

CSS

Estilizando uma Loja de um Game

Informação Importante

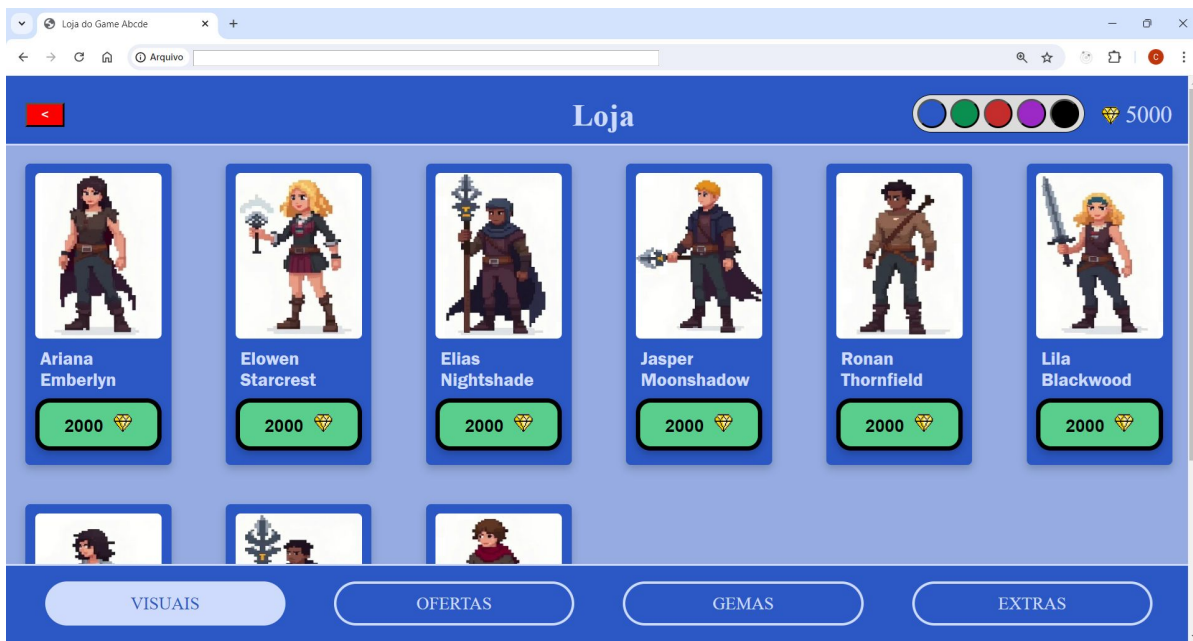
- ▷ Os nomes dos personagens e as imagens apresentadas neste material foram geradas por inteligência artificial.

Objetivo Final



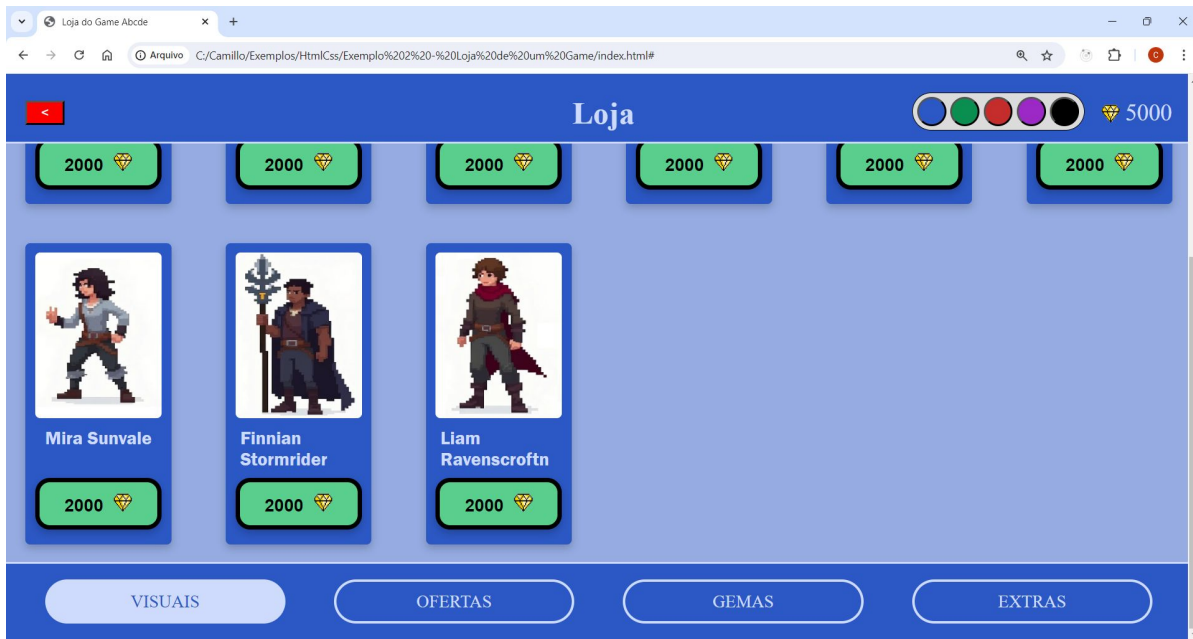
Vamos criar uma loja de games onde os cards que apresentam as skins que podem ser compradas são apresentados em acordo com o espaço disponível, criando linhas novas sempre que necessário.

Objetivo Final



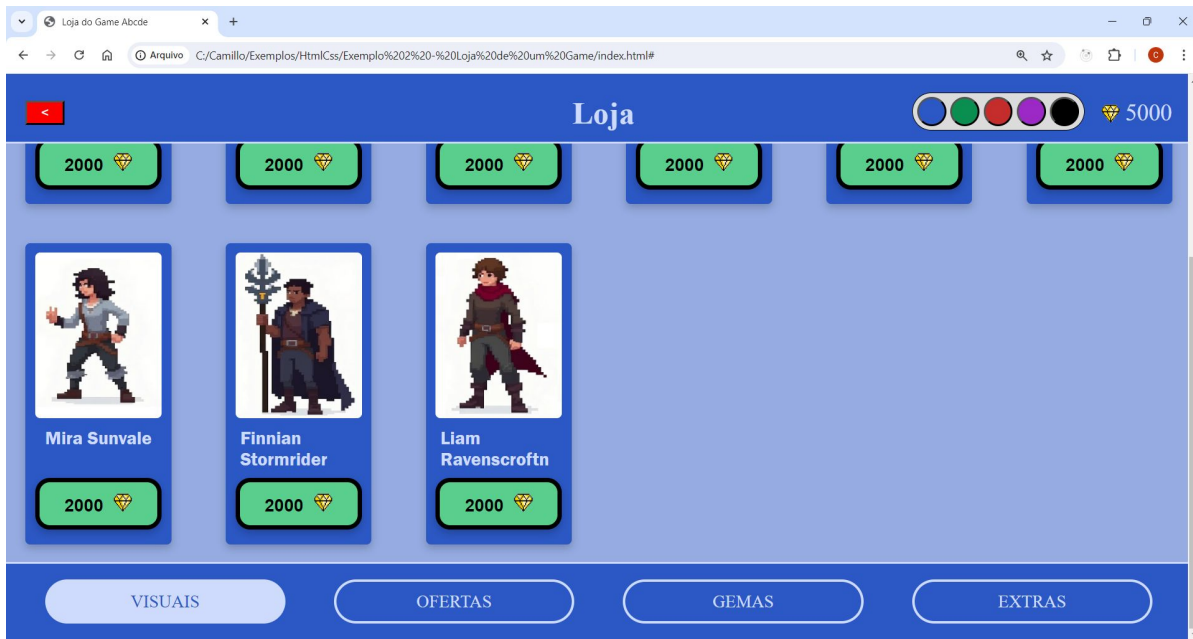
O cabeçalho e o rodapé da loja devem estar sempre visíveis.

Objetivo Final



A rolagem deve ocorrer somente na lista de skins.

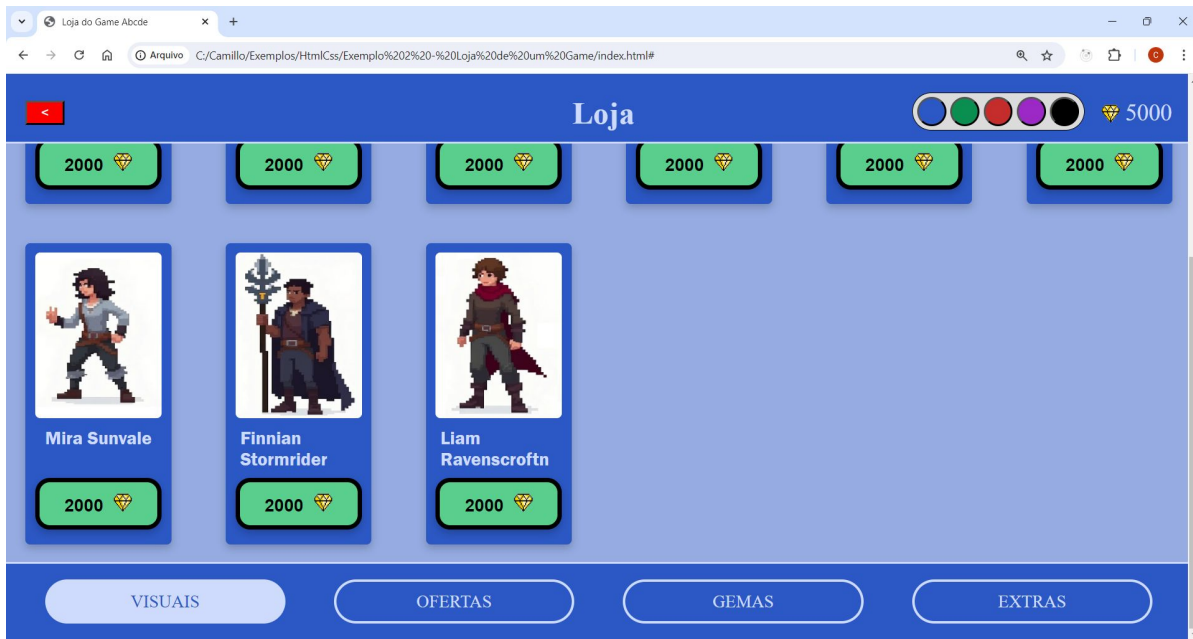
Objetivo Final



Os botões do rodapé devem trocar de cor quando o mouse estiver em cima dos mesmos.

O botão “Visuais” deve estar destacado, indicando que o mesmo está selecionado.

Objetivo Final



No cabeçalho, deve ser criado botões coloridos, que permitem a troca do tema (cores) utilizado.

HTML Inicial

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="./css/loja.css">

  <title>Loja do Game Abcde</title>
</head>
```


HTML Inicial

```
<body>
  <header>
    <button class="voltar">&lt;</button>
    <h1>Loja</h1>
    <div class="carteira">
      
      <p>5000</p>
    </div>
    <div class="tema-area">
      <button id="trocar-tema-azul" class="tema tema-azul"></button>
      <button id="trocar-tema-verde" class="tema tema-verde"></button>
      <button id="trocar-tema-vermelho" class="tema tema-vermelho"></button>
      <button id="trocar-tema-roxo" class="tema tema-roxo"></button>
      <button id="trocar-tema-preto" class="tema tema-preto"></button>
    </div>
  </header>
```

HTML Inicial

```
<main>
  <div class="loja">
    <div class="card">
      
      <div class="container">
        <p>Nome Personagem A</p>
        <button class="botao-comprar">2000
          
        </button>
      </div>
    </div>
    <!-- ... Repita a div com class="card" várias vezes,
    trocando o nome e a imagem -->
  </div>
</main>
```

HTML Inicial

```
<footer>
  <nav class="menu">
    <ul>
      <li class="menu-item-seleccionado"><a href="#">VISUAIS</a></li>
      <li><a href="#">OFERTAS</a></li>
      <li><a href="#">GEMAS</a></li>
      <li><a href="#">EXTRAS</a></li>
    </ul>
  </nav>
</footer>
</body>

</html>
```

Aplicando CSS

- ▷ Nos próximos slides vamos aplicar gradualmente o CSS para estilizarmos a nossa página.
 - O CSS será apresentado à esquerda.
 - O resultado da aplicação do CSS será apresentado à direita.
- ▷ Sugestão: insira as regras uma a uma e, a cada regra, verifique o seu efeito.

Aplicando CSS

- ▷ Obtenha com o professor a estrutura inicial do projeto, contendo o HTML, as imagens e os arquivos CSS, sendo que estes últimos devem estar vazios.
- ▷ Vamos começar importando os outros arquivos CSS no arquivo *loja.css*, que é o único arquivo CSS importado em nosso HTML.

```
@import "../padrao.css";  
@import "../cabecalho.css";  
@import "../rodape.css";
```

Aplicando CSS - padrao.css

```
:root {  
  --destaque: rgb(44, 88, 197);  
  --destaque-compl: rgb(206, 219, 253);  
  --fundo: rgba(44, 88, 197, 0.5);  
  --fundo-compl: rgba(11, 36, 98, 0.5);  
  --comprar: rgb(90, 207, 141);  
  --comprar-compl: rgb(0, 0, 0);  
  --comprar-destaque: rgb(0, 255, 110);  
}
```

- Vamos iniciar o arquivo **padrao.css** declarando algumas **variáveis**.
- O seletor **:root** indica que estamos selecionando todo o HTML.
- Uma variável possui o seguinte padrão de nome:
--nome: valor
 - **nome**: nome da variável
 - **valor**: valor da variável

Aplicando CSS - padrao.css

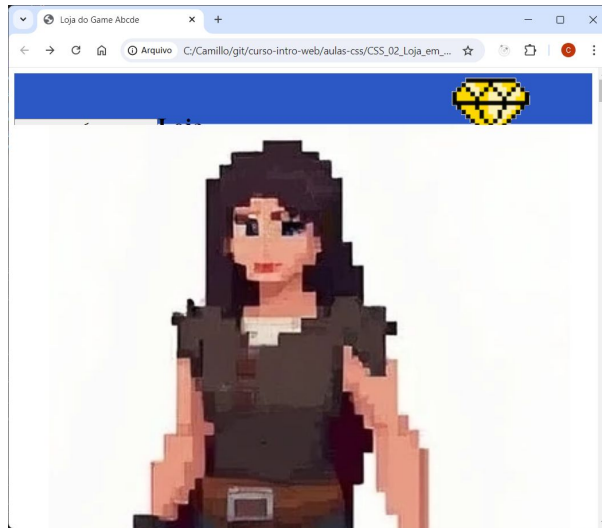
```
:root {  
  --destaque: rgb(44, 88, 197);  
  --destaque-compl: rgb(206, 219, 253);  
  --fundo: rgba(44, 88, 197, 0.5);  
  --fundo-compl: rgba(11, 36, 98, 0.5);  
  --comprar: rgb(90, 207, 141);  
  --comprar-compl: rgb(0, 0, 0);  
  --comprar-destaque: rgb(0, 255, 110);  
}
```

- As variáveis que declaramos aqui serão utilizadas em outros arquivos CSS para aplicar cores a elementos.

Aplicando CSS - cabecalho.css

```
@import "../padrao.css";

header {
  background-color: var(--destaque);
  color: var(--destaque-texto);
  height: 70px;
  display: grid;
  grid-template-columns: 200px auto 200px;
  align-items: center;
  border-bottom: solid 2px var(--destaque-compl);
}
```



Aplicando CSS - cabecalho.css

```
@import "../padrao.css";

header {
  background-color: var(--destaque);
  color: var(--destaque-texto);
  height: 70px;
  display: grid;
  grid-template-columns: 200px auto 200px;
  align-items: center;
  border-bottom: solid 2px var(--destaque-compl);
}
```

- Cor de fundo será a que indicamos na variável --destaque (arquivo padrao.css)
- A cor da fonte é a indicada pela variável --destaque-texto.



Aplicando CSS - cabecalho.css

```
@import "../padrao.css";

header {
  background-color: var(--destaque);
  color: var(--destaque-texto);
  height: 70px;
  display: grid;
  grid-template-columns: 200px auto 200px;
  align-items: center;
  border-bottom: solid 2px var(--destaque-comp)
}
```



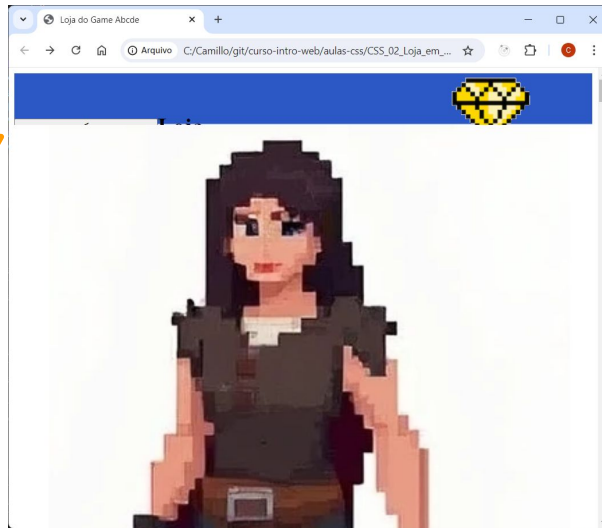
- Alteramos a altura do header para 70px, indicamos que terá display: grid, indicamos que teremos 3 colunas com a do meio ocupando todo o espaço entre as outras duas.

Aplicando CSS - cabecalho.css

```
@import "../padrao
```

```
header {  
  background-color: var(--c  
  height: 70px;  
  display: grid;  
  grid-template-  
  align-items:  
  border-bottom
```

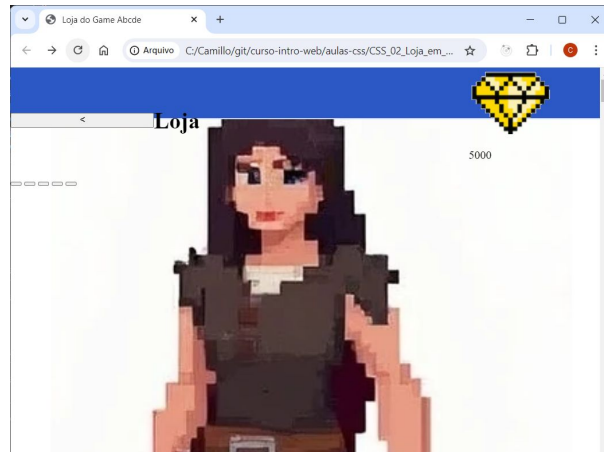
- Ainda temos problemas, pois o header não está encostando nas bordas da viewmodel, além de ser movido quando utilizamos a rolagem.



Aplicando CSS - cabecalho.css

```
@import "../padrao.css";

header {
  background-color: var(--destaque);
  color: var(--destaque-texto);
  height: 70px;
  display: grid;
  grid-template-columns: 200px auto 200px;
  align-items: center;
  border-bottom: solid 2px var(--destaque-compl);
  position: fixed;
  top: 0;
  left: 0;
  width: 100vw;
}
```



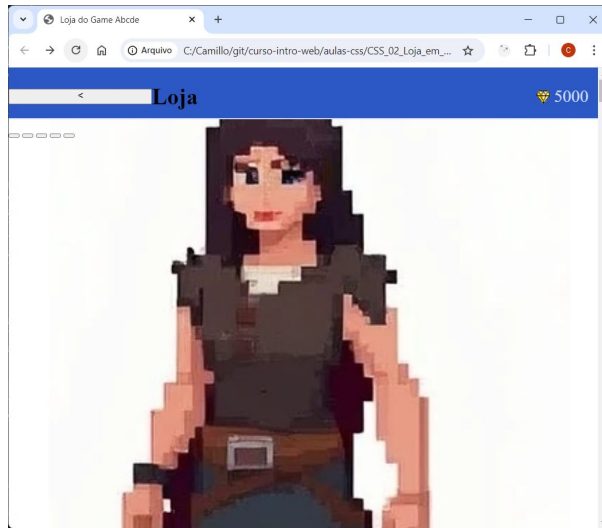
- Alteramos a posição do header para fixed, com distâncias para o topo e para a esquerda iguais a zero e largura total da viewport.

Aplicando CSS - cabecalho.css

```
header .carteira {
  display: grid;
  grid-template-columns: 30px auto;
  align-items: center;
  justify-self: right;
  justify-content: right;
  margin-right: 30px;
  width: 100px;
}

header>.carteira img {
  height: 20px;
  width: 20px;
  margin: 5px;
}

header>.carteira p {
  font-size: 1.5em;
  color: var(--destaque-compl);
}
```

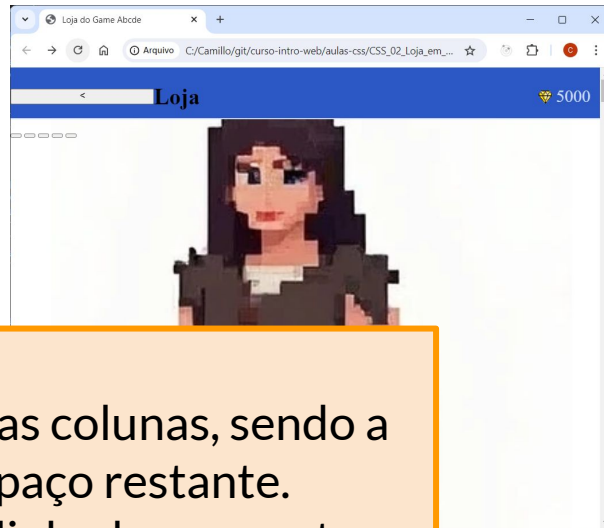


Aplicando CSS - cabecalho.css

```
header .carteira {  
  display: grid;  
  grid-template-columns: 30px auto;  
  align-items: center;  
  justify-self: right;  
  justify-content: right;  
  margin-right: 30px;  
  width: 100px;  
}
```

```
header>.carteira img {  
  height: 20px;  
  width: 20px;  
  margin: 5px;  
}  
  
header>.carteira p {  
  font-size: 1.5em;  
  color: var(--destaqu  
}
```

- A carteira será um grid com duas colunas, sendo a segunda coluna ocupando o espaço restante.
- Os itens serão verticalmente alinhados ao centro.
- A carteira e o conteúdo da mesma serão alinhados à direita.

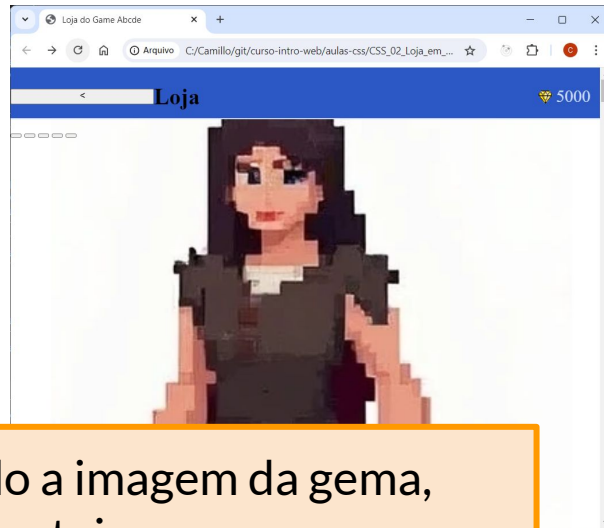


Aplicando CSS - cabecalho.css

```
header .carteira {  
  display: grid;  
  grid-template-columns: 30px auto;  
  align-items: center;  
  justify-self: right;  
  justify-content: right;  
  margin-right: 30px;  
  width: 100px;  
}
```

```
header>.carteira img {  
  height: 20px;  
  width: 20px;  
  margin: 5px;  
}
```

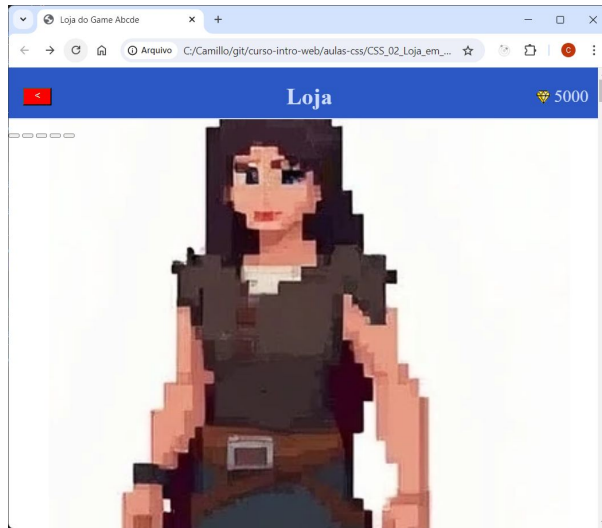
```
header>.carteira p {  
  font-size: 1.5em;  
  color: var(--destaque-compl);  
}
```



- Redimensionando a imagem da gema, apresentada na carteira.
- Alterando a cor e o tamanho da fonte do texto que apresenta o valor em carteira.

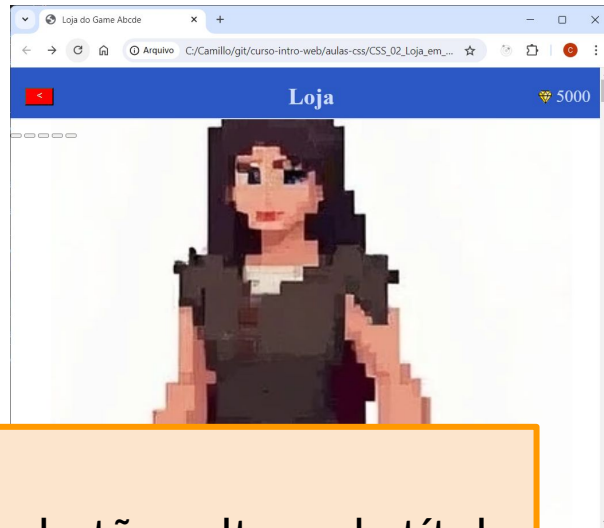
Aplicando CSS - cabecalho.css

```
header .voltar {  
  height: 25px;  
  width: 40px;  
  align-self: center;  
  position: relative;  
  left: 20px;  
  color: white;  
  background-color: red;  
}  
  
header>h1 {  
  font-weight: bolder;  
  align-self: center;  
  justify-self: center;  
  color: var(--destaque-compl);  
}
```



Aplicando CSS - cabecalho.css

```
header .voltar {  
  height: 25px;  
  width: 40px;  
  align-self: center;  
  position: relative;  
  left: 20px;  
  color: white;  
  background-color: red;  
}  
  
header>h1 {  
  font-weight: bolder;  
  align-self: center;  
  justify-self: center;  
  color: var(--destaque-compl);  
}
```



- Estilização do botão voltar e do título principal da loja.

Aplicando CSS - cabecalho.css

- ▷ Agora vamos estilizar o painel que permite a troca do tema da página. Para isso, vamos alterar o arquivo padrao.css:

Aplicando CSS - padrao.css

```
:root {  
  --destaque-azul: rgb(44, 88, 197);  
  --destaque-verde: rgb(10, 143, 81);  
  --destaque-vermelho: rgb(197, 44, 44);  
  --destaque-roxo: rgb(156, 41, 198);  
  --destaque-preto: rgb(0, 0, 0);  
}
```

```
:root body.azul {  
  --destaque: var(--destaque-azul);  
  --destaque-compl: rgb(206, 219, 253);  
  --fundo: rgba(44, 88, 197, 0.5);  
  --fundo-compl: rgba(11, 36, 98, 0.5);  
  --comprar: rgb(90, 207, 141);  
  --comprar-compl: rgb(0, 0, 0);  
  --comprar-destaque: rgb(0, 255, 110);  
}
```

Aplicando CSS - padrao.css

```
:root {  
  --destaque-azul: rgb(44, 88, 197);  
  --destaque-verde: rgb(10, 143, 81);  
  --destaque-vermelho: rgb(197, 44, 44);  
  --destaque-roxo: rgb(156, 41, 198);  
  --destaque-preto: rgb(0, 0, 0);  
}
```

```
:root body.azul {  
  --destaque: var(--destaque-azul);  
  --destaque-compl: rgb(206, 219, 253);  
  --fundo: rgba(44, 88, 197, 0.5);  
  --fundo-compl: rgba(11, 36, 98, 0.5);  
  --comprar: rgb(90, 207, 141);  
  --comprar-compl: rgb(0, 0, 0);  
  --comprar-destaque: rgb(0, 255, 110);  
}
```

- Agora temos variáveis globais apenas para indicar a cor do tema.

Aplicando CSS - padrao.css

```
:root {  
  --destaque-azul: rgb(44, 88, 197);  
  --destaque-verde: rgb(10, 143, 81);  
  --destaque-vermelho: rgb(197, 44, 44);  
  --destaque-roxo: rgb(156, 41, 198);  
  --destaque-preto: rgb(0, 0, 0);  
}
```

```
:root body.azul {  
  --destaque: var(--destaque-azul);  
  --destaque-compl: var(--destaque-verde);  
  --fundo: rgba(44, 88, 197, 0.5);  
  --fundo-compl: var(--destaque-vermelho);  
  --comprar: rgb(90, 207, 141);  
  --comprar-compl: rgb(0, 0, 0);  
  --comprar-destaque: rgb(0, 255, 110);  
}
```

- Seletor para variáveis existentes apenas para o elemento body com classe azul (`<body class="azul">`) em nosso HTML.

Aplicando CSS - padrao.css

```
:root {  
  --destaque-azul: rgb(44, 88, 197);  
  --destaque-verde: rgb(10, 143, 81);  
  --destaque-vermelho: rgb(197, 44, 44);  
  --destaque-roxo: rgb(156, 41, 198);  
  --destaque-preto: rgb(0, 0, 0);  
}
```

```
:root body.azul {  
  --destaque: var(--destaque-azul);  
  --destaque-compl: rgb(206, 219, 253);  
  --fundo: rgba(44, 88, 197, 0.5);  
  --fundo-compl: rgba(11, 36, 98, 0.5);  
  --comprar: rgb(90, 207, 141);  
  --comprar-compl: rgb(0, 0, 0);  
  --comprar-destaque: rgb(0, 255, 110);  
}
```

- A cor --destaque passou a ter o valor armazenado na variável --destaque-azul.

Aplicando CSS - padrao.css

- ▷ Nos próximos slides seguirão as definições das variáveis para quando alterarmos a classe do body para as outras cores.
- ▷ Por enquanto, a troca da classe precisa ser feita manualmente.
 - No final deste material, apenas para ajudar o aluno com sua curiosidade, apresentaremos o Javascript que troca o tema quando se clica no botão correspondente.

Aplicando CSS - padrao.css

```
:root body.verde {  
  --destaque: var(--destaque-verde);  
  --destaque-compl: rgb(170, 234, 194);  
  --fundo: rgba(115, 224, 150, 0.5);  
  --fundo-compl: rgba(4, 48, 24, 0.5);  
  --comprar: rgb(90, 207, 141);  
  --comprar-compl: rgb(0, 0, 0);  
  --comprar-destaque: rgb(0, 255, 110);  
}
```

```
:root body.vermelho {  
  --destaque: var(--destaque-vermelho);  
  --destaque-compl: rgb(253, 206, 206);  
  --fundo: rgba(197, 44, 44, 0.5);  
  --fundo-compl: rgba(98, 11, 11, 0.5);  
  --comprar: rgb(90, 207, 141);  
  --comprar-compl: rgb(0, 0, 0);  
  --comprar-destaque: rgb(0, 255, 110);  
}
```

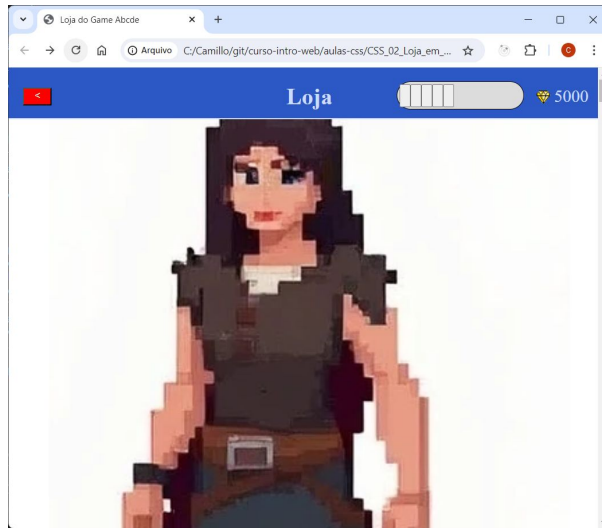

Aplicando CSS - padrao.css

```
:root body.roxo {
  --destaque: var(--destaque-roxo);
  --destaque-compl: rgb(221, 186, 234);
  --fundo: rgb(202, 151, 220);
  --fundo-compl: rgb(108, 47, 130);
  --comprar: rgb(90, 207, 141);
  --comprar-compl: rgb(80, 19, 71);
  --comprar-destaque: rgb(0, 255, 110);
}

:root body.preto {
  --destaque: var(--destaque-preto);
  --destaque-compl: rgb(207, 213, 231);
  --fundo: rgba(36, 52, 92, 0.5);
  --fundo-compl: rgba(211, 215, 225, 0.5);
  --comprar: rgb(90, 207, 141);
  --comprar-compl: rgb(0, 0, 0);
  --comprar-destaque: rgb(0, 255, 110);
}
```

Aplicando CSS - cabecalho.css

```
header .tema-area {  
  display: flex;  
  border: 1px solid black;  
  background: gainsboro;  
  border-radius: 2rem;  
  width: 170px;  
  height: 32px;  
  padding: 3px;  
  position: absolute;  
  top: 19px;  
  right: 120px;  
}
```



Aplicando CSS - cabecalho.css

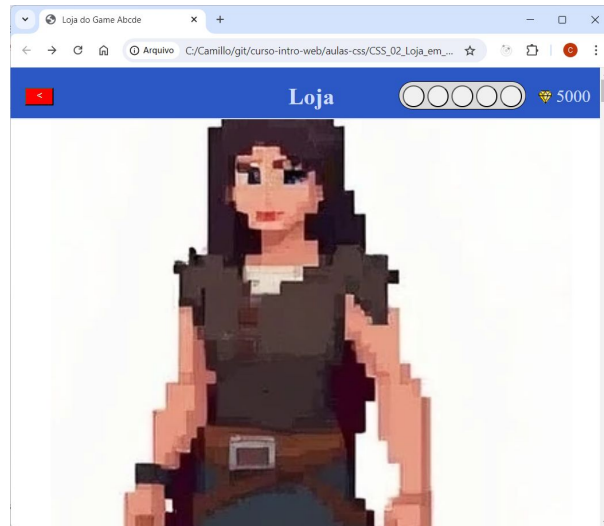
```
header .tema-area {  
  display: flex;  
  border: 1px solid  
  background: gainsb  
  border-radius: 2re  
  width: 170px;  
  height: 32px;  
  padding: 3px;  
  position: absolute  
  top: 19px;  
  right: 120px;  
}
```

- Vamos usar flexbox na área onde trocaremos o tema.
- Note que os botões apareceram um ao lado do outro.



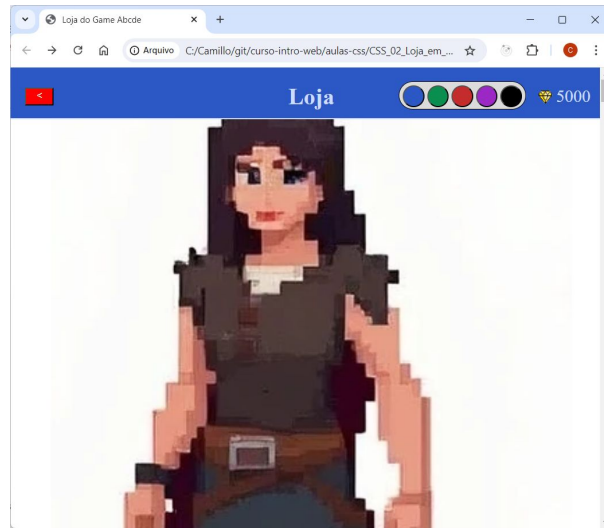
Aplicando CSS - cabecalho.css

```
header .tema {  
  position: relative;  
  border-radius: 50%;  
  width: 30px;  
  height: 30px;  
  left: 0px;  
  margin: 2px;  
}
```



Aplicando CSS - cabecalho.css

```
header .tema-azul {  
    background: var(--destaque-azul);  
}  
  
header .tema-vermelho {  
    background: var(--destaque-vermelho);  
}  
  
header .tema-verde {  
    background: var(--destaque-verde);  
}  
  
header .tema-roxo {  
    background: var(--destaque-roxo);  
}  
  
header .tema-preto {  
    background: var(--destaque-preto);  
}
```

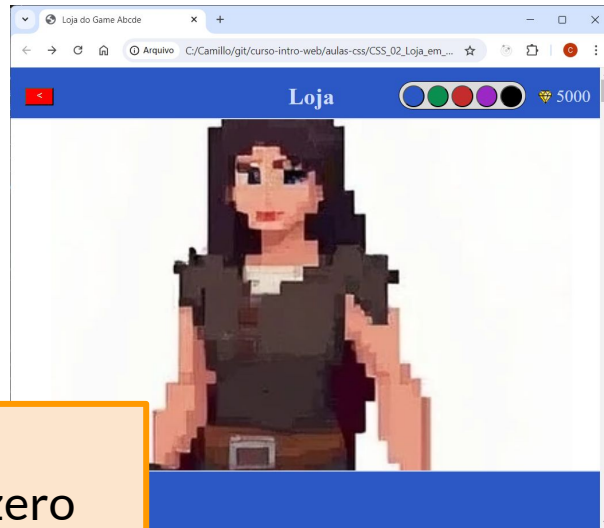


Aplicando CSS - rodape.css

```
@import "../padrao.css";

footer {
  background-color: var(--destaque);
  color: var(--destaque-compl);
  height: 80px;
  align-items: center;
  justify-content: center;
  position: fixed;
  bottom: 0;
  left: 0;
  width: 100vw;
  border-top: solid 2px;
}
```

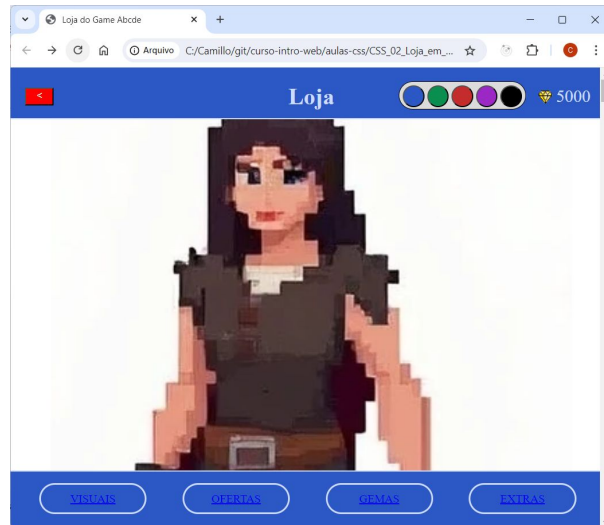
- Rodapé fixo com distância zero para baixo e para a esquerda e com largura total da viewport.



Aplicando CSS - rodape.css

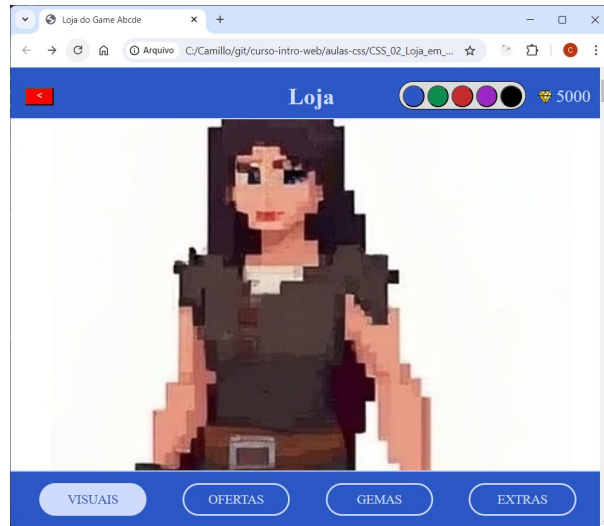
```
footer .menu ul {  
  display: grid;  
  grid-template-columns: 1fr 1fr 1fr 1fr;  
  justify-content: center;  
}
```

```
footer .menu li {  
  display: inline;  
  padding: 10px;  
  border: solid 3px;  
  border-radius: 25px;  
  border-color: var(--destaque-compl);  
  margin-right: 50px;  
  justify-content: center;  
  display: flex;  
}
```



Aplicando CSS - rodape.css

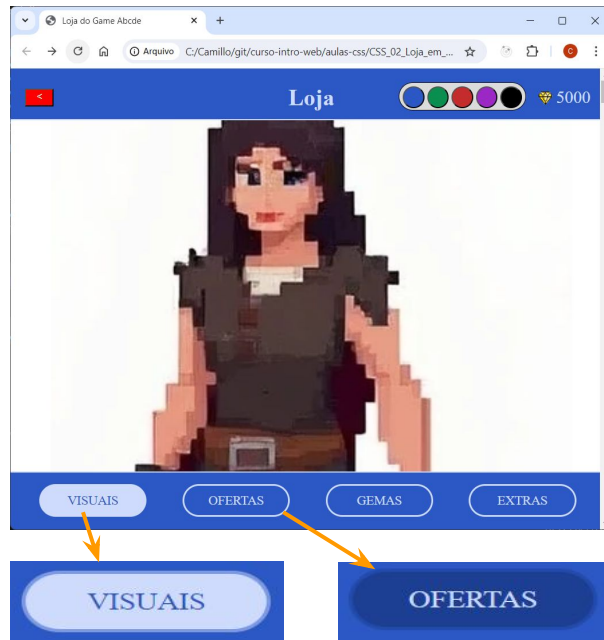
```
footer .menu li a {  
    font-size: large;  
    color: var(--destaque-compl);  
    text-decoration: none;  
}  
  
footer .menu .menu-item-selecionado {  
    background-color: var(--destaque-compl);  
}  
  
footer .menu .menu-item-selecionado a {  
    color: var(--destaque);  
}
```



Aplicando CSS - rodape.css

```
footer .menu li:hover {  
  border-color: var(--fundo);  
  background-color: var(--fundo-compl);  
}  
  
footer .menu .menu-item-selecionado:hover {  
  background-color: var(--destaque-compl);  
}
```

- Regras aplicadas enquanto o ponteiro do mouse estiver em cima do item de menu.

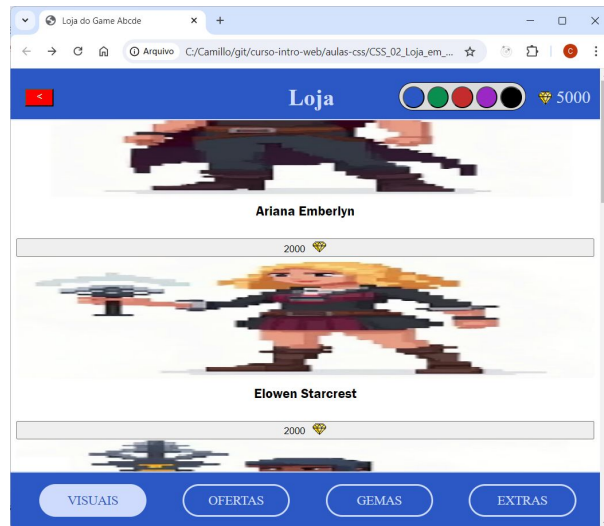


Aplicando CSS - loja.css

```
@import "../padrao.css";
@import "../cabecalho.css";
@import "../rodape.css";

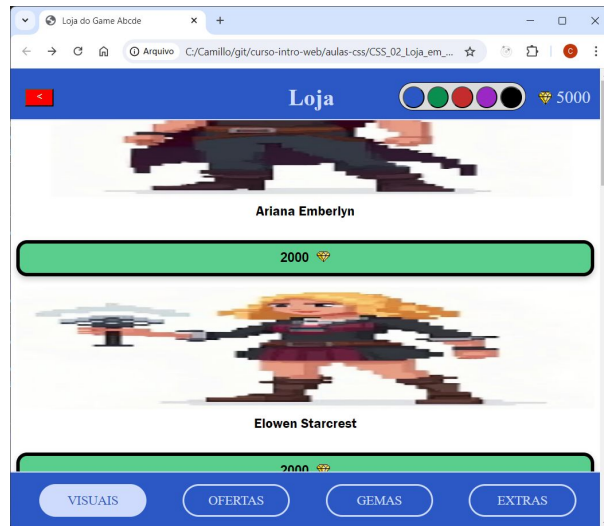
.card>img {
  border-radius: 5px;
  height: 170px;
  justify-self: center;
}

.botao-comprar>img {
  width: 20px;
  padding-left: 10px;
}
```



Aplicando CSS - loja.css

```
.botao-comprar {  
  background-color: var(--comprar);  
  border: solid 5px;  
  font-size: large;  
  font-weight: bold;  
  padding: 10px;  
  box-shadow: 0 4px 8px 0 rgba(0, 0, 0, 0.2);  
  border-radius: 15px;  
  margin: 2px 0 15px 0;  
}  
  
.botao-comprar:hover {  
  background-color: var(--comprar-destaque);  
}
```

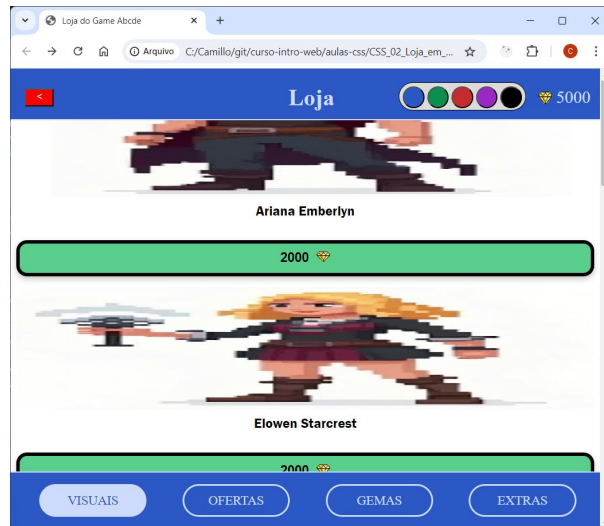


Aplicando CSS - loja.css

```
.botao-comprar {  
  background-color: var(--comprar);  
  border: solid 5px;  
  font-size: large;  
  font-weight: bold;  
  padding: 10px;  
  box-shadow: 0 4px 8px 0 rgba(0, 0, 0, 0.2);  
  border-radius: 15px;  
  margin: 2px 0 15px
```

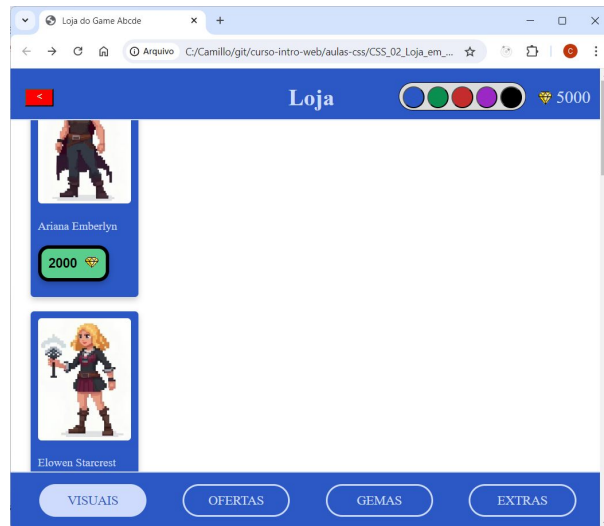
- Sombra no botão.

```
.botao-comprar:hover {  
  background-color: var(--comprar-destaque);  
}
```



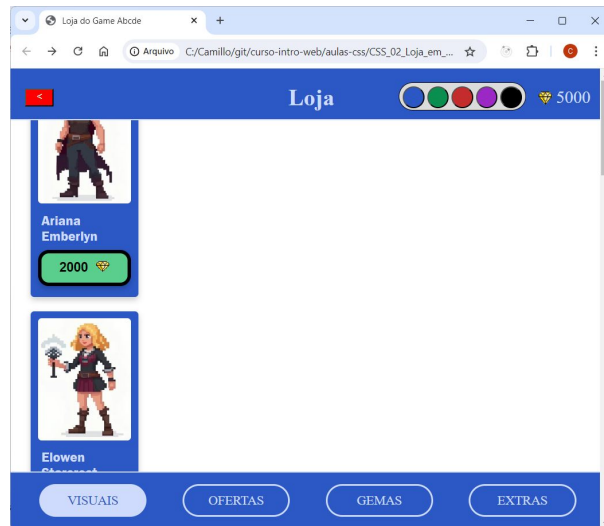
Aplicando CSS - loja.css

```
.card {  
  box-shadow: 0 4px 8px 0 rgba(0, 0, 0, 0.2);  
  border-radius: 5px;  
  width: 130px;  
  height: 300px;  
  color: var(--destaque-compl);  
  padding: 10px 10px 0px 10px;  
  margin: 0 0 20px 20px;  
  background-color: var(--destaque);  
  display: grid;  
  grid-template-columns: auto;  
}  
  
.card>h3 {  
  margin: 0;  
}
```



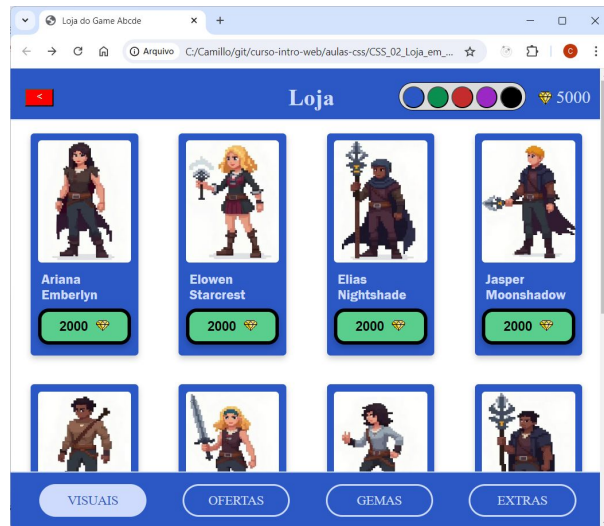
Aplicando CSS - loja.css

```
.card>.container {  
  display: flex;  
  flex-direction: column;  
  margin: 0px;  
  align-self: end;  
}  
  
.card>.container>p {  
  font-family: 'Franklin Gothic Medium', 'Arial  
Narrow', Arial, sans-serif;  
  align-self: center;  
  font-size: 1.2em;  
  margin: 5px;  
  align-self: top;  
  height: 45px;  
}
```



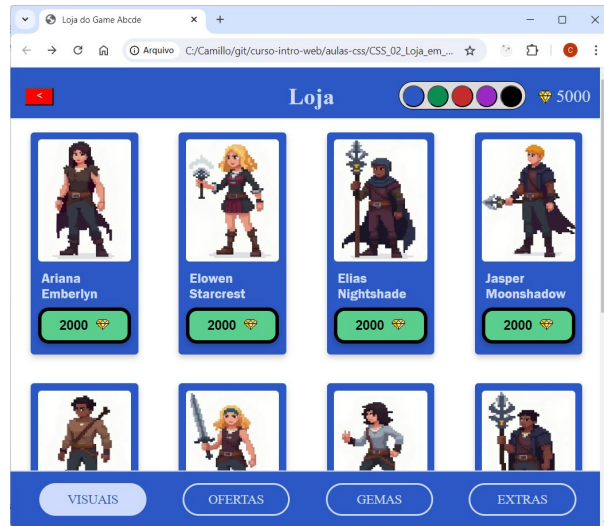
Aplicando CSS - loja.css

```
main {  
  font-family: 'Franklin Gothic Medium', 'Arial  
Narrow', Arial, sans-serif;  
  padding-bottom: 80px;  
}  
  
.loja {  
  display: grid;  
  grid-template-columns: repeat(auto-fit,  
minmax(180px, 1fr));  
  gap: 20px;  
  margin-top: 90px;  
}
```



Aplicando CSS - loja.css

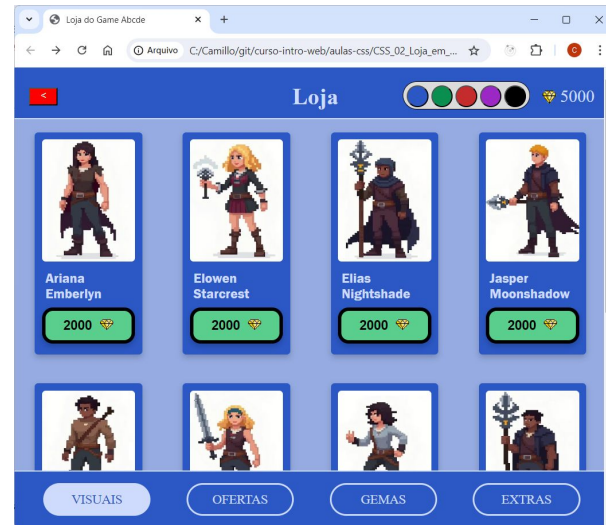
```
main {  
  font-family: 'Franklin Gothic Medium', 'Arial  
Narrow', Arial, sans-serif;  
  padding-bottom: 80px;  
}  
  
.loja {  
  display: grid;  
  grid-template-columns: repeat(auto-fit,  
minmax(180px, 1fr));  
  gap: 20px;  
  margin-top: 90px;  
}
```



- Cada coluna terá no mínimo 180px. Sempre que uma nova coluna com no mínimo 180px puder ser criada, o será.

Aplicando CSS - loja.css

```
body {  
    background-color: var(--fundo);  
}
```



Aplicando JS - `./js/troca-tema.js`

- ▷ Crie a pasta `js` e, dentro da mesma, crie o arquivo `troca-tema.js`.
 - O código deste arquivo se encontra no próximo slide.

Aplicando JS - ./js/troca-tema.js

```
function trocaTema(tema) {  
    document.body.classList.remove('azul', 'vermelho', 'verde', 'roxo', 'preto');  
    document.body.classList.add(tema)  
}  
  
function registrarTrocadorTema(tema) {  
    const elementoTema = document.getElementById(`trocar-tema-${tema}`);  
  
    if (elementoTema) {  
        elementoTema.addEventListener('click', () => trocaTema(tema));  
    }  
}  
  
registrarTrocadorTema('azul');  
registrarTrocadorTema('vermelho');  
registrarTrocadorTema('verde');  
registrarTrocadorTema('roxo');  
registrarTrocadorTema('preto');
```

Fim de material

Dúvidas?