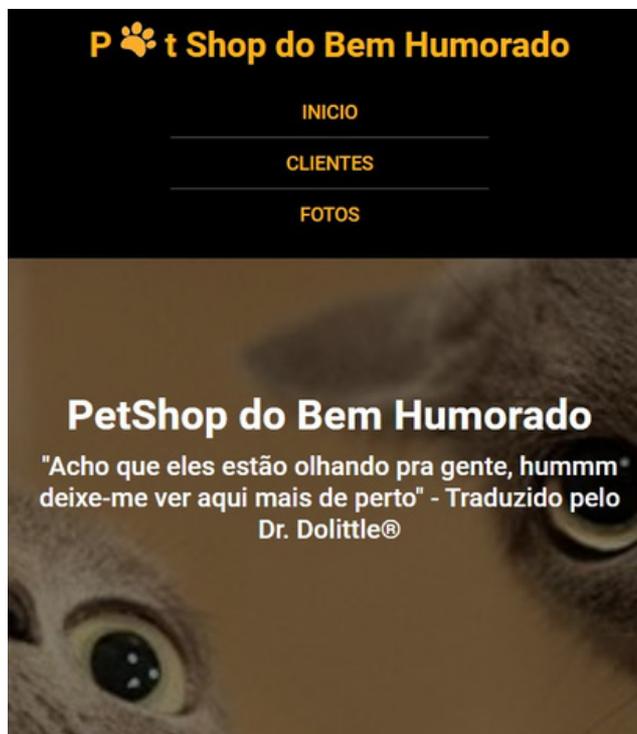
```
/* FINAL - APLICANDO ESTILO RESPONSIVO PARA O BANNER */
```

Explicação de Cada Propriedade do Estilo Responsivo do Site

Antes de aplicarmos os estilos @media screen and (max-width: 767px) para o BANNER ele ficava assim quando renderizado no navegador



Depois de aplicarmos os estilos @media screen and (max-width: 767px) para o BANNER ele vai ficar assim quando renderizado no navegador



Explicação de Cada Propriedade do Estilo Responsivo do Site

```
/* INICIO - APLICANDO ESTILO RESPONSIVO PARA OS CLIENTS */
```

```
.clients-container {
```

```
  grid-template-columns: repeat(2, 1fr);
```

Aqui queremos 2 colunas e que cada coluna ocupe 1 fração

```
}
```

```
.client {
```

```
  width: 50%;
```

Vamos alterar o width para 50%

```
}
```

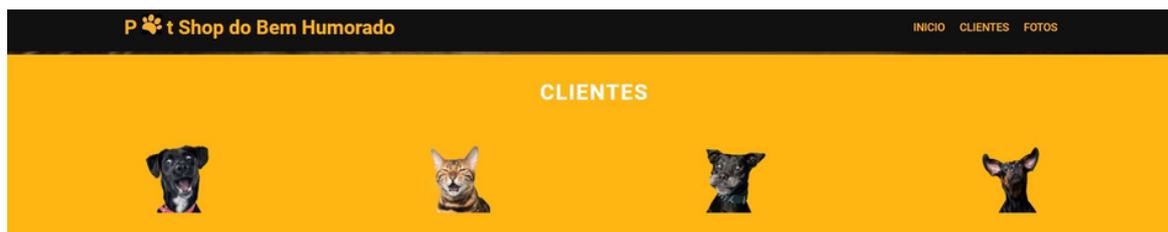
```
/* FINAL - APLICANDO ESTILO RESPONSIVO PARA OS CLIENTS */
```



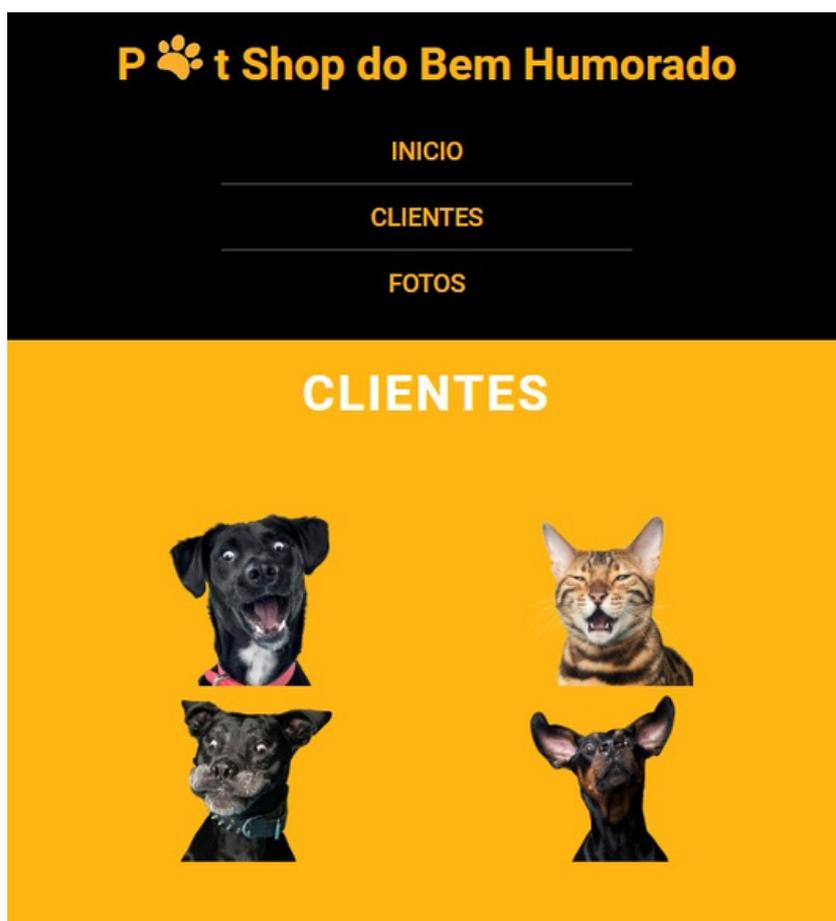
/igor-rebolla

Explicação de Cada Propriedade do Estilo Responsivo do Site

Antes de aplicarmos os estilos @media screen and (max-width: 767px) para os CLIENTES ele ficava assim quando renderizado no navegador



Depois de aplicarmos os estilos @media screen and (max-width: 767px) para os CLIENTES ele vai ficar assim quando renderizado no navegador



Explicação de Cada Propriedade do Estilo Responsivo do Site

```
/* INICIO - APLICANDO ESTILO RESPONSIVO PARA OS PRODUCTS */
```

```
.products-container {
```

```
  grid-template-columns: 1fr;
```

Definimos que as colunas ocupem 1 fração

```
}
```

```
.products-container .product:nth-child(2) {
```

Aqui eu estou definindo que o segundo product dentro de .products-container aplique esse estilo abaixo

```
  margin: 12px 0;
```

Aplicamos uma margem de 12px para (Top e Bottom) e 0 para (Right e Left)

```
}
```

```
/* FINAL - APLICANDO ESTILO RESPONSIVO PARA OS PRODUCTS */
```



/igor-rebolla

Explicação de Cada Propriedade do Estilo Responsivo do Site

Antes de aplicarmos os estilos @media screen and (max-width: 767px) para os PRODUCTS ele ficava assim quando renderizado no navegador



Depois de aplicarmos os estilos @media screen and (max-width: 767px) para os PRODUCTS ele vai ficar assim quando renderizado no navegador. Vou colocar a imagem na próxima página para uma melhor visualização.

Explicação de Cada Propriedade do Estilo Responsivo do Site



Explicação de Cada Propriedade do Estilo Responsivo do Site

```
/* INICIO - APLICANDO ESTILO RESPONSIVO PARA OS SERVICES */
```

```
.services-container {
```

```
  flex-direction: column;
```

Vamos alterar a direção para column

```
}
```

```
.services {
```

```
  width: 85%;
```

Vamos alterar a width para 85%

```
}
```

```
.service-item {
```

```
  flex-direction: column;
```

Vamos alterar a direção para column

```
  align-items: center;
```

Vamos alinhar os itens ao centro

```
  text-align: center;
```

Vamos alinhar todo o texto ao centro

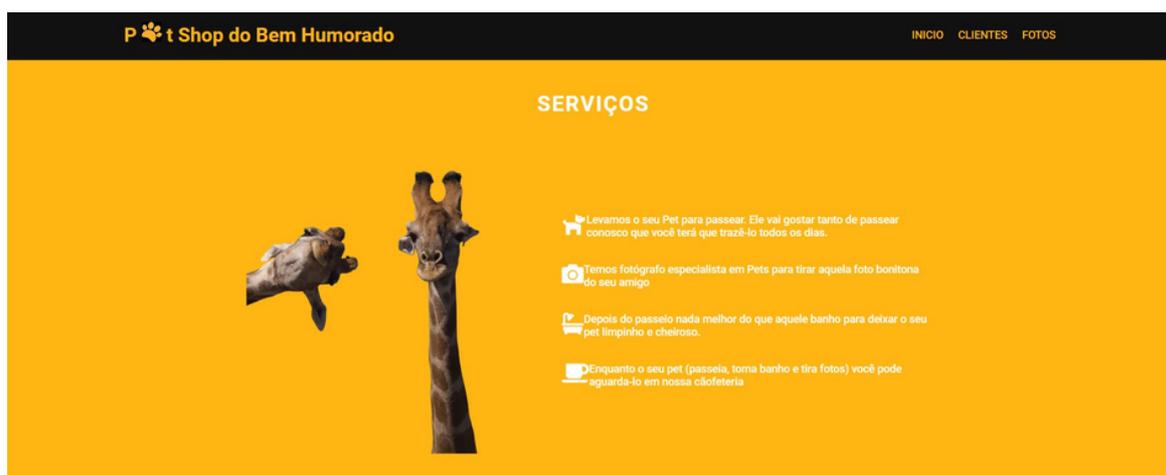
```
}
```

```
/* FINAL - APLICANDO ESTILO RESPONSIVO PARA OS SERVICES */
```



Explicação de Cada Propriedade do Estilo Responsivo do Site

Antes de aplicarmos os estilos @media screen and (max-width: 767px) para os SERVICES ele ficava assim quando renderizado no navegador



Depois de aplicarmos os estilos @media screen and (max-width: 767px) para os SERVICES ele vai ficar assim quando renderizado no navegador. Vou colocar a imagem na próxima página para uma melhor visualização.

Explicação de Cada Propriedade do Estilo Responsivo do Site



Explicação de Cada Propriedade do Estilo Responsivo do Site

```
/* INICIO - APLICANDO ESTILO RESPONSIVO PARA OS DEPOIMENTOS */
```

```
.depoimento {
```

```
  flex-direction: column;
```

Alteramos a direção para column

```
  align-items: center;
```

Alinhamos os itens ao centro

```
  text-align: center;
```

Alinhamos todo o texto ao centro

```
}
```

```
/* FINAL - APLICANDO ESTILO RESPONSIVO PARA OS DEPOIMENTOS */
```

Explicação de Cada Propriedade do Estilo Responsivo do Site

Antes de aplicarmos os estilos @media screen and (max-width: 767px) para os DEPOIMENTOS ele ficava assim quando renderizado no navegador



Depois de aplicarmos os estilos @media screen and (max-width: 767px) para os DEPOIMENTOS ele vai ficar assim quando renderizado no navegador. Vou colocar a imagem na próxima página para uma melhor visualização.

Explicação de Cada Propriedade do Estilo Responsivo do Site



/igor-rebolla

Explicação de Cada Propriedade do Estilo Responsivo do Site

```
/* INICIO - APLICANDO ESTILO RESPONSIVO PARA AS PHOTOS */
```

```
.photos-container {
```

```
grid-template-columns: 1fr;
```

Definimos que a coluna vai ocupar 1fr

```
}
```

```
.photos-image {
```

```
width: 100%;
```

Alteramos o valor do width em 100%

```
}
```

```
} /*Essa chave indica o fechamento de todas estrutura do  
@media screen and (max-width: 767px) { */
```

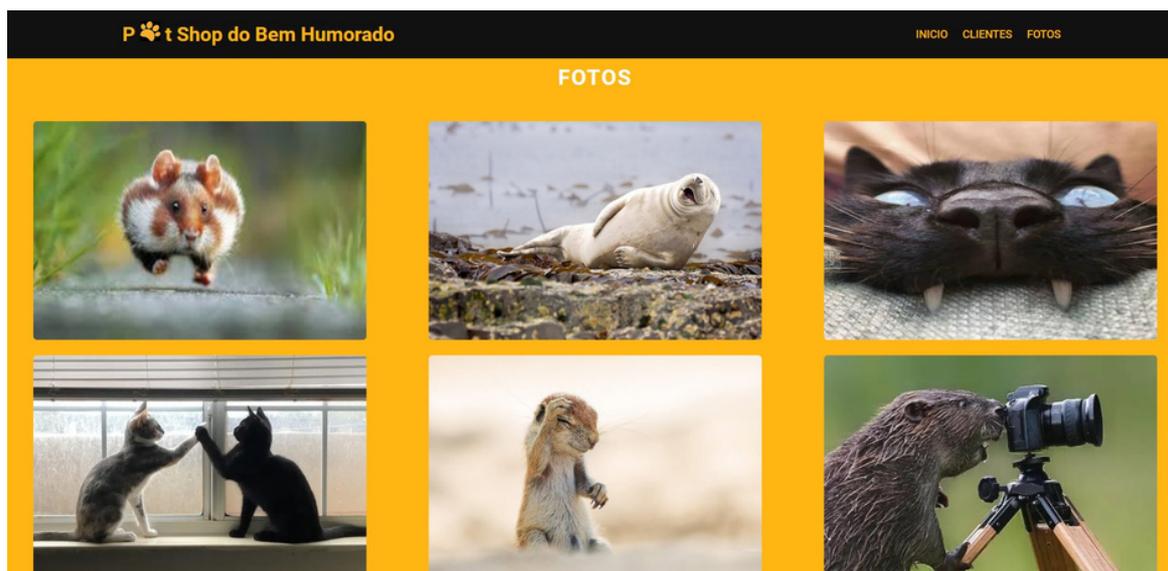
```
/* FINAL - APLICANDO ESTILO RESPONSIVO PARA AS PHOTOS */
```

```
/* FINAL DO CÓDIGO PARA TORNAR O SITE RESPONSIVO */
```



Explicação de Cada Propriedade do Estilo Responsivo do Site

Antes de aplicarmos os estilos @media screen and (max-width: 767px) para as PHOTOS ele ficava assim quando renderizado no navegador



Depois de aplicarmos os estilos @media screen and (max-width: 767px) para as PHOTOS ele vai ficar assim quando renderizado no navegador. Vou colocar a imagem na próxima página para uma melhor visualização.

Explicação de Cada Propriedade do Estilo Responsivo do Site



Como aqui são 6 fotos uma embaixo da outra (Vamos mostrar +2 na próxima página)

Explicação de Cada Propriedade do Estilo Responsivo do Site



Como aqui são 6 fotos uma embaixo da outra (Vamos mostrar as 2 últimas na próxima página)

Explicação de Cada Propriedade do Estilo Responsivo do Site



Sugestões de prática para esse e-Book:

1. Refazer todo o site (entendendo como cada item explicado funciona);
2. Refazer todo o site (sua versão - Baixando e Utilizando novas Fotos(imagens), alterando as cores, criando e ou removendo seções e etc...);
3. Colocar a sua versão do site dentro do GitHub (assim você começa a criação do seu Portfólio).

Programação é - PRÁTICA, CURIOSIDADE e REPETIÇÃO

Eu vou deixar na próxima página os links das 14 aulas do curso Front-End do Bem (caso tenham alguma dúvida e precisem consulta-las).



Links das 14 aulas do Curso Front-End do Bem

Aula 01

https://www.linkedin.com/posts/igor-rebolla_aula01-activity-6874671504194318337-HFs0

Aula 02

https://www.linkedin.com/posts/igor-rebolla_aula-02-activity-6876483117893390336-rTDt

Aula 03

https://www.linkedin.com/posts/igor-rebolla_aula03-activity-6877570766594957312-uTqo

Aula 04

https://www.linkedin.com/posts/igor-rebolla_aula04-activity-6879382544756834305-3Elo

Aula 05

https://www.linkedin.com/posts/igor-rebolla_aula05-activity-6881919110310051840-XORg

Aula 06

https://www.linkedin.com/posts/igor-rebolla_aula06-activity-6884818317207457792-3GwC

Aula 07

https://www.linkedin.com/posts/igor-rebolla_aula07-activity-6887717559639011328-MMSc



Links das 14 aulas do Curso Front-End do Bem

Aula 08

https://www.linkedin.com/posts/igor-rebolla_aula08-activity-6889891830016573440-deJx

Aula 09

https://www.linkedin.com/posts/igor-rebolla_aula09-activity-6892429004176265218-Jwj=

Aula 10

https://www.linkedin.com/posts/igor-rebolla_aula10-activity-6895328577764868096-iiby

Aula 11

https://www.linkedin.com/posts/igor-rebolla_aula11-activity-6897502324269662208-pSrE

Aula 12 -

https://www.linkedin.com/posts/igor-rebolla_aula12-activity-6900039147038617600-9dHv

Aula 13 -

https://www.linkedin.com/posts/igor-rebolla_aula13-activity-6905112194829410305-kTVI

Aula 14 -

[https://www.linkedin.com/posts/igor-rebolla_aula14-activity-6909098992022740992-rpyT?](https://www.linkedin.com/posts/igor-rebolla_aula14-activity-6909098992022740992-rpyT?utm_source=linkedin_share&utm_medium=member_desktop_web)

[utm_source=linkedin_share&utm_medium=member_desktop_web](https://www.linkedin.com/posts/igor-rebolla_aula14-activity-6909098992022740992-rpyT?utm_source=linkedin_share&utm_medium=member_desktop_web)



Créditos das Fotos Utilizadas no Desenvolvimento do Site

banner

<https://portaldoscaese gatos.com.br/fotos-de-gatos-engracados/ver-videos-engracados-de-gatos-animais/>

girafas

<https://casavogue.globo.com/Curiosidades/noticia/2020/09/estas-sao-fotos-de-animais-mais-engracadas-de-2020.html>

depoimento-1

<https://br.pinterest.com/pin/797840890207423885/>

depoimento-2

<https://profanos.com/animais-fazendo-caretas-engracadas/>

client-1

<https://br.jetss.com/virais/bombando/2018/06/fotografo-captura-cachorros-fazendo-caretas-e-o-resultado-e-hilario/>

client-2

<https://br.jetss.com/virais/bombando/2018/06/fotografo-captura-cachorros-fazendo-caretas-e-o-resultado-e-hilario/>

client-3

<https://br.jetss.com/virais/bombando/2018/06/fotografo-captura-cachorros-fazendo-caretas-e-o-resultado-e-hilario/>



/igor-rebolla

Créditos das Fotos Utilizadas no Desenvolvimento do Site

client-4

<https://br.jetss.com/virais/bombando/2018/06/fotografo-captura-cachorros-fazendo-caretas-e-o-resultado-e-hilario/>

image-1

<https://www.hypeness.com.br/2016/03/concurso-premia-as-fotos-mais-engracadas-do-reino-animal-veja-as-imagens/>

image-2

<https://www.hypeness.com.br/2016/03/concurso-premia-as-fotos-mais-engracadas-do-reino-animal-veja-as-imagens/>

image-3

<https://segredosdomundo.r7.com/fotos-de-animais/>

image-4

<https://segredosdomundo.r7.com/fotos-de-animais/>

image-5

<https://segredosdomundo.r7.com/fotos-de-animais/>

image-6

<https://segredosdomundo.r7.com/fotos-de-animais/>



/igor-rebolla