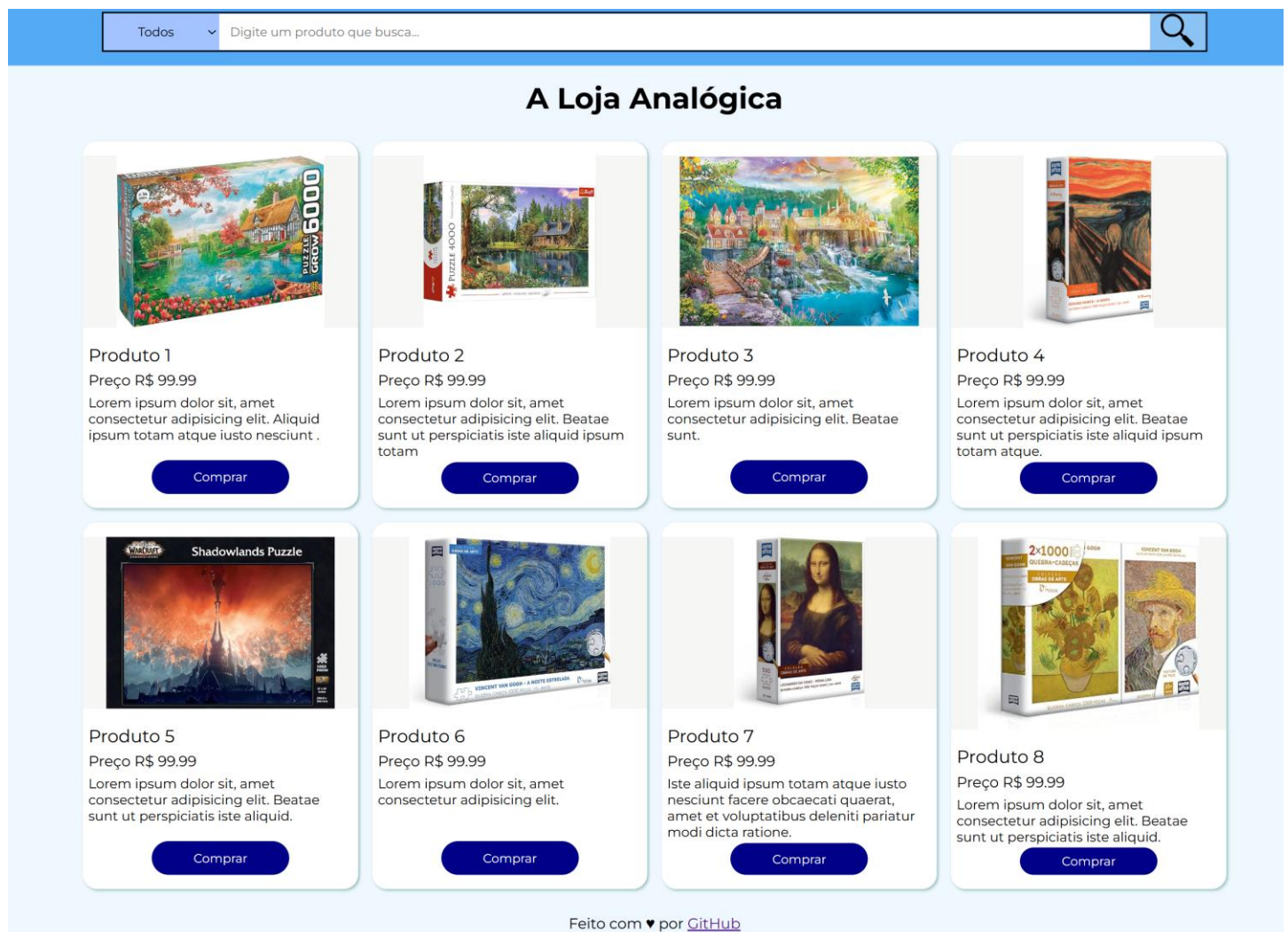




**Instruções Gerais:**

- Atividade Deverá ser Entregue Individualmente.
- Entregar a solução completa em formato ZIP pela tarefa do SIGAA.
- **Horário de Entrega definido na Tarefa do SIGAA**
- Formato de Entrega: Coloque todo arquivo fonte (.html e .css) em um diretório compactado com formato .zip (ZIP) e carregue na tarefa.

Utilizando o que vimos em sala de aula sobre CSS desenvolva uma página web que represente um e-commerce. Utilize o design apresentado na Figura 1 como exemplo.



**Figura 1**

Não é necessário ter as medidas exatas, experimente com as propriedades do CSS para chegar em algo parecido. O objetivo é você treinar as propriedades e ver o resultado. Você deverá também elaborar sua própria estrutura HTML.

No cabeçalho da página temos um formulário que representa uma busca de produtos que pode ser feita nessa loja. Temos uma seleção de categorias, o campo de entrada (input) e um botão “pesquisar”.

Nessa tarefa cada discente deverá implementar um tema de interesse, tais como: Livros, filmes, roupa, eletrônicos, instrumentos musicais etc. No exemplo é apresentada uma loja de quebra-cabeças. As cores e fontes também podem ser adaptadas.

O que se espera é um design que contenha os *cards* devidamente estilizados (como no exemplo), que contenha os seguintes itens: Imagem, nome do produto, descrição, preço e um botão.

Observe o rodapé ao final. Coloque um link para uma página profissional. Caso não possua, crie uma conta no GitHub.

## Utilizando Fontes externas.

Para utilizar fontes externas você precisa informar ao navegador o caminho de onde se encontra a fonte. Você pode buscar fontes em diversas páginas tais como:

[Browse Fonts - Google Fonts](#)

Após encontrar a fonte desejada são necessárias duas informações.

**1** – Na página HTML forneça na área <head> o *link* para a(s) fonte(s).

Exemplo da fonte utilizada na Figura 1:

```
<link rel="preconnect" href="https://fonts.googleapis.com">
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
<link href="https://fonts.googleapis.com/css2?family=Comme:wght@200&family=Montserrat:wght@100;300;500&display=swap"
rel="stylesheet">
```

**2** – Na folha de estilização, informe a fonte utilizada de acordo com o elemento desejado.

Exemplo para o <h1> do exercício:

```
h1{
  ....
  font-family: 'Montserrat', sans-serif;
  ....
}
```

Exemplo para o <label>

```
label{
  .....
  font-family: 'Comme', sans-serif;
  font-style: normal;
}
```

## Mais instruções:

- 1 – Os discentes estão livres para utilizar outra fonte.
- 2 – A temática do exercício **deve** ser modificada. Exemplo: Ao invés de “Loja de Quebra-Cabeças”, pode se construir “Loja de Roupas”, “Loja de Instrumentos Musicais”.
- 3 – Use sua imaginação.
- 4 – Não é permitido utilizar bibliotecas externas para estilização. Utilize CSS “puro” combinado com “grid” e/ou “flex”.

## Dicas de CSS:

1. As propriedades “aspect-ratio” e “object-fit” podem ajudar a manter uma Figura ocupando um espaço padrão. Pesquise sobre estas. Como referência no exemplo fornecido foram utilizados os seguintes valores:
  - a. aspect-ratio: 16/10 (faz a figura utilizar proporção 16/10)
  - b. object-fit: contain (redimensiona a figura para manter a proporção)
2. Elementos flexíveis (dentro de container flex) podem utilizar a propriedade “flex”. Essa propriedade pode ser preenchida com até 3 (três) valores. E é um atalho para as seguintes propriedades
  - a. flex-grow => propriedade relativa dizendo como o espaço extra será distribuído
  - b. flex-shrink => propriedade relativa que dizendo como o espaço extra será removido
  - c. flex-basis => tamanho inicial do elemento flexível.
  - d. No exemplo da Figura 1 temos o seguinte valor na estilização do “card”.  
“flex: 0 300px”. Se apenas dois valores são utilizados a propriedade simboliza “flex-grow” e “flex-basis”.  
Pesquisa mais sobre essas propriedades
3. A propriedade “flex-wrap” pode auxiliar a “empurrar” os elementos para baixo formando novas linhas. No exemplo da Figura 1, foi utilizado na estilização dos container dos cards, o valor “flex-wrap:wrap”