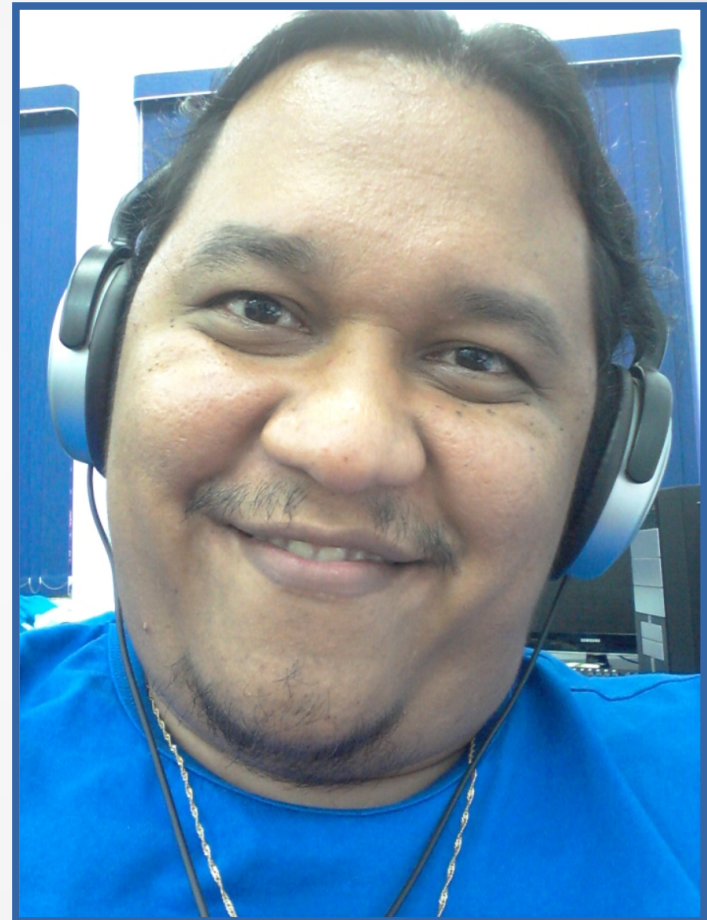


## 2. HTML Semântico e posicionamento



# Instrutor

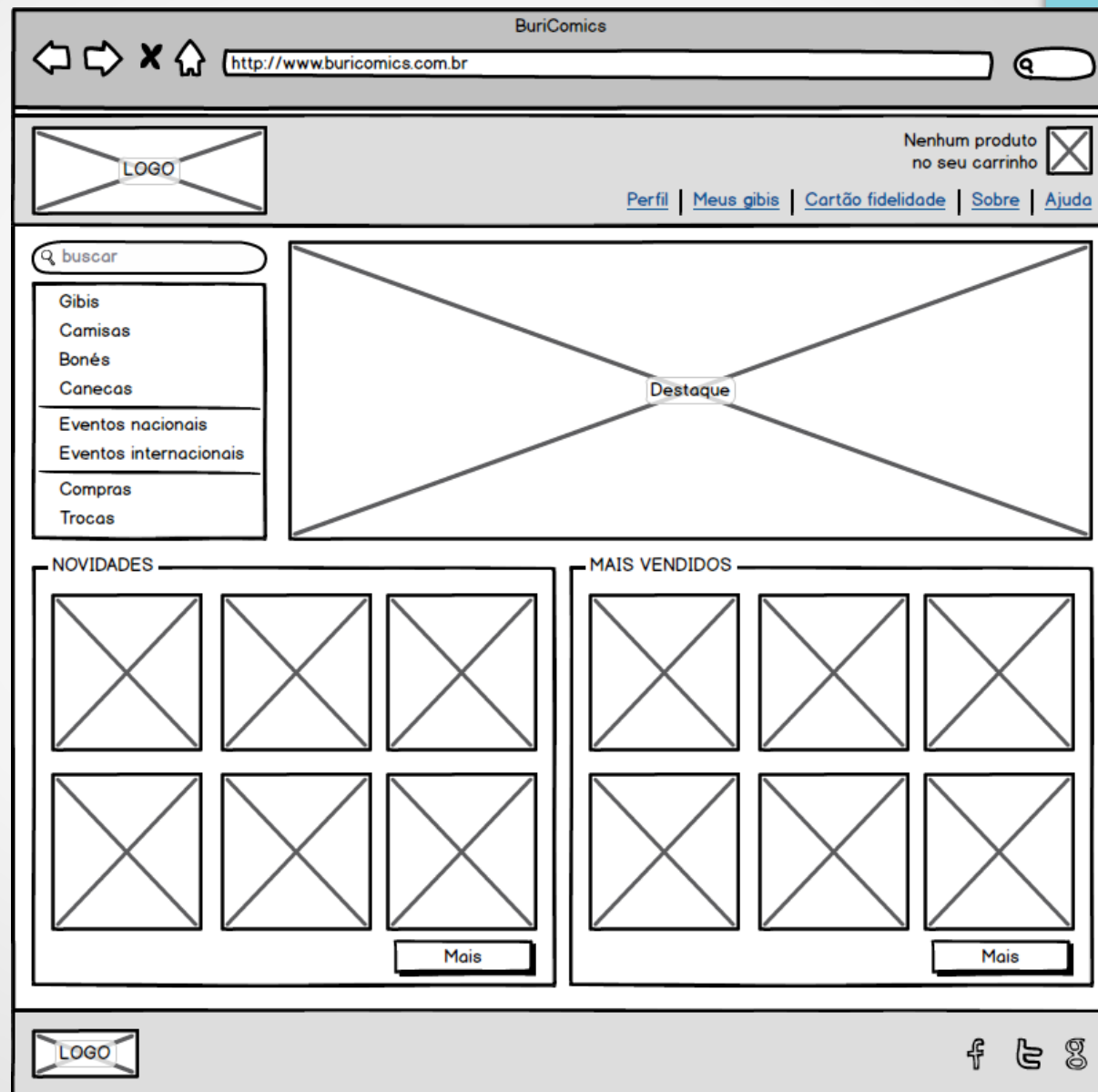
- O problema de filtros e consultas dinâmicas
- Filtros baseados em concatenação de Strings
- Melhorando o código com StringBuider
- Consultas baseadas em Critéria



# Agenda

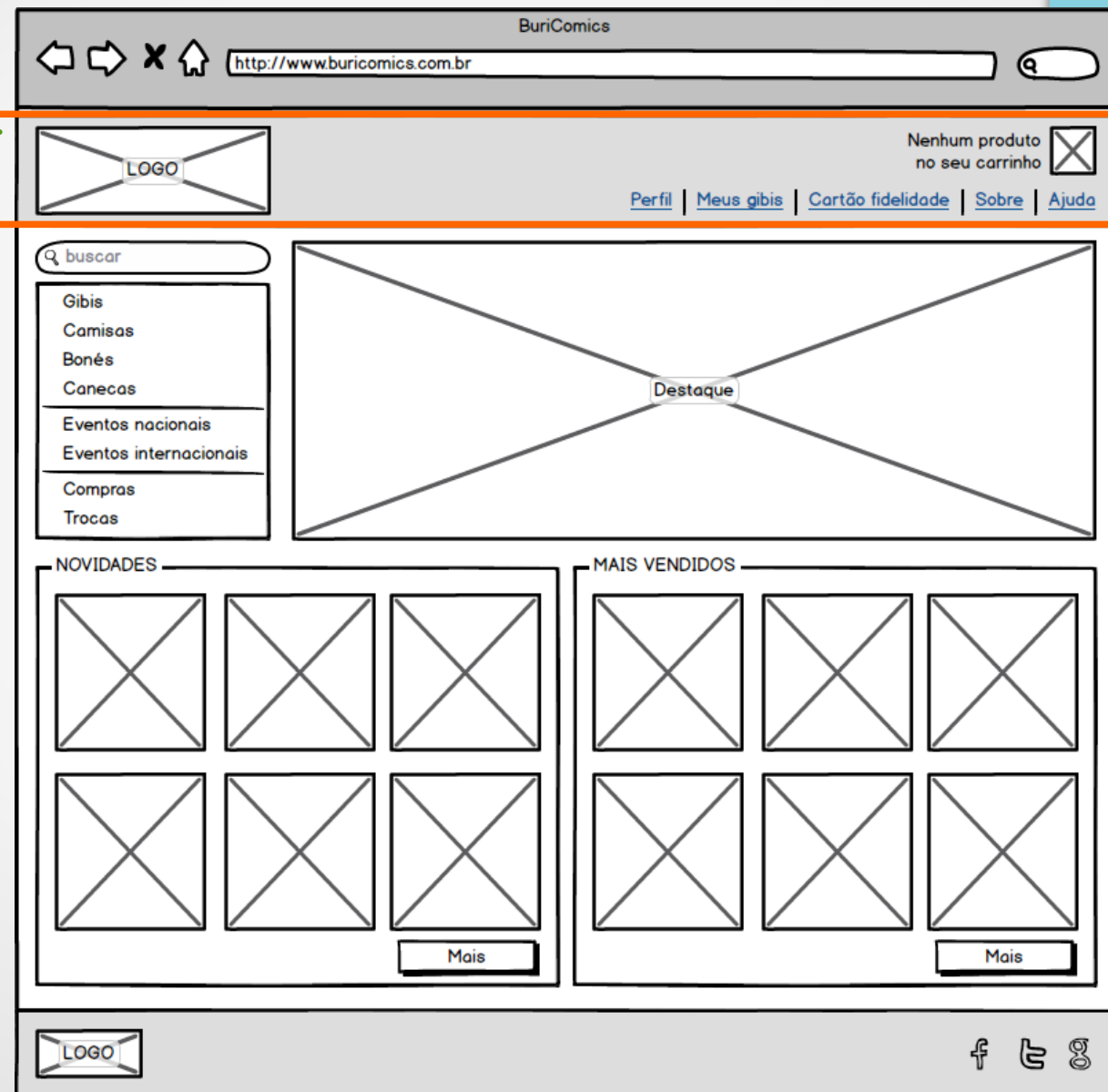
- Definição de padrões
- HTML semântico
- CSS Reset
- Blocks você Inline

# Pensando no projeto - wireframe



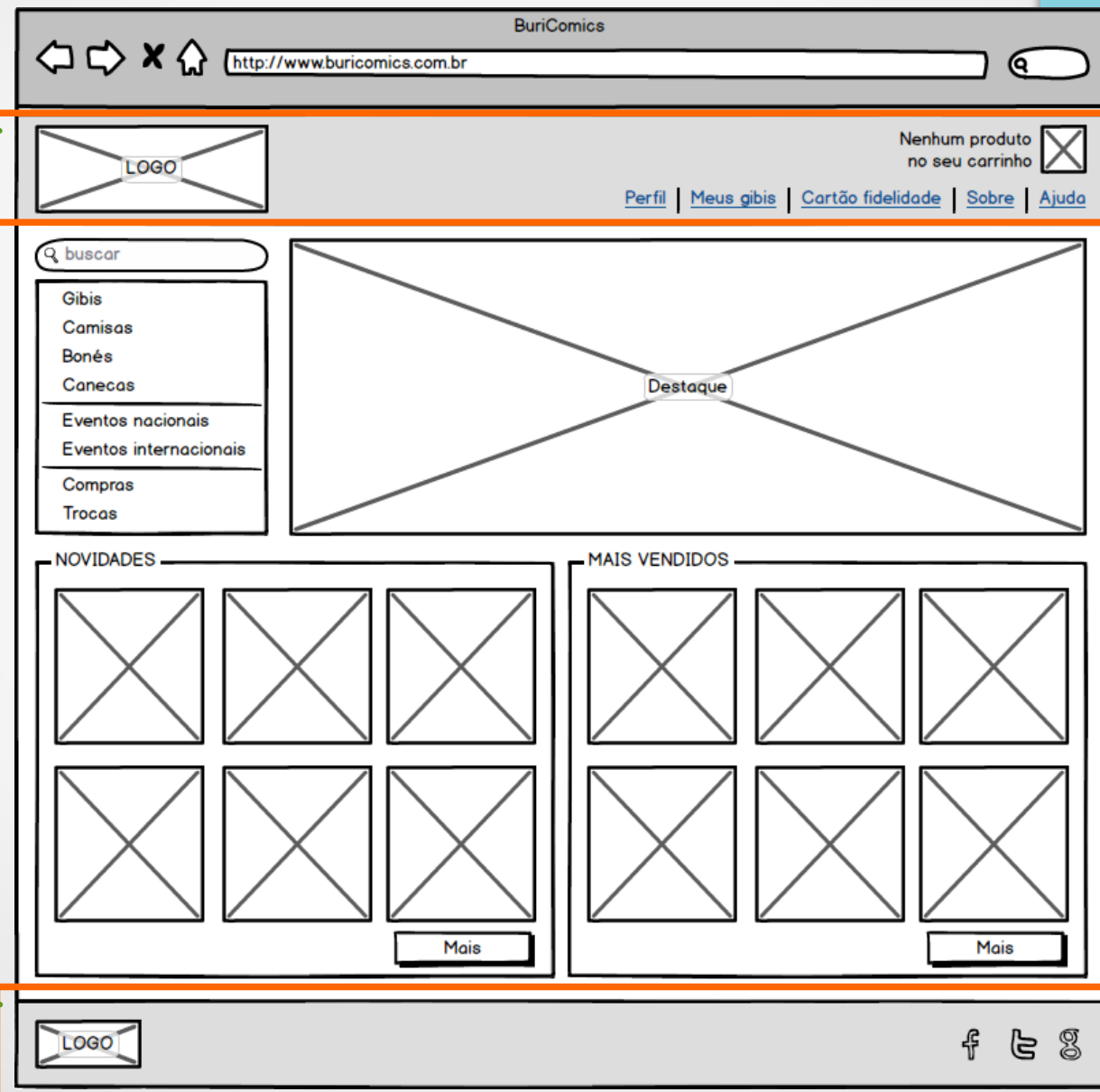
# Pensando no projeto - wireframe

TOPO DA  
PÁGINA



# Pensando no projeto - wireframe

**TOPO** DA  
PÁGINA



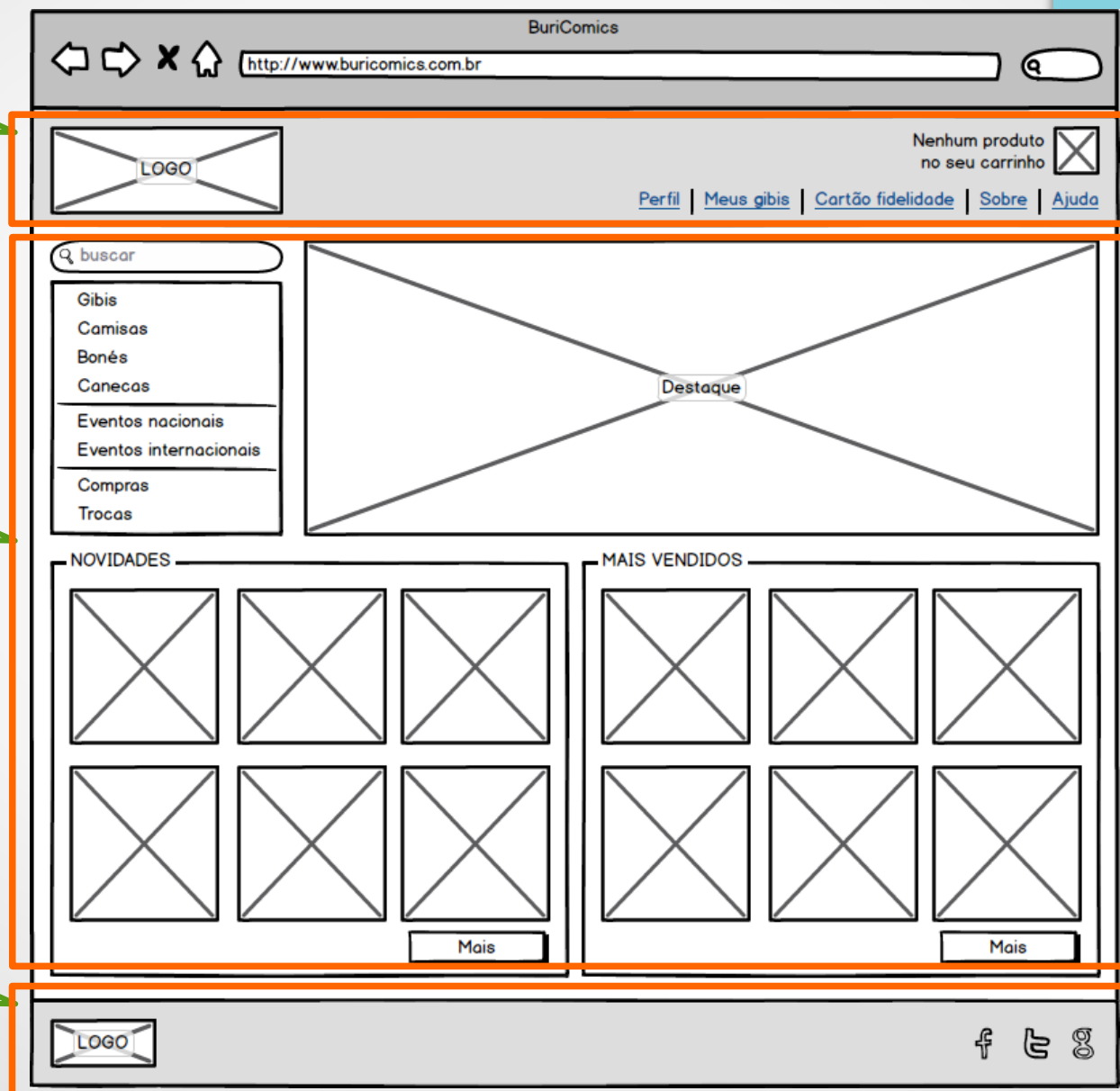
**RODAPE** DA  
PÁGINA

# Pensando no projeto - wireframe

**TOPO** DA  
PÁGINA

**CONTEÚDO**  
DA PÁGINA

**RODAPE** DA  
PÁGINA



# Pensando na implementação

- Em nossa página, temos uma clara **divisão de seções**:
  - Cabeçalho, conteúdo principal e rodapé
- Poderíamos usar as novas **TAGs do HTML 5** para definir as seções do nosso site, ao invés de **DIVs**

```
<body>
  <!-- Layout baseado em novas tags do HTML 5 -->
  <header>
    <!-- Conteúdo do bloco de cabeçalho -->
  </header>
  <section>
    <!-- Conteúdo principal -->
  </section>
  <footer>
    <!-- Conteúdo do rodapé -->
  </footer>
</html>
```



# Pensando no assunto

- Não há **diferença funcional** entre as **DIVs** e as novas **TAGs**
- Contudo, as TAGs **header, section e footer** apresentam **significado**. E isso é importante.
- As novas TAGs do HTML 5 trazem **significado semântico**
- Com o HTML semântico, o programador pode **inferir** a **importância** de uma determinada seção do site
- Essa característica auxilia no trabalho de **máquinas de busca**

# Exercício 01: header

- Inclua a tag `header` na página `index.html`

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

  <!-- Aplicação de CSS baseado em classes -->
  <p class="carrinho">Nenhum item no carrinho de compras</p>

  <!-- Definição do conjunto de links de navegação -->
  <nav class="menu">
    <ul>
      <li><a href="#">Perfil</a></li>
      <li><a href="#">Meus gibis</a></li>
      <li><a href="#">Cartão fidelidade</a></li>
      <li><a href="#">Sobre</a></li>
      <li><a href="#">Ajuda</a></li>
    </ul>
  </nav>
</header>
```

# Exercício 01: head

Usamos **H1** para evidenciar a importância semântica da imagem

- Inclua a tag **header** na página **index.html**

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

  <!-- Aplicação de CSS baseado em classes -->
  <p class="carrinho">Nenhum item no carrinho de compras</p>

  <!-- Definição do conjunto de links de navegação -->
  <nav class="menu">
    <ul>
      <li><a href="#">Perfil</a></li>
      <li><a href="#">Meus gibis</a></li>
      <li><a href="#">Cartão fidelidade</a></li>
      <li><a href="#">Sobre</a></li>
      <li><a href="#">Ajuda</a></li>
    </ul>
  </nav>
</header>
```

# Exercício 01: head

Usamos **H1** para evidenciar a importância semântica da imagem

- Inclui conteúdo textual da imagem na página `index.html`

Conteúdo textual da imagem

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

  <!-- Aplicação de CSS baseado em classes -->
  <p class="carrinho">Nenhum item no carrinho de compras</p>

  <!-- Definição do conjunto de links de navegação -->
  <nav class="menu">
    <ul>
      <li><a href="#">Perfil</a></li>
      <li><a href="#">Meus gibis</a></li>
      <li><a href="#">Cartão fidelidade</a></li>
      <li><a href="#">Sobre</a></li>
      <li><a href="#">Ajuda</a></li>
    </ul>
  </nav>
</header>
```

# Exercício 01: head

Usamos **H1** para evidenciar a importância semântica da imagem

- Inclui o conteúdo textual da imagem na página

Conteúdo textual da imagem

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

  <!-- Aplicação de CSS baseado em classes -->
  <p class="carrinho">Nenhum item no carrinho de compras</p>

  <!-- Definição do conjunto de links de navegação -->
  <nav class="menu">
    <ul>
      <li><a href="#">Perfil</a></li>
      <li><a href="#">Meus gibis</a></li>
      <li><a href="#">Cartão fidelidade</a></li>
      <li><a href="#">Sobre</a></li>
      <li><a href="#">Ajuda</a></li>
    </ul>
  </nav>
</header>
```

Texto do carrinho de compras

# Exercício 01: head

Usamos **H1** para evidenciar a importância semântica da imagem

- Inclui conteúdo textual da imagem na página `index.html`

Conteúdo textual da imagem

Estilo baseado em classes

Texto do carrinho de compras

```
<header>
```

```

```

```
<!-- Aplicação de CSS baseado em classes -->
```

```
<p class="carrinho">Nenhum item no carrinho de compras</p>
```

```
<!-- Definição do conjunto de links de navegação -->
```

```
<nav class="menu">
```

```
<ul>
```

```
<li><a href="#">Perfil</a></li>
```

```
<li><a href="#">Meus gibis</a></li>
```

```
<li><a href="#">Cartão fidelidade</a></li>
```

```
<li><a href="#">Sobre</a></li>
```

```
<li><a href="#">Ajuda</a></li>
```

```
</ul>
```

```
</nav>
```

```
</header>
```

# Exercício 01: head

- Inclui conteúdo textual da imagem na página `index.html`

Usamos **H1** para evidenciar a importância semântica da imagem

Conteúdo textual da imagem

Estilo baseado em classes

Texto do carrinho de compras

Conjunto de links

```
<header>
  
  <!-- Aplicação de classes -->
  <p class="cart">Item no carrinho de compras</p>
  <!-- Definição do conjunto de links de navegação -->
  <nav class="menu">
    <ul>
      <li><a href="#">Perfil</a></li>
      <li><a href="#">Meus gibis</a></li>
      <li><a href="#">Cartão fidelidade</a></li>
      <li><a href="#">Sobre</a></li>
      <li><a href="#">Ajuda</a></li>
    </ul>
  </nav>
</header>
```

# Exercício 02: Atualização do CSS

- Crie o arquivo `estilo.css`, adicione o código abaixo

```
3=body {
4    color: #333333;
5    font-family: "Lucida Sans Unicode", "Lucida Grande", sans-serif;
6 }
7=.carrinho {
8    background: url("../img/carrinho.png") no-repeat top right;
9    font-size: 14px;
10   padding-right: 35px;
11   text-align: right;
12   width: 140px;
13   padding-top: 5px
14 }
15
16=.menu ul {
17     font-size: 15px;
18 }
19
20=.menu a {
21     color: #003366;
22 }
```



# Exercício 02: Atualização do CSS

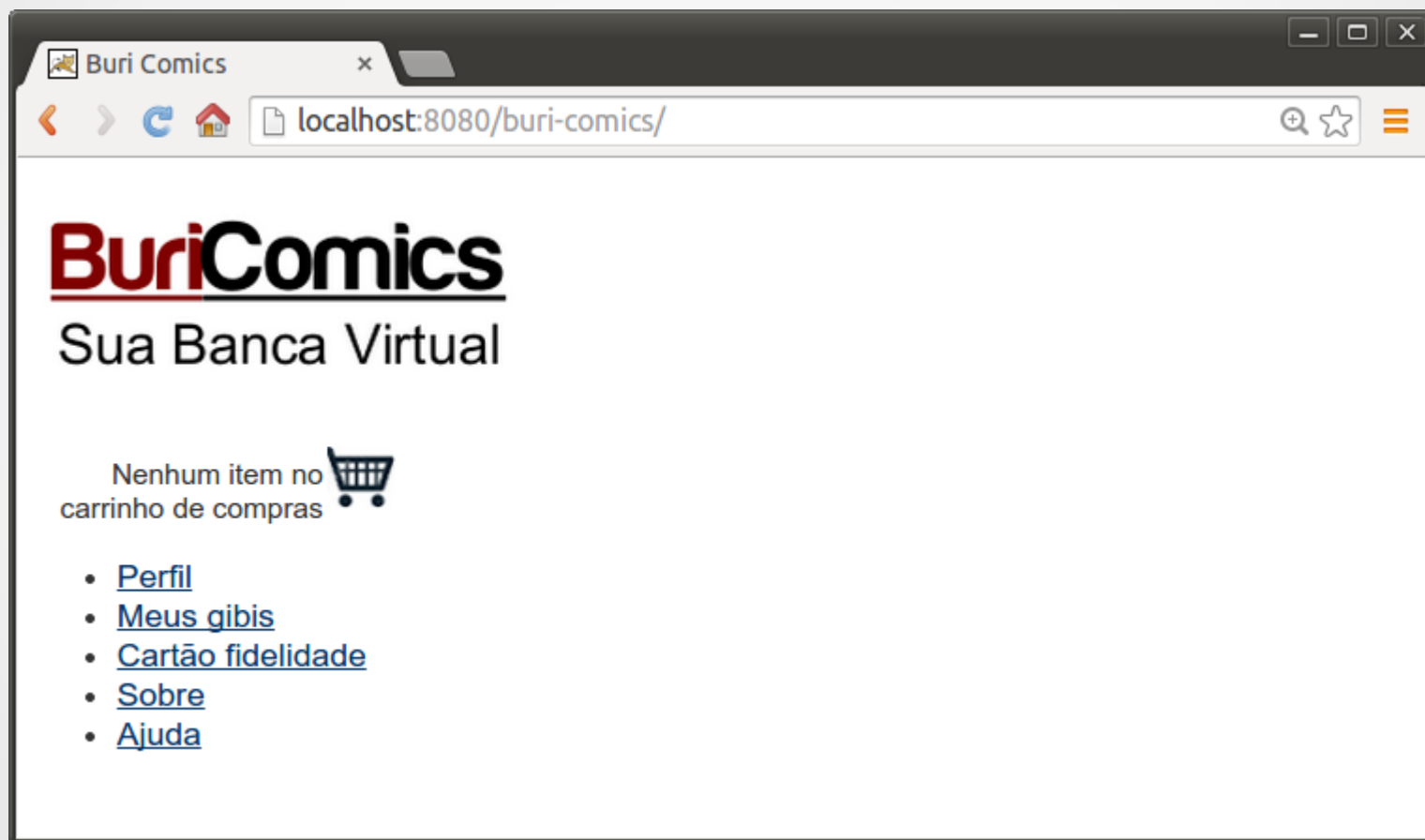
- Crie o arquivo `estilo.css`, adicione o código abaixo

```
3=body {
4    color: #333333;
5    font-family: "Lucida Sans Unicode", "Lucida Grande", sans-serif;
6 }
7=.carrinho {
8    background: url("../img/carrinho.png") no-repeat top right;
9    font-size: 14px;
10   padding-right: 35px;
11   text-align: right;
12   width: 140px;
13   padding-top: 5px;
14 }
15
16=.menu ul {
17   font-size: 15px;
18 }
19
20=.menu a {
21   color: #003366;
22 }
```

- Adicione o **CSS** na `index.html`

```
<title>Buri Comics</title>
<meta charset="utf-8" />
<link rel="stylesheet" href="css/estilo.css" />
</head>
<body>
    <header class="container">
        <h1>
```

# Resultado esperado



# CSS Reset

- Todo navegador tem uma série de **configurações próprias**, que são aplicadas em componentes sem estilo
- Para evitar problemas como quebras de layout sinistras, podemos utilizar estilos conhecidos como **CSS Reset**
- Com o **CSS Reset**, definimos os valores iniciais para todas as características do CSS.
- Dentre as iniciativas nessa área, podemos citar
  - HTML5 Boilerplate, YUI3 CSS Reset e
  - **Eric Meyer CSS Reset**, o nosso escolhido, baixado em:
    - <http://meyerweb.com/eric/tools/css/reset>

# Block vs Inline

- Comportamento de elementos HTML
- **Block**: ocupa toda a largura disponível
  - Exemplos `<h1>` a `<h6>`, `<p>` e `<div>`
- **Inline**: ocupa apenas o espaço necessário
  - Exemplos `<a>`, `<small>`, `<strong>`, `<em>`
- A tag `<li>` de `<ul>` é definida como **block**, por padrão
- Podemos mudar os padrões, usando a propriedade **display**.

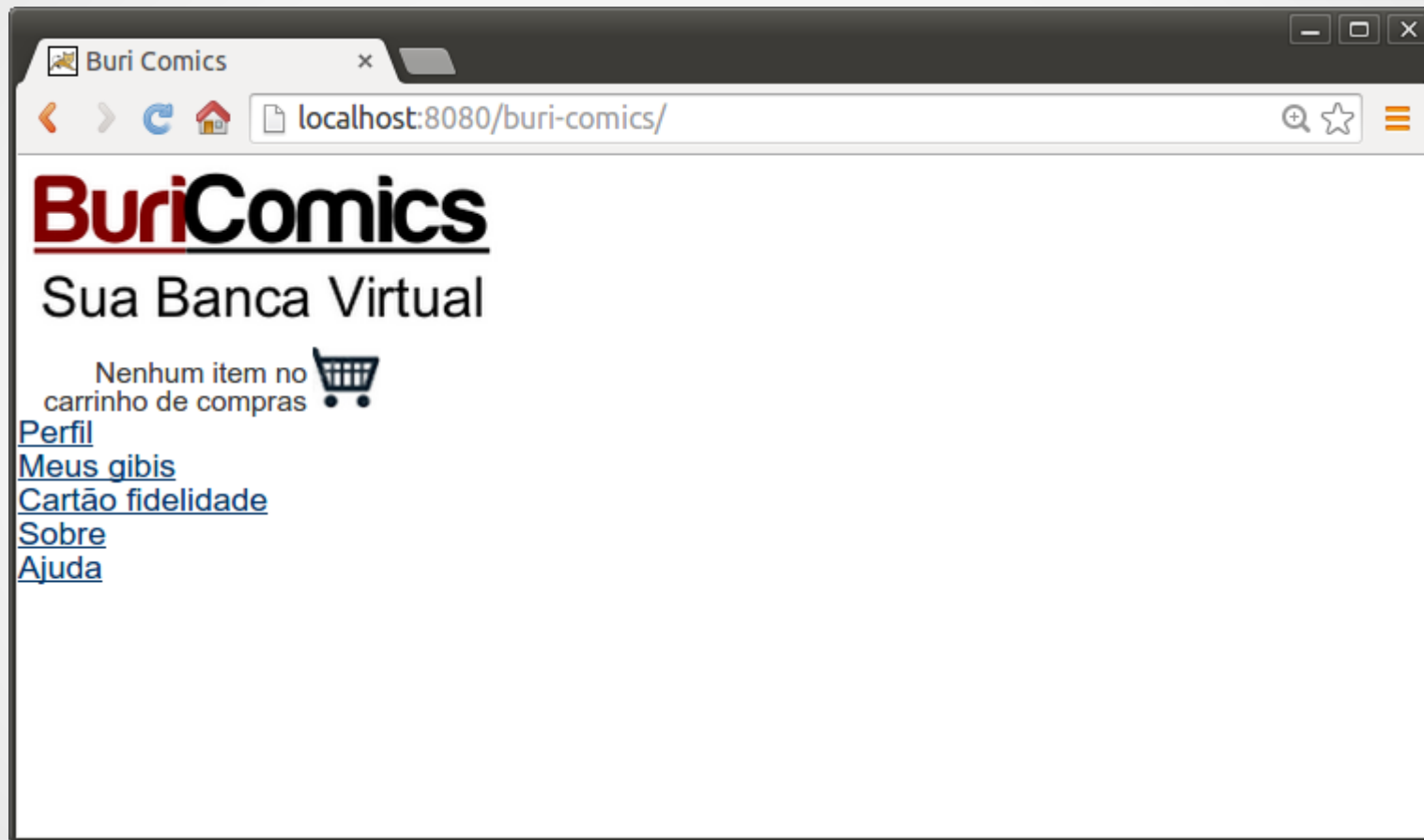
```
ul li{  
    display: inline;  
}
```

# Exercício 03: Reset e Display

- Adicione link para o **reset.css**

```
3 <head>
4 <title>Buri Comics</title>
5 <meta charset="utf-8" />
6 <link rel="stylesheet" href="css/reset.css" />
7 <link rel="stylesheet" href="css/estilo.css" />
8 </head>
9 <body>
10     <header class="container">
11         <h1>
12             
13         </h1>
```

# Resultado do CSS Reset



## Exercício 04: Display

- Vamos usar a propriedade display para gerar o menu horizontal.
- Inclua novas configurações:
  - **Display** e **container**
  - Vamos alinhar o **container**
- Altere a tag Header de index.html

```
16 .menu ul {  
17     font-size: 15|  
18 }  
19  
20 .menu a {  
21     color: #003366;  
22 }  
23  
24 .menu ul li {  
25     display: inline;  
26     margin-left: 20px  
27 }  
28  
29 .container {  
30     margin: 0 auto;  
31     width: 940px;  
32 }
```

# Exercício 04: Display

- Vamos usar a propriedade display para gerar o menu horizontal.
- Inclua novas configurações:
  - **Display** e **container**
  - Vamos alinhar o **container**
- Altere a tag Header de index.html

```
16 .menu ul {  
17     font-size: 15|  
18 }  
19  
20 .menu a {  
21     color: #003366;  
22 }  
23  
24 .menu ul li {  
25     display: inline;  
26     margin-left: 20px  
27 }  
28  
29 .container {  
30     margin: 0 auto;  
    width: 940px;
```

```
<body>  
  <header class="container">  
    <h1>  
        
    </h1>
```



# Resultado esperado

- Agora, os itens estão dispostos na **horizontal**



# Posicionamento de componentes

- A propriedade **position** determina o modo de exibição
- **Static**: o componente permanece no mesmo lugar
- **Relative**: pode ser deslocado, a partir da sua posição original
- **Fixed**: o componente é deslocado para uma coordenada indicada
- **Absolute**: o componente é deslocado para uma coordenada indicada, porém, relativa ao 1º elemento pai

# Exercício 05: Posicionamento

```
.carrinho {  
    background: url("../img/carrinho.png") no-repeat center right;  
    font-size: 14px;  
    padding-right: 35px;  
    text-align: right;  
    width: 140px;
```

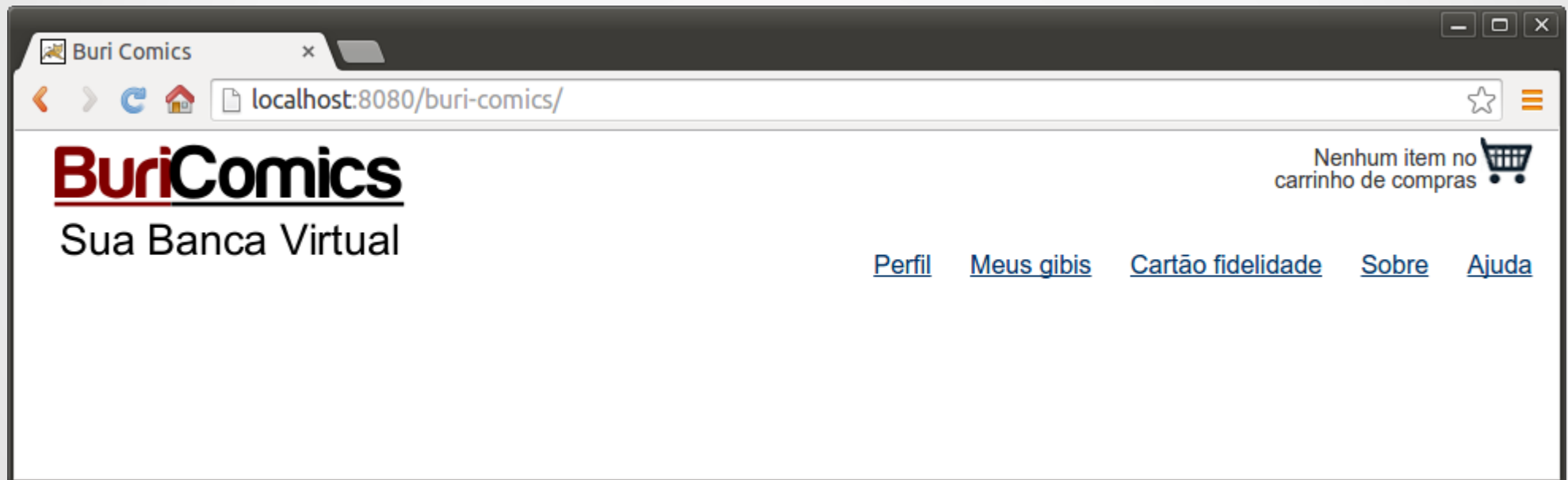
```
    padding-top: 10px;  
    position: absolute;  
    top: 0;  
    right: 0;
```

```
}
```

```
header {  
    position: relative;  
}
```

```
.menu {  
    position: absolute;  
    bottom: 0;  
    right: 0;  
}
```

# Resultado esperado



## 2. HTML Semântico e posicionamento

