

RESPONSIVE WEB DEVELOPMENT

CSS

ANDREY MASIERO



3

LISTA DE FIGURAS

Figura 3.1 – Declarações de CSS em um arquivo HTML.....	5
Figura 3.2 – Precedência da aplicação do CSS	6
Figura 3.2 – Sintaxe da regra do CSS.....	8
Figura 3.3 – Página inicial sem formatação.....	11
Figura 3.4 – Fonte formatada para o título	12
Figura 3.5 – Medidas de espaçamento entre elementos HTML	13
Figura 3.6 – Direções para a aplicação de medidas no box model	13
Figura 3.7 – Margem à esquerda aplicada ao título	15
Figura 3.8 - Resultado da formatação do banner	15
Figura 3.9 – Adição da propriedade object-fit.....	17
Figura 3.10 – Reposicionando a imagem	17
Figura 3.11 – Formatação do conteúdo principal	20
Figura 3.12 – Formatação da área de serviços	22
Figura 3.13 - Espaço em branco entre as seções do site.....	23
Figura 3.14 - Resultado da formatação após a aplicação do arquivo reset.css	24
Figura 3.15 – Aplicação do efeito hover no título	26
Figura 3.16 – Resultado do cabeçalho.....	29
Figura 3.17 – Reformatação do conteúdo principal após a aplicação do reset.css...	30
Figura 3.18 – Resultado da formatação da seção de serviços.....	31
Figura 3.19 – Resultado do rodapé	32
Figura 3.20 – Primeira linha da galeria de foto formatada.....	35
Figura 3.21 – Galeria de fotos finalizada.....	37
Figura 3.22 – Resultado do formulário formatado	40

LISTA DE CÓDIGOS-FONTE

Código-fonte 3.1 – Aplicação inline do CSS	6
Código-fonte 3.2 – CSS interno ao HTML	7
Código-fonte 3.3 – Link para o CSS externo	7
Código-fonte 3.4 – Código HTML da homepage	10
Código-fonte 3.5 – Formatação do elemento h1 por meio do CSS	11
Código-fonte 3.6 – Classe titulo-principal	12
Código-fonte 3.7 – Aplicação da classe titulo-principal no elemento h1 dentro do HTML	12
Código-fonte 3.8 – Aplicação de margin em um seletor de classe	14
Código-fonte 3.9 – Exemplos de configurações do box model	14
Código-fonte 3.10 – Ajuste de margin na classe titulo-principal	15
Código-fonte 3.11 – Ajuste do banner por meio do seletor de id	15
Código-fonte 3.12 – Configuração da imagem dentro do banner	16
Código-fonte 3.13 – Ajuste no posicionamento da imagem	17
Código-fonte 3.14 – Adição das classes nos elementos HTML do conteúdo principal	18
Código-fonte 3.15 – Determinação das configurações da classe principal	18
Código-fonte 3.16 – Configuração da classe titulo-centralizado	19
Código-fonte 3.17 – Configuração para o elemento p	19
Código-fonte 3.18 – Configuração para o elemento com id missão	19
Código-fonte 3.19 – Configuração para o elemento strong, filho de em	20
Código-fonte 3.20 – Adição das classes na sessão de serviços	21
Código-fonte 3.21 – Configuração da classe serviços	21
Código-fonte 3.22 – Configuração do elemento ul	22
Código-fonte 3.23 – Configuração da classe imagem-serviços	22
Código-fonte 3.24 – Inclusão do arquivo reset.css para limpar as formatações padrões do navegador	23
Código-fonte 3.25 – Inclusão do CDN para o fontawesome, de modo a trabalhar com ícones	24
Código-fonte 3.26 – Inclusão de um ícone no cabeçalho	25
Código-fonte 3.27 – Configuração das classes que envolvem o cabeçalho	25
Código-fonte 3.28 – Inclusão de um elemento nav para criação do menu	27
Código-fonte 3.29 – Trabalhando no posicionamento do nav a partir de seu elemento pai	27
Código-fonte 3.30 – Configuração CSS para o menu	28
Código-fonte 3.31 – Configuração da seção de conteúdo principal	30
Código-fonte 3.32 – Configuração da lista na área de serviços	30
Código-fonte 3.33 – Criação do footer	31
Código-fonte 3.34 – Classes para configurar o rodapé	32
Código-fonte 3.35 – Código HTML da página da galeria	33
Código-fonte 3.36 – Seção para criar a linha de imagens da galeria	34
Código-fonte 3.37 – Configuração das classes para a galeria	34
Código-fonte 3.38 – Replicação da linha para criar a galeria de fotos	36
Código-fonte 3.39 – Código HTML para o formulário da página de contato	38
Código-fonte 3.40 – Código CSS para o formulário da página de contato	39

SUMÁRIO

3 CSS	5
3.1 Inline	6
3.1.1 Interna	7
3.1.2 Externa	7
3.1.3 Efeito cascata	8
3.2 Criando uma Regra CSS	8
3.3 Partindo do HTML	9
3.4 Formatando o cabeçalho	11
3.5 Chegou a hora do conteúdo principal	17
3.6 Quais são os serviços oferecidos?	20
3.7 Limpando as formatações	23
3.8 De volta ao cabeçalho: ícones e menu	24
3.9 Flexibilizando a galeria	32
3.10 Entrando em contato	37
REFERÊNCIAS	41

3 CSS

- O Cascading Style Sheets (CSS) é uma linguagem de estilos para trazer melhores recursos visuais a um documento HTML. Foi criada pelo W3C para definir cores, tipologia, posicionamento e mais recursos em páginas web.
- Ele possibilita que determinadas propriedades sejam aplicadas, ao mesmo tempo, a diversos elementos de uma página ou site que estejam marcados com uma tag ou atributo específico. Assim, facilitam a criação, formatação e manutenção de layouts e estilos de páginas web.
- Criado no fim de 1996, é um padrão que define um dos principais conceitos relacionados aos Web Standards. Assim, como o HTML estrutura um documento por meio de blocos de informação, o CSS controla toda a aparência e o layout do documento. É uma solução eficiente para administração de sites com alto volume de páginas.
- Existem três maneiras de aplicar, ou utilizar, um código CSS em um documento HTML. Veja, na Figura “Declarações de CSS em um arquivo HTML”, as três possíveis aplicações de CSS em um documento.



Figura 3.1 – Declarações de CSS em um arquivo HTML
Fonte: Elaborado pelo autor (2020)

É possível utilizar os três tipos de declaração em um único documento HTML. Porém, antes de aplicá-los, é importante entender a regra de prioridade ao atribuir um

estilo, utilizando os três tipos de declaração. Existe a precedência de aplicação, portanto, é importante entender a seguinte regra:

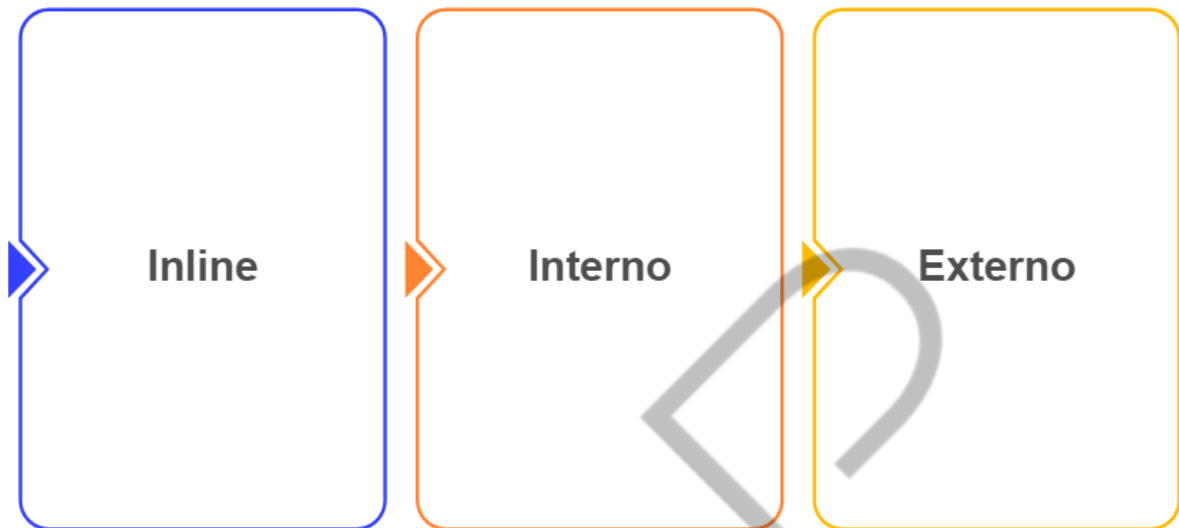


Figura 3.2 – Precedência da aplicação do CSS
Fonte: Elaborado pelo autor (2020)

Vamos abordar em detalhes cada uma das possíveis declarações.

3.1 Inline

Com esse tipo de declaração, você mistura a formatação de um elemento com o seu respectivo conteúdo. Isso significa que, caso você queira que a mesma regra seja usada por outro elemento, terá de refazer a mesma declaração no elemento desejado. Sendo assim, esse tipo de declaração torna a manutenção mais complicada do que nos demais tipos.

Para fazer uma declaração *inline*, utilizamos o atributo da tag HTML `style`. Veja um exemplo:

```
<p style="background-color: #900;">  
Luke, eu sou seu pai!  
</p>
```

Código-fonte 3.1 – Aplicação inline do CSS
Fonte: Elaborado pelo autor (2020)

3.1.1 Interna

Nessa declaração, utiliza-se a tag `style` e o conteúdo dessa tag é o código CSS. Em geral, esse tipo de aplicação é inserido na tag `head` do documento HTML. Assim, conseguimos reaproveitar todo o conteúdo de formatação no documento inteiro. Porém, para reaproveitar essa aplicação, temos que efetuar o famoso Copiar e Colar entre as diversas páginas no sistema. Assim fica complicado realizar a manutenção e manter um padrão de layout e estilos em um sistema inteiro.

Veja o exemplo de uma declaração interna:

```
<style type="text/css">
  p{
    background-color: #900;
  }
</style>
```

Código-fonte 3.2 – CSS interno ao HTML
Fonte: Elaborado pelo autor (2020)

Nesse exemplo, todos os elementos `p` têm a aplicação do plano de fundo na cor vermelha.

3.1.2 Externa

Nesse tipo de declaração, é criado um arquivo externo contendo todo o código CSS. Essa aplicação é inserida na seção `head` do documento HTML e faz menção ao arquivo que encontra o código aplicado ao sistema. Uma mesma regra CSS dentro desse arquivo pode formatar simultaneamente quantas páginas o desenvolvedor precisar. Dessa forma, a manutenção é mais fácil e escalável em sistemas web. Utilize sempre esse padrão para o desenvolvimento do sistema.

A tag que faz o vínculo das regras CSS ao documento é a seguinte:

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

Código-fonte 3.3 – Link para o CSS externo
Fonte: Elaborado pelo autor (2020)

3.1.3 Efeito cascata

O CSS recebe esse nome devido ao efeito cascata encontrado em sua aplicação no documento. Isso significa que, ao aplicar mais de um estilo em um elemento, poderá ocorrer uma sobreposição de estilos. Confira a sequência da cascata, do estilo mais geral ao mais específico:

1. Estilo padrão do navegador;
2. Folha de Estilo Externa (referenciada e/ou importada);
3. Folha de Estilo Interna (definida na área de cabeçalho do documento);
4. Folha de Estilo Inline (definida dentro de um elemento HTML).

3.2 Criando uma Regra CSS

A regra CSS é uma declaração com sintaxe própria que indica como será o estilo de um ou mais elementos HTML. Um conjunto de regras CSS forma o que chamamos de folha de estilos.

A regra é composta por um seletor, propriedades e seus respectivos valores. Veja a Figura “Sintaxe da regra do CSS”, apresentando os detalhes da sintaxe do CSS.



Figura 3.3 – Sintaxe da regra do CSS
Fonte: Elaborado pelo autor (2020)

Seletores podem ser elementos HTML como p, h1, h2 etc. Podem ser, também, identificadores que combinamos com o símbolo # e classes que combinamos com o símbolo ponto (.), ambos aplicados antes do nome.

Agora que você já sabe sobre o que trata o CSS, vamos explorar esse conteúdo todo na prática. Ao longo da apostila, vamos criar um projeto de um site de restaurante. Passo a passo, nos aprofundaremos em cada elemento HTML.

3.3 Partindo do HTML

- A partir do HTML vamos inserir um arquivo externo para a formatação do layout de nosso projeto. Iniciaremos com a página principal e aprofundaremos na formatação à medida que avançarmos no projeto. O código inicial do HTML é o seguinte.

```
<!DOCTYPE html>
<html lang="pt-br" dir="ltr">
  <head>
    <meta charset="utf-8" />
    <title>Restaurante FIAP</title>
    <link rel="stylesheet" href="style.css" type="text/css" />
  </head>
  <body>
    <header>
      <h1>
        Restaurante FIAP
      </h1>
    </header>
    
    <div>
      <h2>
        Sobre o Restaurante FIAP
      </h2>
      <p>
        Localizado no bairro da Aclimação, o <strong>Restaurante FIAP</strong> traz um novo conceito de restaurante para você vivenciar. Fundado em 2019, o Restaurante FIAP entrega um misto de tecnologia com alta gastronomia para gerar uma disruption com a humanização de chefs robôs.
      </p>
      <p id="missao">
```

```
<em>Nossa missão é: <strong>"Proporcionar a união
da tecnologia e gastronomia para uma sociedade mais inclusiva
."</strong>.</em>
</p>
<p>
    O atendimento é personalizado para cada situação
de sua vida. Cada prato é servido e preparado exclusivamente p
ara cada mesa de acordo com a sazonalidade dos ingredientes.
</p>
</div>
<div>
    <h3>
        Serviços
    </h3>
    <ul>
        <li>Carta de vinhos orgânicos</li>
        <li>Pratos personalizados por mesa</li>
        <li>Alta tecnologia em todos os ambientes</li>
        <li>Espaço família em ambiente reservado</li>
    </ul>
    
</div>
</body>
</html>
```

Código-fonte 3.4 – Código HTML da homepage
Fonte: Elaborado pelo autor (2020)

- O resultado é uma página sem formatação, como a apresentada na Figura “Página inicial sem formatação”.

Restaurante FIAP



Sobre o Restaurante FIAP

Localizado no bairro da Aclimação o **Restaurante FIAP** traz um novo conceito de restaurante para você vivenciar. Fundado em 2019, o Restaurante FIAP entrega um misto de tecnologia com alta gastronomia para gerar uma disrupção com a humanização de chefs robôs.

Nossa missão é: *"Proporcionar a união da tecnologia e gastronomia para uma sociedade mais inclusiva."*

O atendimento é personalizado para cada situação de sua vida. Cada prato é servido e preparado exclusivamente para cada mesa de acordo com a sazonalidade dos ingredientes.

Serviços

- Carta de vinhos orgânicos
- Pratos personalizados por mesa
- Alta tecnologia em todos os ambientes
- Espaço família em ambiente reservadosazonalidade



Figura 3.4 – Página inicial sem formatação
Fonte: Elaborado pelo autor (2020)

3.4 Formatando o cabeçalho

Vamos começar trabalhando como o cabeçalho. O primeiro passo é formatar a fonte do cabeçalho. Aplicaremos a fonte Arial, tamanho 20, com negrito no elemento h1. Veja como fica o código.

```
h1 {  
    font-family: Arial, Helvetica, sans-serif;  
    font-size: 20;  
    font-weight: bold;  
}
```

Código-fonte 3.5 – Formatação do elemento h1 por meio do CSS
Fonte: Elaborado pelo autor (2020)

Na propriedade font-family, definimos uma lista de fontes. Se o sistema não encontrar a fonte Arial, ele deve procurar pela Helvética e, caso não a encontre

também, deve utilizar qualquer fonte da categoria sans-serif. A font-size define o tamanho da fonte, como no Word, ou qualquer outro editor de texto. E font-weight é a intensidade do peso dessa fonte, ou seja, quão negrito você vai deixar a fonte.

Aplicando diretamente no elemento h1, temos a vantagem de determinar a formatação para todos os elementos h1. Porém, nesse caso, h1 significa o título principal do site e isso pode gerar um conflito com outras partes do código. Uma boa prática é, sempre que possível, utilizar o recurso de classes no CSS. Sendo assim, vamos definir a classe chamada titulo-principal para aplicar nesse elemento em específico.

```
.titulo-principal {  
    font-family: Arial, Helvetica, sans-serif;  
    font-size: 20;  
    font-weight: bold;  
}
```

Código-fonte 3.6 – Classe titulo-principal
Fonte: Elaborado pelo autor (2020)

O elemento h1 no HTML deve definir a propriedade class, vinculando, assim, a formatação. Veja como fica.

```
<h1 class="titulo-principal">  
    Restaurante FIAP  
</h1>
```

Código-fonte 3.7 – Aplicação da classe titulo-principal no elemento h1 dentro do HTML
Fonte: Elaborado pelo autor (2020)

Confira o resultado por meio da Figura “Fonte formatada para o título”.



Figura 3.5 – Fonte formatada para o título
Fonte: Elaborado pelo autor (2020)

Apesar de a fonte estar em um padrão bacana agora, ela está muito próxima da borda esquerda do site. Para trabalhar com esse espaçamento, precisamos entender como os elementos HTML funcionam em relação ao layout. Diferentemente do formato apresentado na tela, todo elemento HTML possui um formato de bloco.

Cada bloco possui três medidas em sua volta: padding, border e margin. Veja a Figura “Medidas de espaçamento entre elementos HTML”.

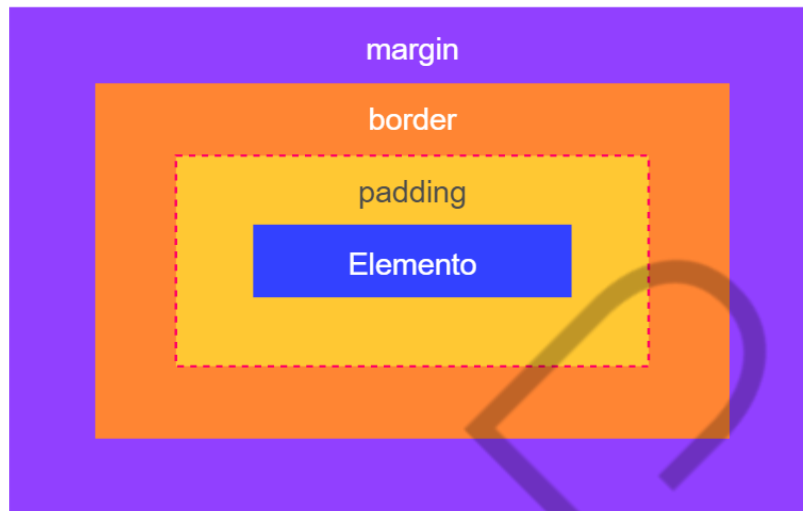


Figura 3.6 – Medidas de espaçamento entre elementos HTML
Fonte: Elaborado pelo autor (2020)

Utilizamos cada uma dessas medidas para dar um respiro ao redor do elemento desejado. Essas medidas fazem parte do que conhecemos como box model no mundo HTML/CSS. Podemos atribuir as métricas de maneira individual ou de forma abreviada. A Figura “Direções para aplicação de medidas no box model” apresenta as direções das medidas do box model.

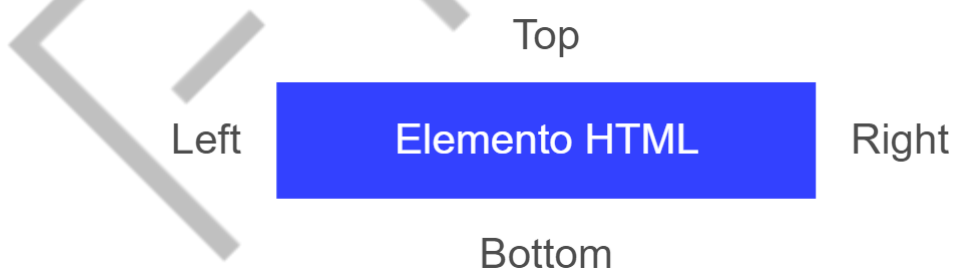


Figura 3.7 – Direções para a aplicação de medidas no box model
Fonte: Elaborado pelo autor (2020)

Vamos ver, a seguir, como definir o espaçamento de cada uma das direções:

```
.seletor {  
  /*  
    Escolha a medida para aplicar o espaçamento:  
    - margin  
    - padding  
    - border
```

```
        No exemplo escolhemos a margin, mas o comando
        para as demais é o mesmo.
    */
    margin-top: 10px;
    margin-right: 15px;
    margin-bottom: 20px;
    margin-left: 25px;
}
```

Código-fonte 3.8 – Aplicação de margin em um seletor de classe
Fonte: Elaborado pelo autor (2020)

Além disso, temos a forma abreviada, que também utiliza o mesmo princípio de ser aplicável aos três elementos do box model. Dessa vez, selecionaremos para exemplo o padding. Na forma abreviada, podemos informar uma, duas ou quatro medidas. Ao informar uma medida, essa será aplicada em todas as direções. Duas medidas são replicadas nas direções vertical e horizontal, respectivamente. Já, se forem informadas as quatro medidas, teremos a aplicação nas direções de maneira individual, começando do topo e seguindo no sentido horário. Vamos ver um exemplo:

```
.seletorUnico {
    /*todas direções */
    padding: 10px;
}

.seletorDuplo {
    /*vertical | horizontal */
    padding: 20px 10px;
}

.seletorQuadruplo {
    /* top | right | bottom | left */
    padding: 5px 10px 15px 20px;
}
```

Código-fonte 3.9 – Exemplos de configurações do box model
Fonte: Elaborado pelo autor (2020)

Agora que entendemos o box model, vamos ajustar a margem à esquerda de nosso título principal. Aplicaremos 20px para dar o respiro necessário.

```
.titulo-principal {
    font-family: Arial, Helvetica, sans-serif;
    font-size: 20;
    font-weight: bold;
    margin-left: 20px;
}
```

```
}
```

Código-fonte 3.10 – Ajuste de margin na classe titulo-principal
Fonte: Elaborado pelo autor (2020)

Confira o resultado na Figura “Margem à esquerda aplicada ao título”.

Restaurante FIAP



Figura 3.8 – Margem à esquerda aplicada ao título
Fonte: Elaborado pelo autor (2020)

O próximo passo é trabalhar com a imagem logo abaixo. Vamos deixá-la ocupando a tela toda e uma altura máxima de 550px. Vamos utilizar o seletor de um identificador para a imagem que representa o banner do nosso site.

```
#banner {  
  height: 550px;  
  width: 100%;  
}
```

Código-fonte 3.11 – Ajuste do banner por meio do seletor de id
Fonte: Elaborado pelo autor (2020)

Vamos conferir o resultado na Figura “Resultado da formatação do banner”.

Restaurante FIAP



Sobre o Restaurante FIAP

Trabalhando no bairro de Arslançık, o Restaurante FIAP tem um menu variado de restaurantes para cada ocasião. Fundado em 2010, o Restaurante FIAP oferece um menu de variedade com alta qualidade e mais uma diversidade com a harmonização da culinária sobre

Figura 3.9 - Resultado da formatação do banner
Fonte: Elaborado pelo autor (2020)

Perceba que a imagem está um pouco distorcida. Para solucionarmos isso, vamos trabalhar com a propriedade chamada object-fit. Essa propriedade define como o elemento responderá à altura e à largura do seu box. Cinco valores podem ser dados para essa propriedade:

- fill: valor padrão, vai preencher o box com todo o conteúdo sem se preocupar com a distorção aplicada.
- contain: expandirá ou reduzirá o tamanho da imagem para ocupar o tamanho do box, se preocupando com a proporção correta do elemento.
- cover: preencherá a altura e a largura do elemento, mantendo a sua proporção, porém realizará um corte na imagem para se encaixar de maneira adequada.
- none: ignora as medidas e mantém a imagem no tamanho original.
- scale-down: compara a imagem entre o none e contain e mantém a de menor escala para o box.

O que mais nos agrada para o nosso projeto é a opção de cover, vamos aplicá-la.

```
#banner {  
  height: 550px;  
  object-fit: cover;  
  width: 100%;  
}
```

Código-fonte 3.12 – Configuração da imagem dentro do banner
Fonte: Elaborado pelo autor (2020)

Confira o resultado da aplicação na Figura “Adição da propriedade object-fit”.

Restaurante FIAP



Sobre o Restaurante FIAP

É proibido a utilização de imagens de terceiros sem autorização expressa do autor. Todas as imagens são de domínio público e foram utilizadas apenas para fins ilustrativos.

Figura 3.10 – Adição da propriedade object-fit
Fonte: Elaborado pelo autor (2020)

Melhorou bastante o visual, mas podemos fazer um pequeno ajuste no posicionamento da imagem para ficar mais agradável. Nessa operação, utilizaremos a propriedade object-position, que permite manipular a imagem dentro do box em x e y.

```
#banner {  
  height: 550px;  
  object-fit: cover;  
  object-position: 0% 65%;  
  width: 100%;  
}
```

Código-fonte 3.13 – Ajuste no posicionamento da imagem
Fonte: Elaborado pelo autor (2020)

Pronto, agora vamos ver o resultado na Figura “Reposicionando a imagem”.

Restaurante FIAP



Sobre o Restaurante FIAP

Figura 3.11 – Reposicionando a imagem
Fonte: Elaborado pelo autor (2020)

Neste momento, atingimos o resultado desejado. Agora, vamos passar para a segunda parte do conteúdo, a área de “Sobre o Restaurante FIAP”.

3.5 Chegou a hora do conteúdo principal

Vamos atribuir algumas classes ao conteúdo HTML para facilitar nossa formatação.

```
<section class="conteudo-principal">  
  <h2 class="titulo-centralizado">
```

```

        Sobre o Restaurante FIAP
    </h2>
    <p>
        Localizado no bairro da Aclimação, o
        <strong>Restaurante FIAP</strong> traz um novo co
nheiro de restaurante
        para você vivenciar. Fundado em 2019, o Restauran
te FIAP entrega um
        misto de tecnologia com alta gastronomia para ger
ar uma disrupção com a
        humanização de chefs robôs.
    </p>
    <p id="missao">
        <em>
            Nossa missão é:
            <strong>
                "Proporcionar a união da tecnologia e gastron
omia para uma sociedade
                mais inclusiva."
            </strong>
        </em>
    </p>
    <p>
        O atendimento é personalizado para cada situação
de sua vida. Cada prato
        é servido e preparado exclusivamente para cada me
sa de acordo com a
        sazonalidade dos ingredientes.
    </p>
</section>

```

Código-fonte 3.14 – Adição das classes nos elementos HTML do conteúdo principal

Fonte: Elaborado pelo autor (2020)

Vamos começar formatando a div do conteúdo principal. Aplicaremos uma cor de fundo verde e um padding de 30px.

```

.conteudo-principal {
    background-color: #28bb28;
    padding: 30px;
}

```

Código-fonte 3.15 – Determinação das configurações da classe principal

Fonte: Elaborado pelo autor (2020)

Na sequência, vamos centralizar o texto do título da seção:

```

.titulo-centralizado {
    text-align: center;
}

```

Código-fonte 3.16 – Configuração da classe titulo-centralizado
Fonte: Elaborado pelo autor (2020)

Prosseguindo na formatação, vamos garantir que todos os elementos p recebam o alinhamento de texto justificado (esse alinhamento não é recomendado em web, mas será aplicado para fins didáticos).

```
p {  
  text-align: justify;  
}
```

Código-fonte 3.17 – Configuração para o elemento p
Fonte: Elaborado pelo autor (2020)

Agora, formataremos a parte que representa a missão de nosso restaurante. Aplicaremos uma cor de fundo que possa gerar um contraste com a do elemento superior (section), deixaremos os cantos arredondados por meio da propriedade border-radius e aplicaremos uma sombra com a propriedade box-shadow. A propriedade box-shadow recebe os valores de deslocamento em x e y, sendo que valores positivos mantêm o sentido baixo à direita e negativos o oposto.

O terceiro parâmetro é o espalhamento dessa sombra (blur, em inglês) e a cor da sombra. Deixaremos a fonte maior por meio da propriedade font-size. Ao trabalharmos com o alinhamento de um box, não existe uma propriedade específica como text-align. Então, o truque é aplicar uma margem vertical, mesmo que seja 0, e para a margem horizontal utilizaremos a palavra auto. Dessa forma, o navegador preencherá as margens esquerda e direita com o mesmo tamanho, centralizado o box. Aplicaremos um padding de 15px, com o texto alinhado ao centro e ocuparemos a largura de 50% da página.

```
#missao {  
  background-color: #90e690;  
  border-radius: 5px;  
  box-shadow: 1px 1px 15px #003300;  
  font-size: 20px;  
  margin: 20px auto;  
  padding: 15px;  
  text-align: center;  
  width: 50%;  
}
```

Código-fonte 3.18 – Configuração para o elemento com id missão
Fonte: Elaborado pelo autor (2020)

Por fim, vamos alterar a cor do texto do elemento strong, dentro do elemento em. Para isso, combinaremos o nome dos dois elementos na sequência da hierarquia apresentada no HTML.

```
em strong {
  color: #003300;
}
```

Código-fonte 3.19 – Configuração para o elemento strong, filho de em
Fonte: Elaborado pelo autor (2020)

Podemos conferir o resultado dessa formatação por meio da figura “Formatação do conteúdo principal”.

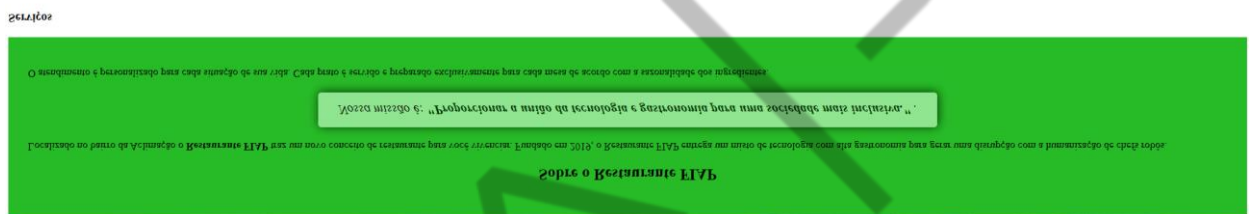


Figura 3.12 – Formatação do conteúdo principal
Fonte: Elaborado pelo autor (2020)

Pronto! Agora só falta a seção de serviços do nosso site.

3.6 Quais são os serviços oferecidos?

Para trabalharmos com a parte de serviços, que é nosso conteúdo secundário, vamos definir algumas classes de apoio na seção. Vejamos como fica o HTML.

```
<section class="servicos">
  <h3 class="titulo-centralizado">
    Serviços
  </h3>
  <ul>
    <li class="item">Carta de vinhos orgânicos</li>
    <li class="item">Pratos personalizados por mesa</li>
    <li class="item">Alta tecnologia em todos os ambientes</li>
    <li class="item">Espaço família em ambiente reservado</li>
    <li class="item">Respeito à sazonalidade dos alimentos</li>
  </ul>
```

```
  
</section>
```

Código-fonte 3.20 – Adição das classes na sessão de serviços
Fonte: Elaborado pelo autor (2020)

Ao aplicar a classe titulo-centralizado já obtemos a formatação anteriormente definida. O que queremos fazer agora é definir, no layout, que a lista fique posicionada à esquerda da imagem. Vamos trabalhar isso agora no CSS.

Começamos do plano de fundo da seção e estabelecendo um padding padrão para a seção.

```
.servicos {  
  background-color: #ffffff;  
  padding: 30px;  
}
```

Código-fonte 3.21 – Configuração da classe serviços
Fonte: Elaborado pelo autor (2020)

Na sequência, vamos trabalhar com o elemento da lista, o ul. Reforçando um pouco sobre o conceito base de HTML, cada elemento, basicamente, é um bloco retangular, indiferente à aparência exibida por meio do CSS. O ul não é diferente. Para ajustar a exibição dos elementos, utilizaremos uma propriedade chamada display. Essa propriedade é usada para trabalhar com o comportamento dos elementos HTML em relação à interação entre eles. Os possíveis valores para a propriedade display são:

- inline: valor padrão para elementos textos, em que cada um é exibido em frente ao outro. Na mesma linha.
- inline-block: similar ao inline, mas com a possibilidade de trabalhar com as propriedades de largura e altura do bloco.
- block: valor padrão para elementos containers tipo div, section, main, ul etc.
- flex: utilizado para trabalhar com um novo método de layout chamado Flexbox.
- grid: usado para trabalhar com um novo método de layout chamado Grid Layout.
- none: exclui o elemento da página.

No caso da nossa ul, vamos utilizar o display inline-block para que a nossa imagem possa ficar ao lado da ul. Vamos mantê-lo alinhado ao topo do bloco, já que a imagem será maior, e faremos isso por meio da propriedade vertical-align. Ajustaremos a largura e a margem à direita, para não ficar no limite com a imagem.

```
ul {  
  display: inline-block;  
  vertical-align: top;  
  width: 30%;  
  margin-right: 10%;  
}
```

Código-fonte 3.22 – Configuração do elemento ul
Fonte: Elaborado pelo autor (2020)

Por fim, faremos algumas pequenas formatações nos itens e na imagem da parte de serviço.

```
.item {  
  font-size: 20px;  
  font-style: italic;  
}  
  
.imagem-servicos {  
  width: 50%;  
  height: 300px;  
  object-fit: cover;  
  object-position: 0% 10%;  
}
```

Código-fonte 3.23 – Configuração da classe imagem-serviços
Fonte: Elaborado pelo autor (2020)

O resultado da formatação é apresentado na Figura “Formatação da área de serviços”.



Figura 3.13 – Formatação da área de serviços
Fonte: Elaborado pelo autor (2020)

Apesar de termos concluído a formatação da página principal, ainda temos muito a melhorar, como o espaçamento entre as seções, apresentado na Figura “Espaço em branco entre as seções do site”. Precisamos de um formulário para reserva, menu de navegação, rodapé e uma galeria de fotos dos jantares já servidos.



Figura 3.14 - Espaço em branco entre as seções do site
Fonte: Elaborado pelo autor (2020)

Vamos começar a discutir esses pontos na sequência.

3.7 Limpando as formatações

- Todo documento HTML já possui uma formatação preexistente para cada elemento. O ideal é, sempre que iniciarmos um projeto CSS, limpar todos os elementos de qualquer estilo preexistente. Para isso, podemos baixar um arquivo chamado reset.css que já vem com a limpeza minuciosa de cada elemento existente no HTML. Você pode encontrar esse arquivo no endereço: <<https://meyerweb.com/eric/tools/css/reset/>>.
- Vamos aplicar esse arquivo em nosso site, antes da importação de nosso arquivo de estilo. Veja como fica o código:

```
<title>Restaurante FIAP</title>  
<link rel="stylesheet" href="reset.css" type="text/css" />  
<link rel="stylesheet" href="style.css" type="text/css" />
```

Código-fonte 3.24 – Inclusão do arquivo reset.css para limpar as formatações padrões do navegador
Fonte: Elaborado pelo autor (2020)

- Confira o resultado na Figura “Resultado da formatação após a aplicação do arquivo reset.css”.

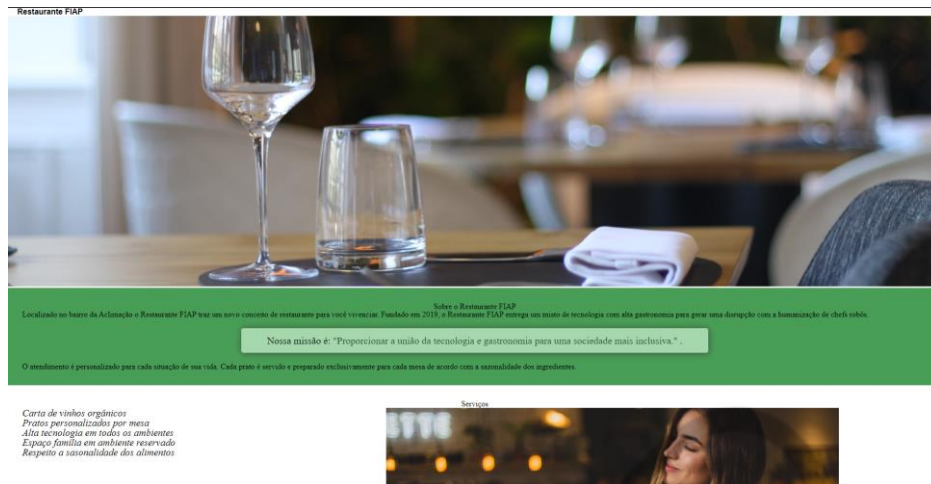


Figura 3.15 - Resultado da formatação após a aplicação do arquivo reset.css
Fonte: Elaborado pelo autor (2020)

Agora, precisamos acertar algumas questões. Vamos começar adicionando um ícone ao lado do logo e colocando um menu em nosso cabeçalho.

3.8 De volta ao cabeçalho: ícones e menu

- Para trabalharmos com ícones, há diversas bibliotecas disponíveis hoje na internet. Utilizaremos em nosso exemplo o Font Awesome, que é bem conhecido e possui uma versão gratuita para uso. Para o incluirmos ao nosso projeto, basta inserir a seguinte tag após o arquivo de reset, no head da página:

```
<link rel="stylesheet" href="https://pro.fontawesome.com/releases/v5.10.0/css/all.css" integrity="sha384-AYmEC3Yw5cVb3ZcuHtOA93w35dYTsvhLPVnYs9eStHfGJvOvKxVfELGroGkvsg+tp" crossorigin="anonymous"/>
```

Código-fonte 3.25 – Inclusão do CDN para o fontawesome, de modo a trabalhar com ícones
Fonte: Elaborado pelo autor (2020)

- E já vamos adicionar o ícone junto ao título principal, no cabeçalho. Adicionaremos também alguns elementos no HTML para melhorar a formatação de nosso cabeçalho. Veja como ficará nosso HTML.

```
<header class="cabecalho">
  <div class="container">
    <h1 class="titulo-principal">
      <i class="fas fa-seedling"></i>
      Restaurante FIAP
```



```
        </h1>
      </div>
    </header>
```

Código-fonte 3.26 – Inclusão de um ícone no cabeçalho

Fonte: Elaborado pelo autor (2020)

- O próximo passo é trabalhar com o CSS para formatarmos os elementos do nosso novo cabeçalho. Vamos analisá-los juntos, confira o código.

```
.cabecalho {
  background-color: #26522d;
  padding: 30px;
  height: 140px;
}

.container {
  margin: 0 auto;
  width: 900px;
}

.titulo-principal {
  color: #fff;
  font-family: Arial, Helvetica, sans-serif;
  font-size: 40px;
  font-weight: bold;
  margin-top: 50px;
  transition: all .5s ease-in-out;
}

.titulo-principal:hover {
  transform: translateX(50px) scale(1.2);
}
```

Código-fonte 3.27 – Configuração das classes que envolvem o cabeçalho

Fonte: Elaborado pelo autor (2020)

- Aplicamos ao cabeçalho um fundo verde bem escuro e uma altura de 140 pixels; somando-se com o padding, temos um total de 200px de área de trabalho. A classe container foi construída para delimitar a largura do conteúdo e deixá-lo centralizado. Isso facilitará em resoluções grandes.

- A maior mudança é no logo. Deixamos a cor da fonte branca para contrastar com o fundo verde, aumentamos a fonte para 40px. Assim, podemos aplicar uma margem de 50px no topo para centralizar verticalmente no cabeçalho. Aplicamos uma nova propriedade ao título, chamada transition, que permite aos elementos alterarem as propriedades com um determinado tempo e efeito de animação. No exemplo, queremos que todas as propriedades sofram esse efeito (all), o tempo é meio segundo e a animação possui um efeito para suavizar tanto a entrada quanto a saída da execução da animação (ease-in-out).
- Logo após a classe titulo-principal, existe a repetição da classe com a seguinte informação “:hover”. O que seria essa informação? O hover é uma pseudoclassee que aplica o efeito no elemento quando posicionamos o mouse sobre ele. Aqui, o efeito é executar a transformação da escala do elemento em 20% e a movimentação no eixo X em 50px para a direita. A movimentação é para corrigir o efeito de deslocamento causado a partir da escala. A Figura “Aplicação do efeito hover no título” apresenta como ficou o cabeçalho com o efeito do mouseover aplicado.



Figura 3.16 – Aplicação do efeito hover no título
Fonte: Elaborado pelo autor (2020)

Para finalizarmos o cabeçalho, vamos criar o menu para navegação entre as páginas do projeto. Modificaremos o HTML para inserir nosso menu.

```
<div class="container">
  <h1 class="titulo-principal">
    <i class="fas fa-seedling"></i>
    Restaurante FIAP
  </h1>
```

```
<nav>
  <ul>
    <li><a href="index.html">Home</a></li>
    <li><a href="galeria.html">Galeria</a></li>
    <li><a href="contato.html">Contato</a></li>
  </ul>
</nav>
</div>
```

Código-fonte 3.28 – Inclusão de um elemento nav para criação do menu

Fonte: Elaborado pelo autor (2020)

Para posicionarmos nosso menu, vamos utilizar uma nova propriedade chamada position. Essa propriedade tem o efeito de um post-it, aquele adesivo para colocar avisos em vários lugares. Existem três valores possíveis para a propriedade position:

- fixed: o elemento ficará fixo de acordo com as coordenadas estipuladas. Indiferente em relação ao restante dos elementos.
- relative: faz com que o posicionamento seja tratado a partir do canto esquerdo superior do próprio elemento e não mais com o navegador como referência.
- absolute: como referência de posicionamento, usa a referência do elemento pai.

Sendo assim, para conseguirmos posicionar nosso menu no cabeçalho, vamos transformar a posição da div container para relative. Dessa maneira, o elemento nav vai se referenciar por ela.

```
.container {
  margin: 0 auto;
  width: 940px;
  position: relative;
/* Faz a camada de container ser referencia para o position absolute do nav. Auxiliando o posicionamento dentro da div */
}
```

Código-fonte 3.29 – Trabalhando no posicionamento do nav a partir de seu elemento pai

Fonte: Elaborado pelo autor (2020)

Agora, podemos posicionar o nav e já formatarmos nosso menu.

```
nav {
  position: absolute;
  top: 20px;
```

```
        right: 0;
        width: 50%;
    }

    nav ul {
        /* remove alguns espaços em branco entre li */
        font-size: 0;
        width: 100%;
        text-align: right;
    }

    nav li {
        display: inline-block;
        border-right: 2px #fff solid;
        padding: 8px 0;
    }

    nav li:last-child {
        border-right: 0;
    }

    nav a {
        font-family: Arial, Helvetica, sans-serif;
        color: #fff;
        font-size: 22px;
        text-decoration: none;
        text-transform: uppercase;
        padding: 8px 16px;
    }

    nav a:hover {
        text-decoration: underline;
        background-color: #3e5241;
    }
```

Código-fonte 3.30 – Configuração CSS para o menu
Fonte: Elaborado pelo autor (2020)

A propriedade `position absolute` trabalha com as coordenadas passadas por meio das propriedades `top`, `left`, `bottom` e `right`. No exemplo, fixamos a 20px do topo e 0px da direita. Definimos também uma largura de 50% para ocupar metade do cabeçalho. Outro ponto interessante na formatação é a pseudoclasse `last-child`. Ela modifica o último elemento da tag `li` dentro da tag `nav`. O resultado do cabeçalho é apresentado na Figura “Resultado do cabeçalho”.

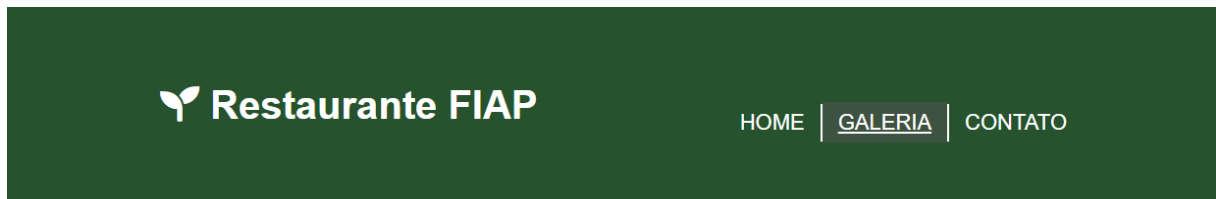


Figura 3.17 – Resultado do cabeçalho
Fonte: Elaborado pelo autor (2020)

Agora, ajustaremos o conteúdo principal basicamente no HTML, vamos colocá-lo em um container para manter a largura determinada no cabeçalho e assegurar um padrão no design. A consistência no design é algo muito importante para garantir identidade visual e harmonia no sistema ou website.

```
.conteudo-principal {
  background-color: #499e57;
  color: white;
  padding: 30px;
  margin-top: 0;
}

.titulo-centralizado {
  text-align: center;
}

.titulo-secao {
  border-bottom: 1px lightgray dashed;
  font-family: Arial, Helvetica, sans-serif;
  font-size: 26px;
  margin-bottom: 16px;
  padding: 16px;
}

p {
  text-align: justify;
  font-size: 20px;
}

#missao {
  background-color: #a5d9ad;
  border-radius: 5px;
  box-shadow: 1px 1px 15px #3e5241;
  color: #000;
  font-size: 20px;
  margin: 20px auto;
  padding: 15px;
  text-align: center;
  width: 70%;
}
```

```
em strong {  
    color: #26522d;  
    font-style: italic;  
}
```

Código-fonte 3.31 – Configuração da seção de conteúdo principal
Fonte: Elaborado pelo autor (2020)

Confira o resultado da reformatação na Figura “Reformatação do conteúdo principal após a aplicação do reset.css”.



Figura 3.18 – Reformatação do conteúdo principal após a aplicação do reset.css
Fonte: Elaborado pelo autor (2020)

Agora, precisamos formatar a área de serviços, primeiro, para colocá-la no container e, depois, para ajustar os bullets (marcadores de tópicos) da lista, que sumiram quando fizemos a limpeza do CSS padrão.

```
ul {  
    display: inline-block;  
    vertical-align: top;  
    width: 30%;  
    margin-right: 19.5%;  
}  
  
li {  
    list-style: square;  
    margin-left: 20px;  
}
```

Código-fonte 3.32 – Configuração da lista na área de serviços
Fonte: Elaborado pelo autor (2020)

Alteramos apenas os elementos da lista para solucionar o problema gerado pela limpeza do CSS. Perceba que é possível utilizar valores quebrados nas medidas,

como no caso da margem à direita do elemento `ul`. A propriedade `list-style` do elemento `li` define qual é o ícone que aparecerá no tópico. Vamos conferir o resultado da formatação na seção de serviços de nossa home page, apresentado na Figura “Resultados da formatação da seção de serviços”.

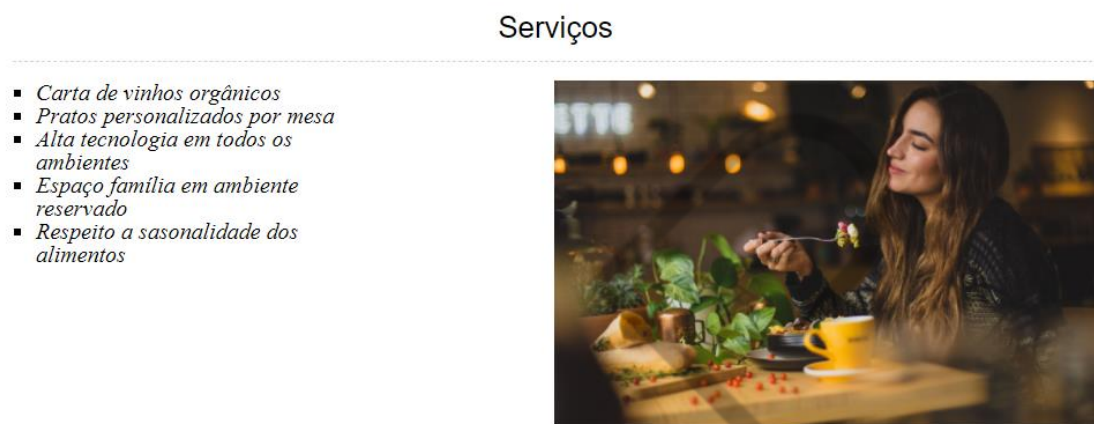


Figura 3.19 – Resultado da formatação da seção de serviços
Fonte: Elaborado pelo autor (2020)

Para finalizar, vamos inserir um rodapé na página. É uma prática muito boa para trazer a sensação de finalizar o conteúdo exibido. Após a seção de Serviços, inserimos o seguinte código HTML.

```
<footer>
  <div class="container">
    <p class="copyright">
      Feito com <span>&hearts;</span> por FIAP | Co
pyright &copy; 2020 - Restaurante FIAP
    </p>
  </div>
</footer>
```

Código-fonte 3.33 – Criação do footer
Fonte: Elaborado pelo autor (2020)

Neste momento, só precisamos formatar a classe `copyright`. Vamos conferir como fica o código.

```
footer {
  background-color: #26522d;
  padding: 30px;
  height: 40px;
}
```

```
.copyright {  
    color: white;  
    font: 18px Arial, Helvetica, sans-serif;  
    text-align: center;  
    margin-top: 10px;  
}  
  
.copyright span {  
    color: red;  
    font-size: 22px;  
}
```

Código-fonte 3.34 – Classes para configurar o rodapé
Fonte: Elaborado pelo autor (2020)

A Figura “Resultado do rodapé” apresenta o rodapé da página.

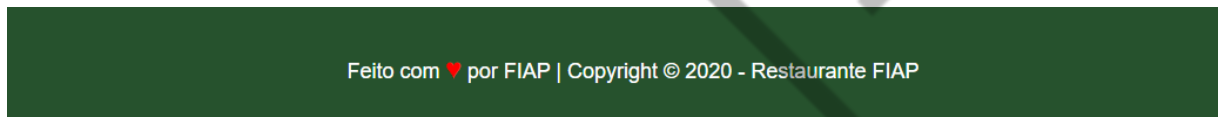


Figura 3.20 – Resultado do rodapé
Fonte: Elaborado pelo autor (2020)

Com a página principal finalizada, podemos cuidar da nossa galeria de fotos. É o que veremos na próxima seção.

3.9 Flexibilizando a galeria

Para começar a página da galeria, vamos aproveitar a estrutura principal da página principal. O HTML deverá ficar da seguinte maneira:

```
<!DOCTYPE html>  
<html lang="pt-br" dir="ltr">  
  <head>  
    <meta charset="utf-8" />  
    <title>Restaurante FIAP</title>  
    <link rel="stylesheet" href="reset.css" type="text/css" />  
    <link  
      rel="stylesheet"  
      href="https://pro.fontawesome.com/releases/v5.10.0/css/all.css"  
      integrity="sha384-AYmEC3Yw5cVb3ZcuHtOA93w35dYTsvhLPVnYs9eStHfGJvOvKxVfELGroGkvsg+p"  
      crossorigin="anonymous">
```



```

        />
        <link rel="stylesheet" href="style.css" type="text/css" />
    </head>
    <body>
        <header class="cabecalho">
            <div class="container">
                <h1 class="titulo-principal">
                    <i class="fas fa-seedling"></i>
                    Restaurante FIAP
                </h1>
                <nav>
                    <ul>
                        <li><a href="index.html">Home</a>
                    </li>
                        <li><a href="galeria.html">Galeria</a>
                    </li>
                        <li><a href="contato.html">Contato</a>
                    </li>
                    </ul>
                </nav>
            </div>
        </header>
        <section class="galeria">
            <div class="container">
                <h3 class="titulo-centralizado">Galeria de Fotos</h3>
            </div>
        </section>
        <footer>
            <div class="container">
                <p class="copyright">
                    Feito com <span>&hearts;</span> por FIAP | Copyright &copy; 2020 -
                    Restaurante FIAP
                </p>
            </div>
        </footer>
    </body>
</html>

```

Código-fonte 3.35 – Código HTML da página da galeria
 Fonte: Elaborado pelo autor (2020)

Vamos criar agora uma estrutura de cards para exibir as imagens que nossos clientes compartilham nas redes sociais. Na estrutura, haverá uma div que representará a linha com as fotos e mais três divs dentro, como um container para cada foto.

```
<div class="linha-foto">
```

```
<div class="foto"></div>
<div class="foto"></div>
<div class="foto"></div>
</div>
```

Código-fonte 3.36 – Seção para criar a linha de imagens da galeria
Fonte: Elaborado pelo autor (2020)

O próximo passo é trabalhar com o CSS para formatar a galeria. Para isso, vamos utilizar o flexbox, que é um novo meio para auxiliar na disposição dos elementos dentro do HTML. Um guia completo do flexbox é apresentado no endereço: <https://css-tricks.com/snippets/css/a-guide-to-flexbox/>. Vamos conferir como ficou o código CSS.

```
.linha-fotos {
  display: flex;
  flex-direction: row;
  justify-content: space-between;
  margin-bottom: 20px;
  padding-bottom: 20px;
  border-bottom: 1px rgba(0, 0, 0, 0.1) solid;
}

.foto {
  border: 2px lightgray solid;
  padding: 8px;
}

.foto img {
  width: 280px;
  height: 180px;
  object-fit: cover;
  transition: all 1s ease-in-out;
}

.foto img:hover {
  transform: scale(1.5) rotate(10deg);
}
```

Código-fonte 3.37 – Configuração das classes para a galeria
Fonte: Elaborado pelo autor (2020)

Apesar de utilizarmos o flex como um valor da propriedade display, ele não é apenas uma pequena propriedade. Flexbox é um módulo para a formatação de layout completo, que veio facilitar o trabalho de posicionamento dos elementos e seus

containers. A propriedade flex-direction nos permite trabalhar com os elementos nas direções: vertical (column) ou horizontal (row). Além disso, essa propriedade possui os valores reversos que podem ser alinhados no sentido oposto ao natural da leitura ocidental (de cima para baixo, da esquerda para a direita). Além disso, aplicamos a propriedade justify-content, que auxilia o posicionamento entre os elementos dentro do container (linha-fotos). No exemplo, foi aplicado space-between, que coloca um espaço proporcional entre os elementos até preencher toda a largura do container. Ainda podemos utilizar os valores flex-start (alinhar à esquerda), flex-end (alinhar à direita), center, space-around (distribuir os espaços em volta dos elementos) e space-evenly (distribuir os espaços entre dois elementos igualmente).

Além do flex, adicionamos um efeito de escala com rotação nas fotos ao passar o mouse por cima. Veja, na Figura “Primeira linha da galeria de foto formatada”, como ficou a primeira linha da nossa galeria.



Figura 3.21 – Primeira linha da galeria de foto formatada
Fonte: Elaborado pelo autor (2020)

Vamos completar a galeria com mais duas linhas, mantendo nove fotos ao total.

```
<div class="linha-fotos">
  <div class="foto">
    
  </div>
  <div class="foto">
    
  </div>
  <div class="foto">
    
  </div>
</div>
<div class="linha-fotos">
```

```
<div class="foto">
  
</div>
<div class="foto">
  
</div>
<div class="foto">
  
</div>
</div>
<div class="linha-fotos">
  <div class="foto">
    
  </div>
  <div class="foto">
    
  </div>
  <div class="foto">
    
  </div>
</div>
```

Código-fonte 3.38 – Replicação da linha para criar a galeria de fotos

Fonte: Elaborado pelo autor (2020)

Já conferimos o resultado da galeria completa na Figura “Galeria de fotos finalizada”, apresentada a seguir.

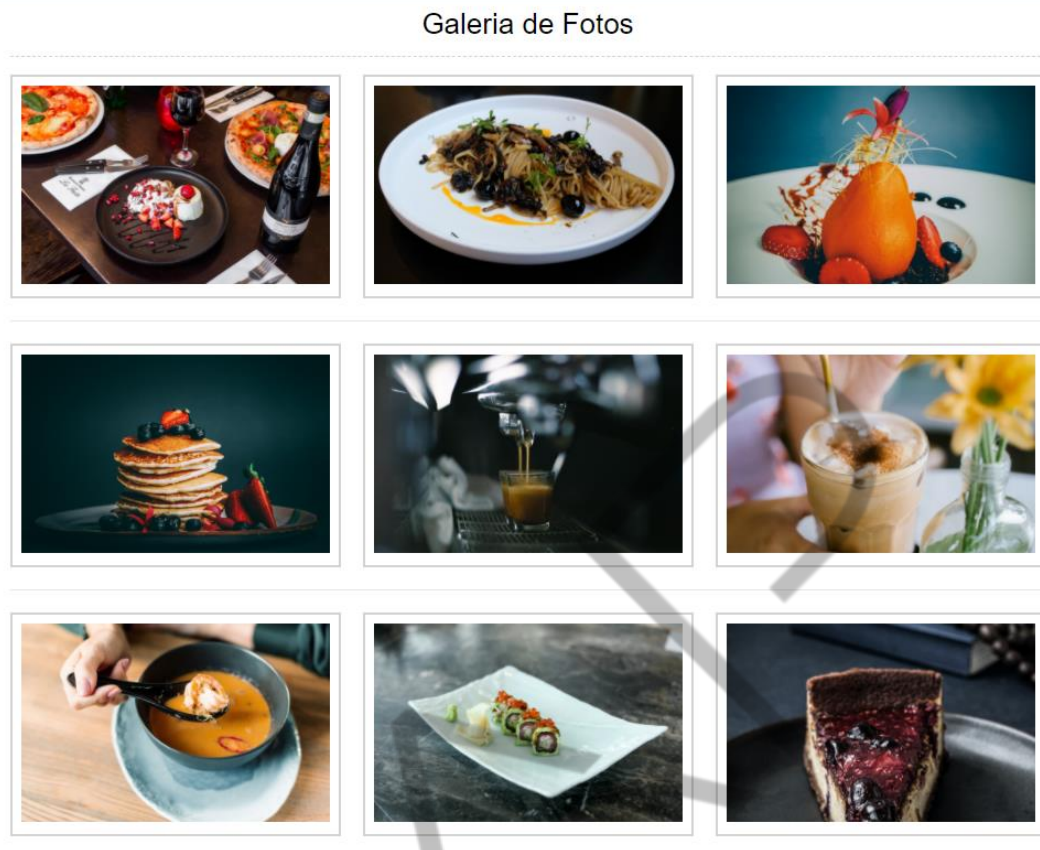


Figura 3.22 – Galeria de fotos finalizada
Fonte: Elaborado pelo autor (2020)

Para finalizar o site e entregá-lo ao cliente, precisamos criar o formulário de contato.

3.10 Entrando em contato

Utilizando a mesma base de contato, vamos construir o HTML do contato para reserva.

```
<form>
<label for="nome">Nome:</label>
<input
  type="text"
  id="nome"
  name="nome"
  placeholder="Digite o nome do responsável pela reserva"
/>
<label for="telefone">Telefone:</label>
```

```
<input
  type="tel"
  id="telefone"
  name="telefone"
  placeholder="Digite o telefone de contato"
/>
<label for="qtdPessoas">Mesa para quantos:</label>
<input
  type="number"
  id="qtdPessoas"
  name="qtdPessoas"
  step="1"
  placeholder="Digite a quantidade de pessoas na mesa"
/>
<label for="observacoes">Informações adicionais:</label>
<textarea
  id="observacoes"
  name="observacoes"
  placeholder="Informe suas restrições alimentares"
></textarea>
<button type="submit"><i class="far fa-paper-
plane"></i> Enviar</button>
</form>
```

Código-fonte 3.39 – Código HTML para o formulário da página de contato
Fonte: Elaborado pelo autor (2020)

Vamos formatar o nosso formulário agora, continuando a utilizar o flexbox para auxiliar o layout.

```
form {
  display: flex;
  flex-direction: column;
}

label {
  color: #26522d;
  font: 18px Arial, Helvetica, sans-serif;
  padding: 8px 0;
}

input,
textarea {
  border-radius: 4px;
  font: 16px Arial, Helvetica, sans-serif;
  padding: 8px 16px;
}

textarea {
```

```
        height: 200px;
        resize: none;
    }

    ::placeholder {
        color: #499e57;
        opacity: 0.5;
    }

    button {
        align-self: flex-end;
        background: linear-
gradient(to right bottom, #499e57, #26522d);
        border: 0;
        border-radius: 4px;
        color: #fff;
        cursor: pointer;
        display: block;
        font: 18px Arial, Helvetica, sans-serif;
        margin: 10px 0;
        padding: 12px 16px;
        width: 200px;
    }

    button:hover {
        background: rgba(38, 82, 45, 0.8);
    }

    button i {
        margin: 0 5px 0 -5px;
    }
```

Código-fonte 3.40 – Código CSS para o formulário da página de contato
Fonte: Elaborado pelo autor (2020)

Vamos utilizar o próprio form como container e a propriedade flex-direction como column para um elemento ficar em cima do outro. Outra propriedade que ainda não utilizamos nas demais formatações é o resize, que aplicaremos no textarea. O textarea possibilita ao usuário modificar seu tamanho clicando nele e arrastando-o, com o resize none, impedimos isso e preservamos nosso layout de ser distorcido com a ação do usuário. A pseudoclassee placeholder permite a formatação dos placeholders de todos os elementos de entrada de dados. Por fim, usaremos a função linear-gradient para deixar o fundo do botão com um efeito de transição de cor.

A Figura “Resultado do formulário formatado” apresenta o resultado do nosso formulário.

Faça sua Reserva

Nome:

Telefone:

Mesa para quantos:

Informações adicionais:

Enviar

Figura 3.23 – Resultado do formulário formatado
Fonte: Elaborado pelo autor (2020)

Agora você já tem todas as ferramentas para atribuir cada vez mais estilos a suas aplicações. Mas não ache que essas ferramentas são as únicas. Todos os dias são lançadas e aprimoradas novas features no CSS. Fique sempre atento!

REFERÊNCIAS

W3SCHOOLS. **CSS** **Tutorial**. Disponível em:
<<https://www.w3schools.com/css/default.asp>>. Acesso em: 28 ago. 2020.

EMASP