**ESCOLA ESTADUAL SELIM JOSÉ DE SALLES**

ICARO MIGUEL NERES

ÍTALO HIAGO FERREIRA DIAS

KAYCK RODRIGUES CAMPOS

MARIA EDUARDA EGUES SILVA

**YOURSELF**

**IPATINGA-MG**

**2024**

ICARO MIGUEL NERES

ÍTALO HIAGO FERREIRA DIAS

KAYCK RODRIGUES CAMPOS

MARIA EDUARDA EGUES SILVA

**YOURSELF**

**Trabalho Avaliativo, apresentado ao Técnico de Análise de Desenvolvimento de Sistemas como parte dos requisitos necessários para a obtenção da nota.**

**Professor(a): Marley Andrade**

**Disciplina: DESENVOLVIMENTO BACK-END & DESENVOLVIMENTO FRONT-END**

**Turma: 302**

**IPATINGA - MG**

**2024**

**RESUMO**

TEMA e Objetivo: Este estudo foi realizado no contexto da saúde emocional e do uso da tecnologia para apoio e acolhimento de usuários. O objetivo geral é explorar como a plataforma "Yourself" auxilia o usuário a entender e gerenciar suas emoções, oferecendo uma interface personalizada e adaptativa que responde aos estados emocionais.

Metodologia: O desenvolvimento da plataforma Yourself utilizou tecnologias web, como HTML, CSS e JavaScript, no Visual Studio Code. A coleta de dados incluiu testes com usuários para avaliar a eficácia do site em identificar e reagir aos estados emocionais. A análise focou na resposta dos usuários ao conteúdo sugerido, de acordo com as emoções relatadas.

Resultados: Os principais resultados indicam que a plataforma é capaz de adaptar sugestões de conteúdo, como filmes, séries, animes, livros e músicas, de maneira personalizada, proporcionando uma experiência de acolhimento emocional para estados como tristeza, felicidade e tédio. A estrutura interativa e as recomendações dinâmicas se mostraram eficazes em promover a regulação emocional dos usuários.

Contribuição: Este estudo contribui para o desenvolvimento de interfaces digitais com foco na saúde mental, demonstrando o potencial da tecnologia em criar ambientes de suporte emocional personalizados e acessíveis.

**PALAVRAS-CHAVE**:Saúde emocional, Interface adaptativa, Tecnologia interativa, Plataforma digital, Acolhimento emocional.

**SUMÁRIO**

[**1. INTRODUÇÃO 5**](#_gjdgxs)

[**2. DESENVOLVIMENTO 7**](#_30j0zll)

[2.1 Coleta de Dados Emocionais 7](#_1fob9te)

[2.2 Análise de Dados Emocionais 7](#_3znysh7)

[2.3 Modelagem de Interação e Algoritmos de Recomendação 7](#_2et92p0)

[2.4 Programação e Infraestrutura de Suporte 8](#_tyjcwt)

[2.6 Desenvolvimento do Personagem Tef 55](#_3dy6vkm)

[2.7 Personalidade e Função no Site 55](#_1t3h5sf)

[2.8 Tef como Elemento Motivador e Curador 55](#_4d34og8)

[2.9 A Estética e o Simbolismo de Tef 56](#_2s8eyo1)

[2.10 Impacto na Experiência do Usuário 56](#_17dp8vu)

[**3. CONCLUSÃO 57**](#_3rdcrjn)

[**4. REFERÊNCIA 58**](#_26in1rg)

# 1. INTRODUÇÃO

No contexto atual, marcado pela crescente digitalização e individualização das interações, a saúde emocional das pessoas se tornou uma preocupação relevante. Estudos indicam que a tecnologia tem potencial não apenas para conectar as pessoas, mas também para oferecer suporte emocional, auxiliando os usuários a compreenderem melhor seus sentimentos e a gerenciarem suas emoções de maneira mais eficaz. Contudo, muitas plataformas digitais carecem de recursos interativos personalizados que se ajustem de forma sensível ao estado emocional dos usuários, oferecendo alternativas de apoio e entretenimento adequadas a cada situação. Esta lacuna aponta a necessidade de um estudo que explore interfaces digitais capazes de identificar e responder a diferentes estados emocionais.

O presente estudo busca investigar como o site "Yourself" pode contribuir para o bem-estar emocional dos usuários ao adaptar suas interações e recomendações de conteúdo com base no humor relatado. A plataforma Yourself apresenta um personagem que, sensível ao estado emocional do usuário, proporciona um espaço virtual onde, independentemente de estar feliz, triste ou entediado, o usuário recebe sugestões personalizadas de séries, filmes, animes, livros e músicas. Este modelo oferece uma experiência alternativa ao consumo de conteúdo, visando não apenas o entretenimento, mas também o acolhimento e o apoio emocional.

A relevância deste estudo é justificada tanto pelo potencial para o desenvolvimento de tecnologias mais empáticas quanto pela contribuição para o campo da saúde mental. A proposta visa atender à necessidade de soluções emocionais em plataformas digitais, um tema importante em um mundo cada vez mais mediado pela tecnologia. Este trabalho pretende preencher essa lacuna no conhecimento e oferecer uma compreensão mais aprofundada sobre como plataformas digitais podem integrar suporte emocional de forma eficaz.

Para o desenvolvimento da plataforma Yourself, foram utilizadas tecnologias web como HTML, CSS e JavaScript, com especial ênfase no uso de CSS para criar um design que reflete e responde aos estados emocionais dos usuários. O HTML foi empregado para estruturar o conteúdo do site, definindo a hierarquia das informações e facilitando a navegação. Cada tag foi cuidadosamente escolhida para garantir acessibilidade e facilidade de uso. O JavaScript, por sua vez, foi implementado para criar funcionalidades dinâmicas e responsivas, permitindo que o site adapte seu conteúdo de acordo com as emoções do usuário. A estilização, por meio do CSS, desempenhou um papel central, utilizando animações, transições e uma paleta de cores específica para proporcionar uma experiência visual que reflete os diferentes estados emocionais. Ao longo deste estudo, cada elemento de HTML, CSS e JavaScript será explicado, com demonstrações visuais do resultado em tela para ilustrar cada etapa.

O objetivo deste estudo é avaliar o impacto da plataforma Yourself na promoção do bem-estar emocional dos usuários, por meio de entretenimento personalizado e adaptação da interface ao estado emocional. A hipótese central é que uma interface sensível às emoções, como a oferecida pela plataforma Yourself, contribui para a regulação emocional e o sentimento de acolhimento dos usuários.

A estrutura deste artigo está organizada da seguinte forma: inicialmente, será apresentado o referencial teórico que embasa o desenvolvimento de tecnologias focadas no bem-estar emocional. Em seguida, a metodologia da pesquisa será descrita, incluindo o processo de coleta e análise de dados, bem como uma análise detalhada dos elementos técnicos usados no desenvolvimento (HTML, CSS e JavaScript). Cada tag e funcionalidade será explicada, juntamente com capturas de tela que demonstram o resultado visual em cada etapa, ilustrando como esses elementos contribuem para a interação e usabilidade da plataforma. Posteriormente, os resultados da análise serão discutidos, avaliando o impacto da plataforma Yourself no bem-estar emocional dos usuários. Por fim, a conclusão abordará as principais contribuições do estudo, limitações e sugestões para futuras pesquisas.

Esse estudo está fundamentado em uma contextualização teórica sobre regulação emocional e a crescente aplicação de tecnologias voltadas ao suporte emocional. A pesquisa considera estudos prévios que exploram o potencial das interfaces digitais interativas, bem como o impacto de plataformas de entretenimento personalizadas para a saúde mental dos usuários. A inclusão de explicações técnicas e visuais sobre o uso de HTML, CSS e JavaScript busca fornecer uma perspectiva prática e aplicável, demonstrando como cada elemento dessas tecnologias enriquece a experiência e funcionalidade da plataforma Yourself.

# 2. DESENVOLVIMENTO

O site Yourself é uma plataforma interativa desenvolvida com o objetivo de ajudar os usuários a entender e gerenciar suas emoções por meio de recomendações personalizadas de conteúdos como filmes, séries, músicas, livros e animes. O desenvolvimento dessa plataforma é composto por várias etapas, desde a coleta dos dados emocionais dos usuários até a geração de recomendações específicas para cada estado emocional identificado. Abaixo, são apresentadas as principais fases do desenvolvimento do site:

## 2.1 Coleta de Dados Emocionais

O primeiro passo para o desenvolvimento da plataforma Yourself é a coleta de dados emocionais, realizada por meio de uma pergunta direta sobre o estado emocional do usuário. O site oferece três opções de resposta: "feliz", "triste" e "entediado". Esse formato simples e direto permite ao usuário identificar rapidamente como se sente, enquanto o sistema coleta essa informação para adaptar as recomendações. Cada opção é processada para definir a experiência ideal, mas é importante ressaltar que a plataforma não oferece apoio médico ou psicológico, funcionando apenas como um ambiente de acolhimento e entretenimento emocional.

## 2.2 Análise de Dados Emocionais

Após o usuário escolher seu estado emocional, os dados são analisados para oferecer uma resposta personalizada. A análise é orientada para recomendar conteúdos que ajudam o usuário a processar suas emoções. Se o usuário se sente triste, o site sugere filmes, séries, músicas, livros e animes que o auxiliem a refletir sobre e entender a tristeza, promovendo alívio e superação. Para quem está feliz, a plataforma indica conteúdos que amplificam esse sentimento positivo, proporcionando uma experiência de felicidade prolongada. No caso do tédio, são oferecidas atividades que estimulam e entretêm o usuário, ajudando-o a erradicar o tédio. Esse processo de análise é conduzido de forma a enriquecer a experiência do usuário sem substituir a orientação profissional de saúde.

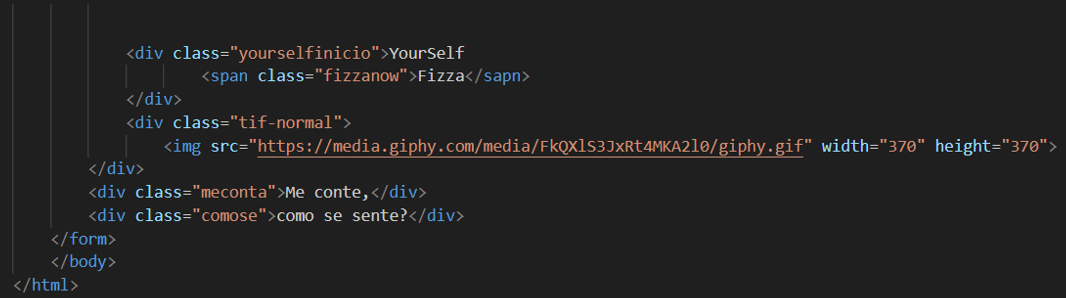
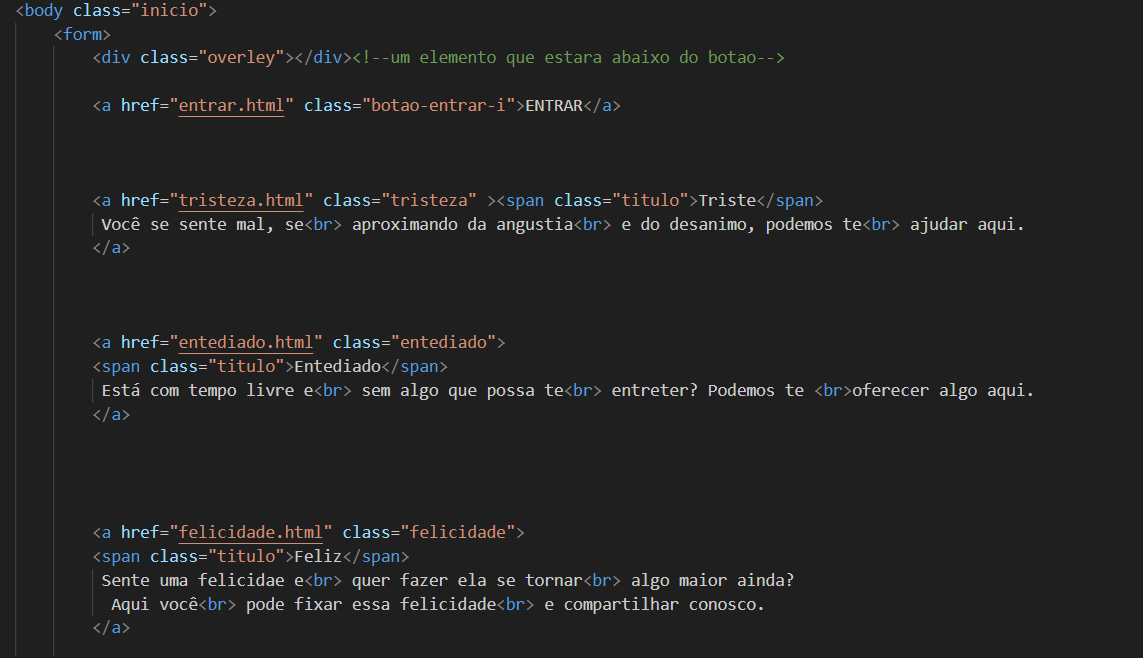
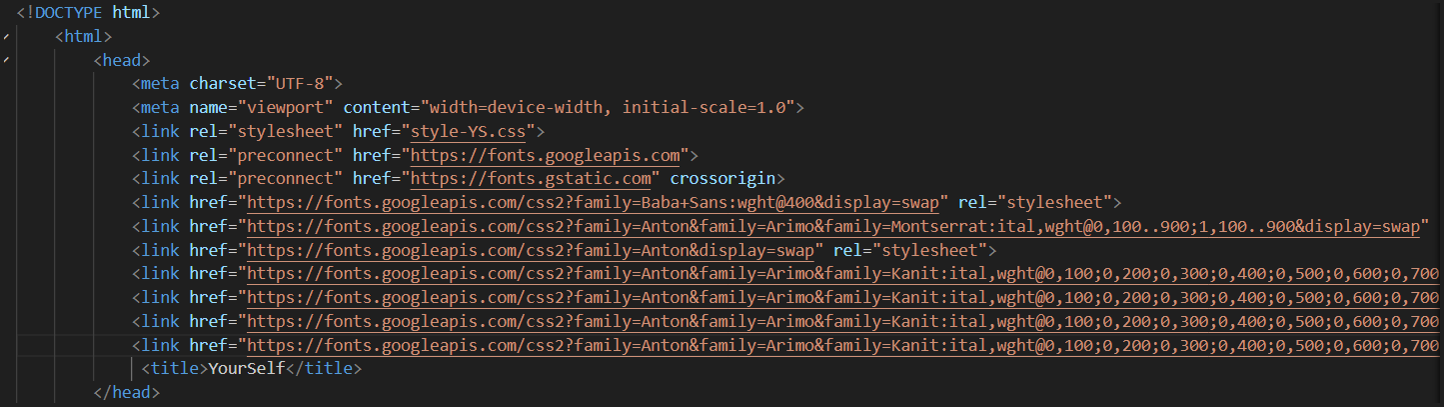
## 2.3 Modelagem de Interação e Algoritmos de Recomendação

A modelagem de interação no Yourself utiliza algoritmos de recomendação para adequar as sugestões de conteúdo ao estado emocional selecionado pelo usuário. Com base em modelos de machine learning e recomendações pré-configuradas, o site "aprende" a ajustar sugestões com base nos padrões de uso. Assim, conteúdos como filmes, livros, músicas, séries e animes são recomendados de acordo com o humor relatado, com o objetivo de ajudar o usuário a explorar e processar suas emoções. No entanto, essas recomendações visam apenas ao entretenimento e acolhimento emocional, sem substituição ao suporte de profissionais de saúde.

## 2.4 Programação e Infraestrutura de Suporte

A construção da plataforma Yourself utilizou HTML, CSS e JavaScript, desenvolvidos no Visual Studio Code para criar uma interface responsiva que se adapta ao estado emocional do usuário. O HTML fornece a estrutura do site, o CSS cria uma experiência visual harmonizada com as emoções do usuário, e o JavaScript permite a funcionalidade dinâmica, exibindo conteúdos específicos para cada resposta emocional. Essa abordagem de programação permite ao site mudar sua interface conforme o sentimento relatado, oferecendo uma experiência personalizada e envolvente. Mesmo com essa personalização, o site permanece como um recurso de apoio e entretenimento emocional, sem oferecer diagnóstico, tratamento ou qualquer substituição ao suporte médico.

**HTML**

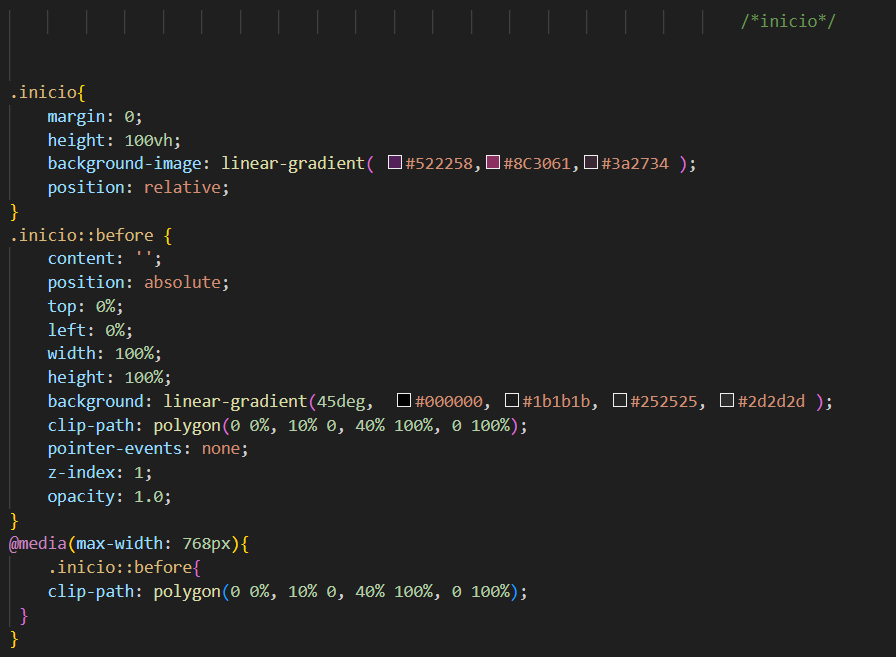
Fonte : Autores (2024)

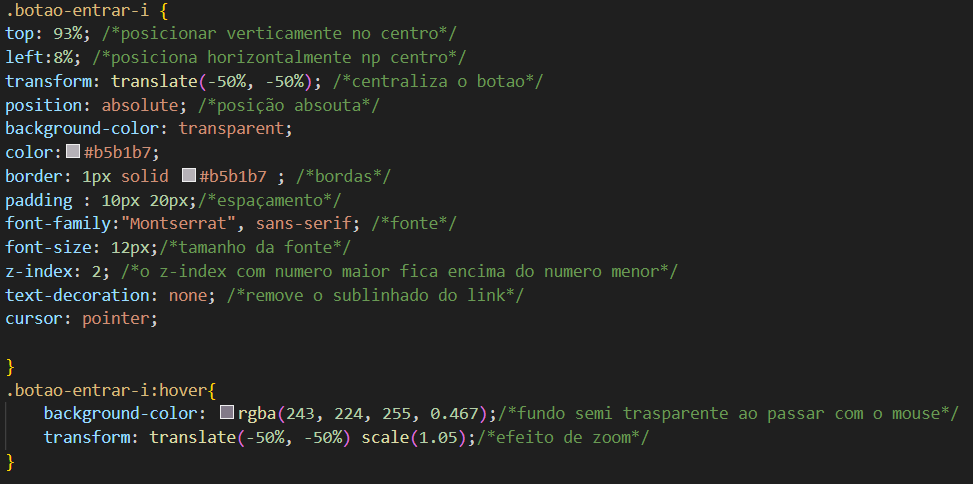
* **<!DOCTYPE html>:** Declara o tipo do documento como HTML5, indicando que o navegador deve interpretar o conteúdo como HTML5.
* **<html>:** Elemento raiz do documento HTML, dentro do qual todos os elementos da página são organizados.
* **<head>:** Contém informações meta e recursos para o navegador, como links de estilos e o título da página.
* **<meta charset="UTF-8">:** Define a codificação de caracteres para UTF-8, permitindo a exibição correta de caracteres especiais.
* **<meta name="viewport" content="width=device-width, initial-scale=1.0">:** Configura a largura da página para corresponder à largura da tela do dispositivo, facilitando a visualização em dispositivos móveis.
* **<link rel="stylesheet" href="style-YS.css">:** Conecta um arquivo CSS externo (style-YS.css) que define o estilo visual do HTML.
* **<link rel="preconnect" href="https://fonts.googleapis.com"> e <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>:** Preconecta aos servidores do Google Fonts para melhorar o desempenho no carregamento das fontes.
* **<link href="https://fonts.googleapis.com/css2?...:** Links para várias fontes hospedadas no Google Fonts, como "Baba Sans", "Anton", "Arimo", "Montserrat", "Kanit" e "League Spartan", para aplicar diferentes estilos de texto.
* **<title>YourSelf</title>:** Define o título da página, que aparece na aba do navegador.
* **<body class="inicio">:** Corpo do HTML, onde o conteúdo principal é inserido. A classe "inicio" aplica o estilo definido para este elemento no CSS.
* **<form>:** Define um formulário, que neste caso está sendo usado para organizar elementos na página sem enviar dados.
* **<div class="overley"></div>:** Cria uma camada de sobreposição na tela para efeitos visuais. Não contém conteúdo, sendo utilizada para posicionamento e estilo.
* **<a href="entrar.html" class="botao-entrar-i">ENTRAR</a>:** Um link estilizado como botão, que redireciona o usuário para a página entrar.html. A classe "botao-entrar-i" aplica o estilo definido no CSS.
* **<a href="tristeza.html" class="tristeza">:** Link para a página tristeza.html, usado para indicar a emoção "triste". O texto adicional e a classe "tristeza" são aplicados para o estilo.
* **<span class="titulo">Triste</span>:** O título dentro do link, com uma classe para aplicar estilos específicos.
* **<a href="entediado.html" class="entediado">:** Link para a página entediado.html, representando o estado emocional "entediado".
* **<span class="titulo">Entediado</span>:** Texto de título com uma classe que aplica estilo específico.
* **<a href="felicidade.html" class="felicidade">:** Link para a página felicidade.html, referente ao estado emocional "feliz".
* **<span class="titulo">Feliz</span>:** Título com a classe "titulo" que estiliza o texto.
* **<div class="yourselfinicio">YourSelf <span class="fizzanow">Fizza</span></div>:** Div com o nome do site "YourSelf" estilizado, com um subtítulo Fizza que usa a classe "fizzanow".
* **<div class="tif-normal">:** Div que contém uma imagem em movimento (gif), representando o mascote "Tef" animado.
* **<img src="https://media.giphy.com/media/FkQXlS3JxRt4MKA2l0/giphy.gif" width="370" height="370">:** A imagem GIF com o tamanho especificado.
* **<div class="meconta">Me conte,</div>:** Div para o texto "Me conte,", com classe "meconta" para o estilo aplicado.
* **<div class="comose">como se sente?</div>:** Div para o texto "como se sente?", também estilizado com a classe "comose".

**RESULTADO:**

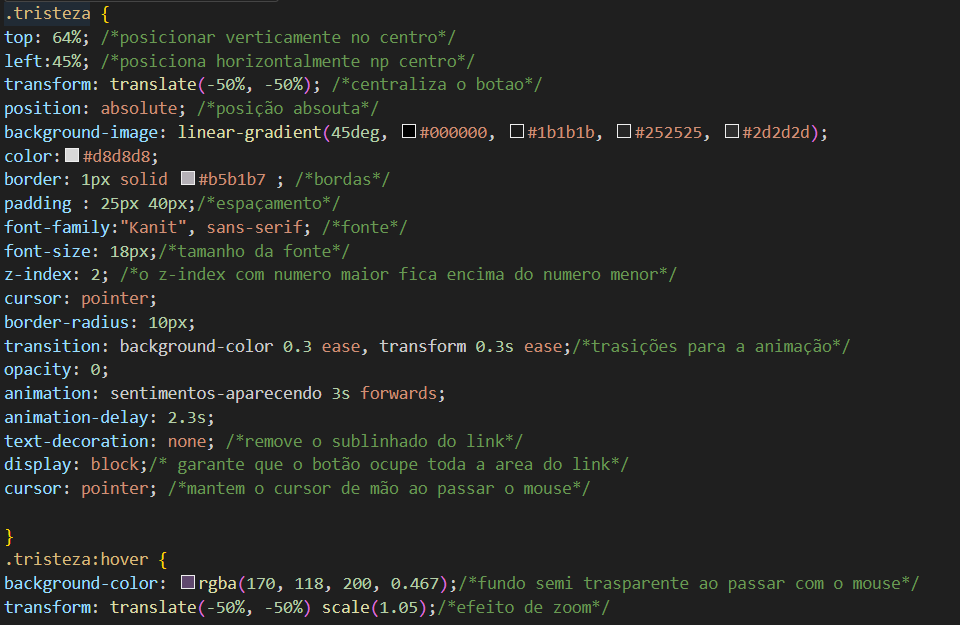
Fonte: Autores (2024)

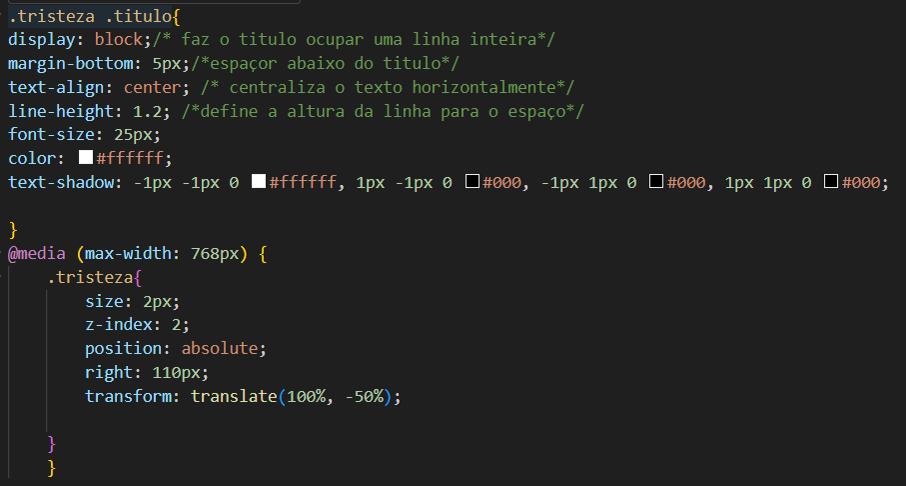
**CSS**





Fonte: Autores (2024)

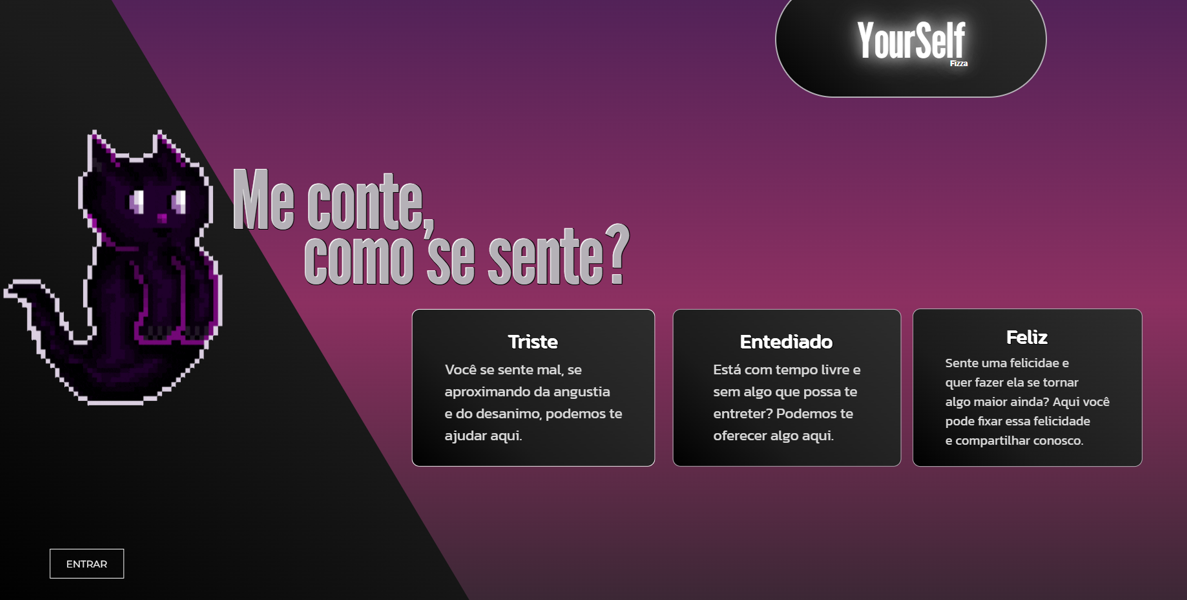




Fonte: Autores (2024)

* **margin:** 0: Remove margens externas.
* **height:** 100vh: Define altura igual à altura da janela.
* **background-image: linear-gradient(...):** Define um fundo com gradiente linear, criando transições suaves de cor.
* **position:** relative: Define o contexto de posicionamento para elementos filhos.
* **.inicio::before**
* **content:** '': Gera um elemento vazio para efeitos visuais.
* **position:** absolute: Posiciona o elemento em relação ao pai.
* **clip-path: polygon(...):** Corta o elemento em uma forma personalizada (polígono), criando um efeito visual específico.
* **pointer-events:** none: Evita que o elemento interaja com eventos do mouse.
* **opacity:** 1.0: Define a opacidade completa do elemento.
* **@media(max-width: 768px):** Define estilos responsivos para telas de até 768px de largura.
* **.botao-entrar-i e .tristeza**
* **top, left, transform, position:** Posiciona o elemento no layout.
* **background-color e background-image:** Definem cor ou gradiente do fundo.
* **border:** Define a borda do botão.
* **padding:** Adiciona espaço interno.
* **font-family e font-size:** Define a fonte e o tamanho do texto.
* **z-index:** Controla a sobreposição no layout.
* **cursor: pointer:** Exibe um cursor de mão ao passar sobre o botão.
* **text-decoration: none**: Remove sublinhado de links.
* **transition:** Define animação para mudanças suaves.
* **animation:** Aplica uma animação de fade-in ou aparecimento.
* **border-radius:** Arredonda as bordas do botão.
* **text-shadow:** Aplica uma sombra ao texto, criando destaque.
* **.titulo**
* **margin-bottom:** Espaço inferior para separar do próximo conteúdo.
* **text-align: center:** Centraliza o texto.
* **line-height:** Ajusta o espaçamento vertical entre linhas.
* **brilho:** Define uma animação de brilho no texto.
* **tif-aparecendo e aparecendo:** Controlam a opacidade do elemento, criando uma transição de invisível para visível.
* **.yourselfinicio**
* **display:** inline-block: Exibe como um bloco em linha.
* **padding:** Adiciona espaçamento interno.
* **border-radius:** Arredonda as bordas do elemento.
* **.tif-normal e .meconta**
* **display:** flex e display: block: Controlam o layout do elemento.
* **object-fit:** cover: Ajusta a imagem para preencher o espaço definido.
* **.fizzanow**
* **Define um elemento de texto com position:** absolute, ajustando sua posição e estilo para complementar o layout.

**RESULTADO:**



Fonte: Autores (2024)

**HTML**







Fonte: Autores (2024)

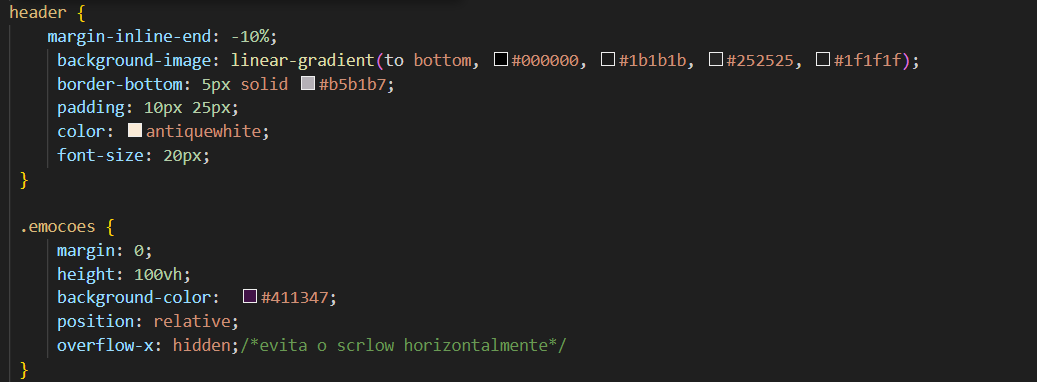
* **<!DOCTYPE html>:** Declara o tipo de documento HTML5 para ajudar os navegadores a renderizar a página corretamente.
* **<html>:** Elemento raiz do documento HTML.
* **Cabeçalho do Documento (<head>)**
* **<meta charset="UTF-8" />:** Define a codificação de caracteres como UTF-8, garantindo a compatibilidade com diferentes caracteres.
* **<title>Felicidade</title>:** Define o título da página que aparece na aba do navegador.
* **<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0-beta3/css/all.min.css">:** Importa o CSS do Font Awesome para adicionar ícones.
* **<link rel="stylesheet" href="style-YS.css">:** Importa o arquivo CSS principal, style-YS.css, que define o estilo da página.
* **Corpo do Documento (<body class="emocoes">):** Define o conteúdo da página com a classe emocoes, que provavelmente aplica estilos específicos relacionados a essa categoria de emoções.
* **Header (<header>)**
* **<div class="tif-entediado">:** Cria uma seção de introdução com a classe tif-entediado.
* **<img src="https://media.giphy.com/media/pgs2S4c8b0GozBTyVe/giphy.gif" width="130" height="130">:** Exibe uma imagem GIF de tamanho 130x130 pixels.
* **<p class="tif-fala">:** Parágrafo com uma mensagem motivacional sobre felicidade. A classe tif-fala define o estilo do texto exibido.
* **Link de Navegação (<a href="entrar.html" class="botao-entrar">ENTRAR</a>):** Botão de navegação que direciona o usuário para a página de login (entrar.html). A classe botao-entrar estiliza o botão.
* **Seção de Filmes (<div class="catalog-container">):** Esta seção exibe uma lista de filmes recomendados.
* **<label class="filme">FILMES:</label>:** Título da seção que indica que os itens abaixo são relacionados a filmes.
* **<button class="catalog-btn prev-btn">&lt;</button>:** Botão de navegação para percorrer a lista de filmes para a esquerda.
* **Contêiner e Itens do Catálogo de Filmes**
* **<div class="catalog-track-container">:** Contém o carrossel dos filmes para rolamento horizontal.
* **<div class="catalog-track">:** Agrupa todos os itens do catálogo.
* **<div class="catalog-item">:** Representa um item individual no catálogo.
* **<img src="...">:** Exibe uma imagem do pôster do filme.
* **<h1 class="titulo-filme">Mamma Mia!</h1>:** Título do filme.
* **<p class="discricao">:** Contém a descrição do filme, incluindo informações como gênero, duração, ano e elenco.
* **<h5 class="descricao">:** Subtítulo para destacar cada detalhe sobre o filme.
* <div class="actions">: Agrupa botões de interação para o usuário.
* **<button class="like-btn">:** Botão para curtir, identificado por um coração (ícone do Font Awesome).
* **<button class="save-btn">:** Botão para salvar, identificado por um ícone de adição.
* **<button class="assistido-btn">:** Botão para marcar como "visto", com um ícone de marca de verificação.

**RESULTADO:**

****

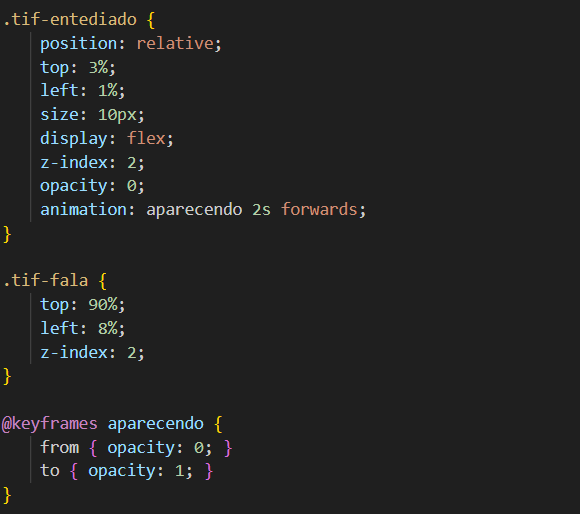
Fonte: Autores (2024)

**CSS**



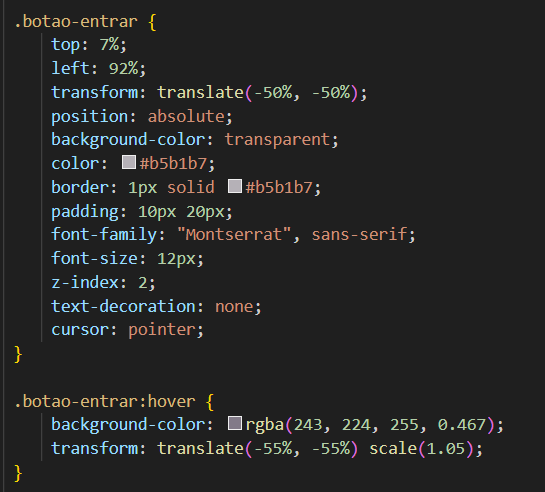
Fonte: Autores (2024)

* **1. header**
* **Propriedades:**
* **background:** Define o fundo do cabeçalho com um gradiente de cores.
* **border-bottom:** Adiciona uma borda fina na parte inferior do cabeçalho.
* **font-family:** Define a família da fonte utilizada no texto dentro do cabeçalho (neste caso, "Georgia", uma fonte serifada).
* **font-size:** Define o tamanho da fonte para o texto dentro do cabeçalho.
* **padding:** Define o espaçamento interno do cabeçalho, ou seja, o espaço entre o conteúdo (como o texto) e as bordas do elemento.
* **2. .emocoes**
* **Propriedades:**
* **height:** Define a altura do elemento .emocoes como 100% da altura da janela de visualização (viewport).
* **background:** Define um fundo com um gradiente linear.
* **overflow-x:** Especifica o comportamento do conteúdo que ultrapassa o limite da largura do contêiner. Aqui, é hidden, para evitar rolagem horizontal.
* **overflow-y:** Define o comportamento da rolagem vertical como auto, ou seja, uma barra de rolagem vertical aparecerá quando o conteúdo ultrapassar a altura definida.



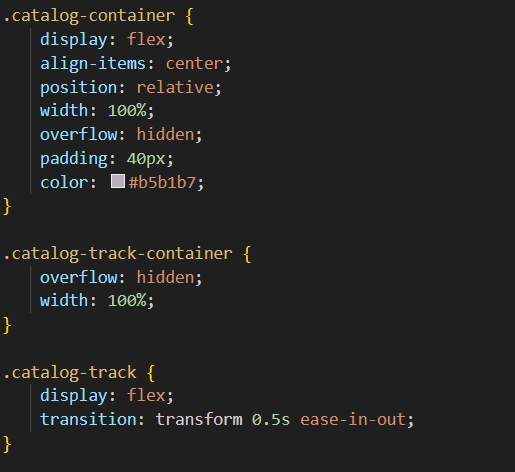
Fonte: Autores (2024)

* **3. .tif-entendiado**
* **Propriedades:**
* **animation:** Aplica uma animação chamada fadein que faz a transição de opacidade do mascote.
* **animation-duration:** Define que a animação levará 2 segundos para ser concluída.
* **animation-fill-mode:** Define que, após a animação, o estilo final da animação será mantido (opacidade 1).



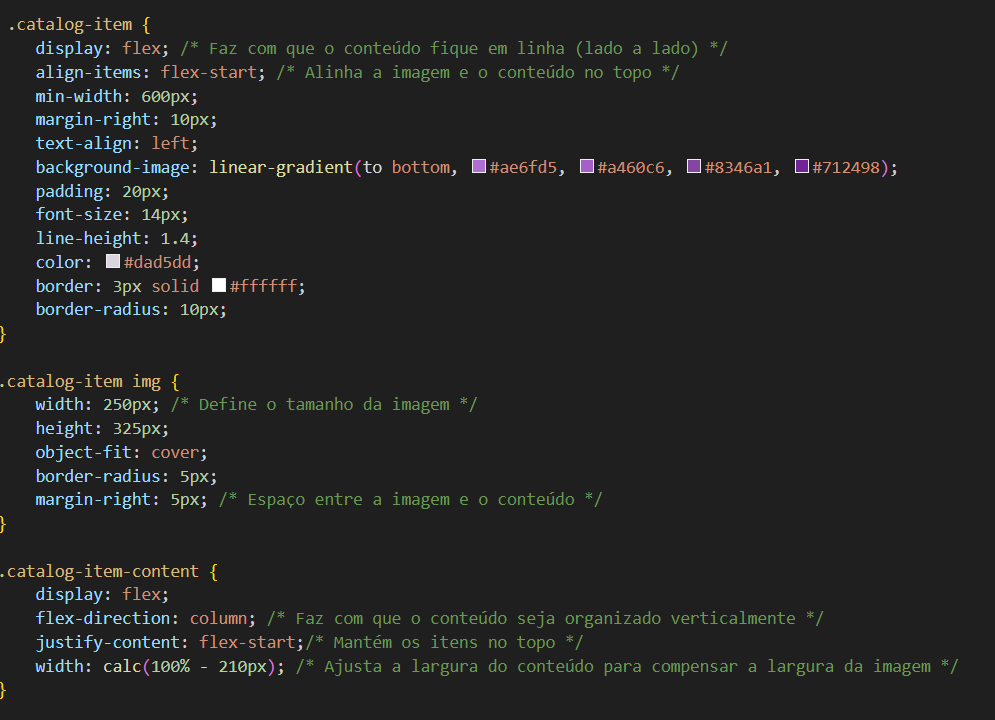
Fonte: Autores (2024)

* **4. .botao-entrar**
* **Propriedades:**
* **position:** Define a posição do botão como absolute, posicionando-o em relação ao elemento pai (neste caso, o body).
* **top e right:** Posiciona o botão no canto superior direito.
* **padding:** Define o espaçamento interno do botão.
* **border:** Define o estilo da borda do botão.
* **border-radius:** Arredonda os cantos do botão.
* **background:** Define a cor de fundo como transparente inicialmente.
* **color:** Define a cor do texto dentro do botão.
* **transition:** Define uma animação de transição para a mudança da cor de fundo e transformação (quando o botão é pressionado ou o mouse passa sobre ele).



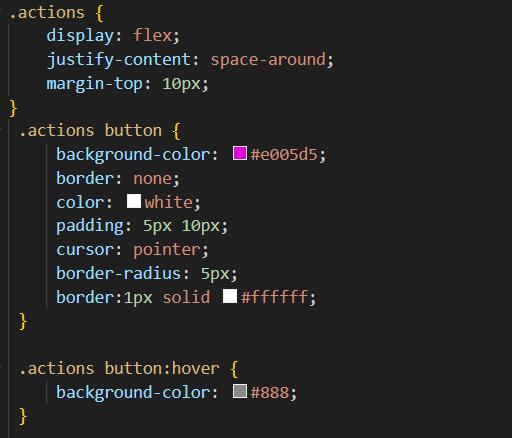
Fonte: Autores (2024)

* **5. .catalog-container**
* **Propriedades:**
* **display:** Define que o contêiner usará o modelo de layout flexível (flex), o que permite organizar os elementos filhos em uma linha ou coluna.
* **align-items:** Alinha os itens ao centro verticalmente dentro do contêiner.
* **justify-content:** Alinha os itens no centro horizontalmente.
* **overflow-x:** Define que a rolagem horizontal estará ativada quando necessário (caso os itens ultrapassem o tamanho da tela).
* **6. .catalog-track-container**
* **Propriedades:**
* **display:** Também usa o modelo de layout flexível (flex), para dispor os itens dentro dele de maneira alinhada.
* **transition:** Permite uma transição suave quando os itens se movem para a esquerda ou direita.
* **7. .catalog-track**
* **Propriedades:**
* **display:** Define o layout como flex, permitindo que os itens se alinhem automaticamente na direção do eixo horizontal.
* **transition:** Aplica uma animação de transição suave para o movimento de transformação (como rolar os itens).



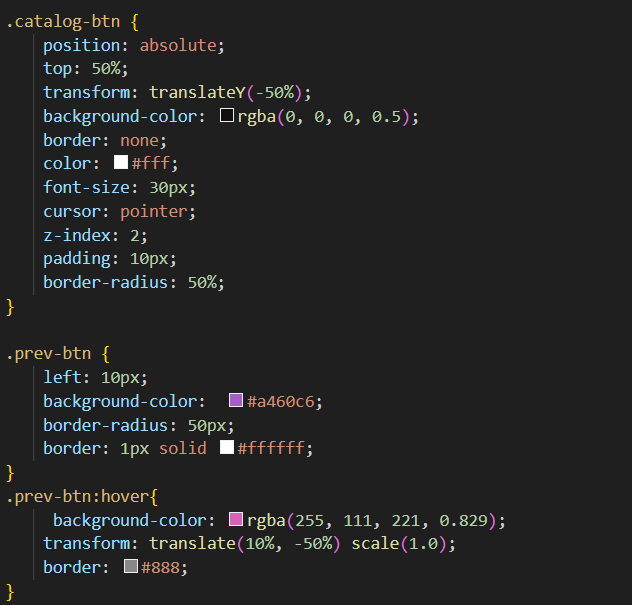
Fonte: Autores (2024)

* **8. .catalog-item**
* **Propriedades:**
* **position:** Define o position do item como relative, permitindo ajustar a posição dos filhos.
* **margin:** Define as margens ao redor de cada item do catálogo.
* **border-radius:** Arredonda os cantos do item do catálogo.
* **background:** Aplica um fundo com gradiente no item.
* **width:** Define a largura do item do catálogo.
* **box-shadow:** Aplica uma sombra suave ao redor do item.
* **display:** Define o uso de flex para organizar a imagem e o conteúdo do item.
* **9. .catalog-item img**
* **Propriedades:**
* **width e height:** Define o tamanho fixo da imagem dentro do item.
* **object-fit:** Define o comportamento da imagem para cobrir toda a área designada sem distorção.
* **10. .catalog-item-content**
* **Propriedades:**
* **display:** Organiza o conteúdo do item do catálogo verticalmente.
* **align-items:** Centraliza o conteúdo verticalmente dentro do item.
* **flex-direction:** Organiza o conteúdo na direção de coluna.

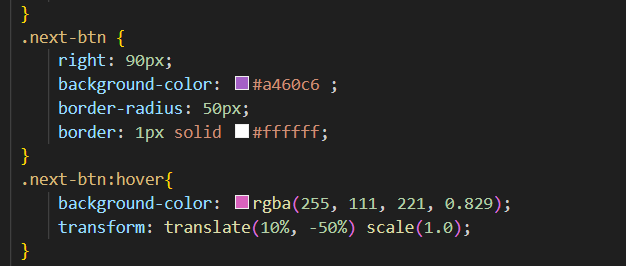


Fonte: Autores (2024)

* **11. .actions**
* **Propriedades:**
* **display:** Define o layout do contêiner como flex para alinhar os botões de ação horizontalmente.
* **12. .actions button**
* **Propriedades:**
* **background:** Define a cor de fundo dos botões de ação.
* **color:** Define a cor do texto.
* **padding:** Define o espaçamento interno dos botões.
* **border:** Remove a borda padrão dos botões.
* **border-radius:** Arredonda os cantos dos botões.
* **cursor:** Define que o cursor se transforma em uma mão ao passar sobre os botões.
* **transition:** Define uma animação suave para a mudança de cor do botão ao interagir.

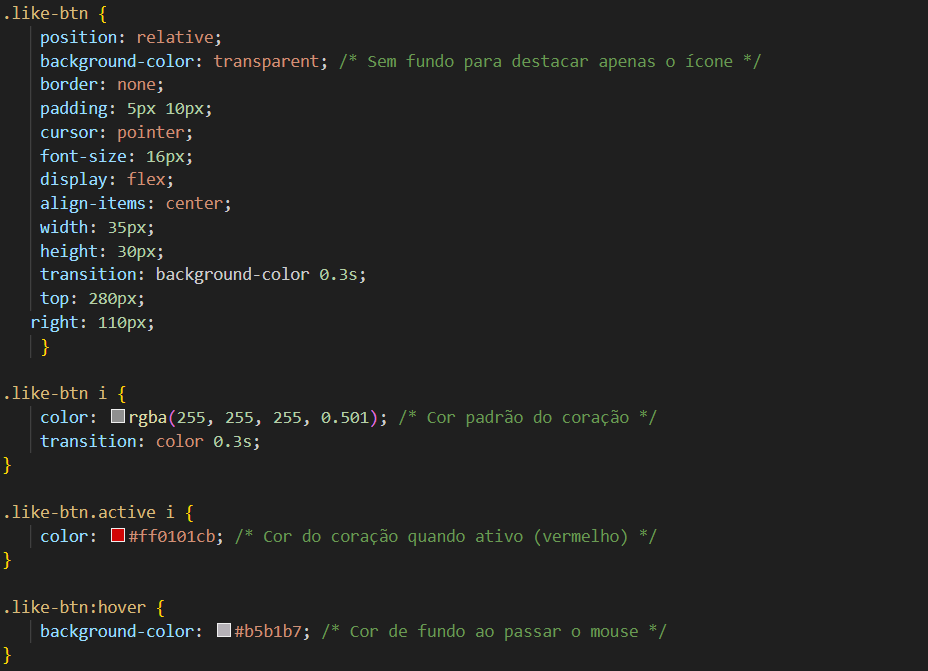


Fonte: Autores (2024)



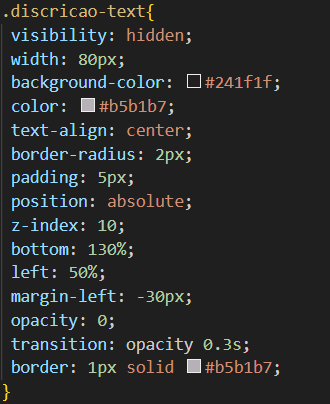
Fonte: Autores (2024)

* **13. .catalog-btn**
* **Propriedades:**
* **background:** Define a cor de fundo do botão.
* **border-radius:** Arredonda os cantos do botão.
* **padding:** Ajusta o espaçamento interno do botão.
* **14. .prev-btn e .next-btn**
* **Propriedades:**
* **position:** Define o posicionamento absoluto dos botões de navegação, permitindo movê-los para os cantos esquerdo e direito da tela.



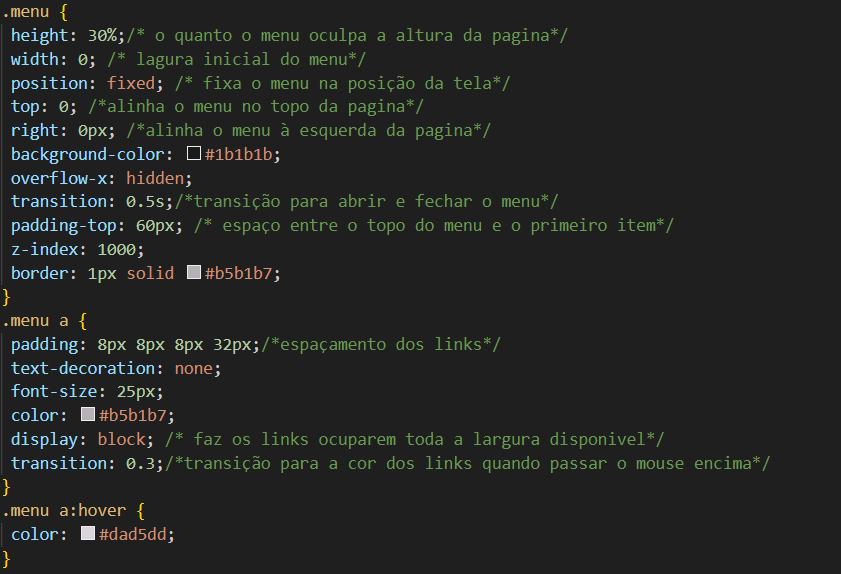
Fonte: Autores (2024)

* **15. .like-btn, .save-btn, .assistido-btn**
* **Propriedades:**
* **background:** Define a cor de fundo do botão.
* **border-radius:** Arredonda os cantos.
* **color:** Define a cor do texto no botão.
* **transition:** Aplica uma animação quando o botão é pressionado ou interagido com.



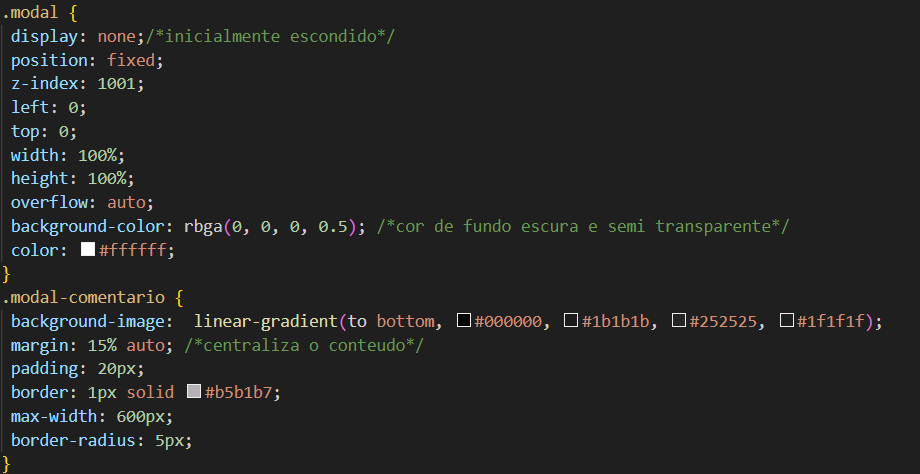
Fonte: Autores (2024)

* **16. .discricao-text**
* **Propriedades:**
* **color:** Define a cor do texto de descrição.
* **font-size:** Define o tamanho da fonte para a descrição.



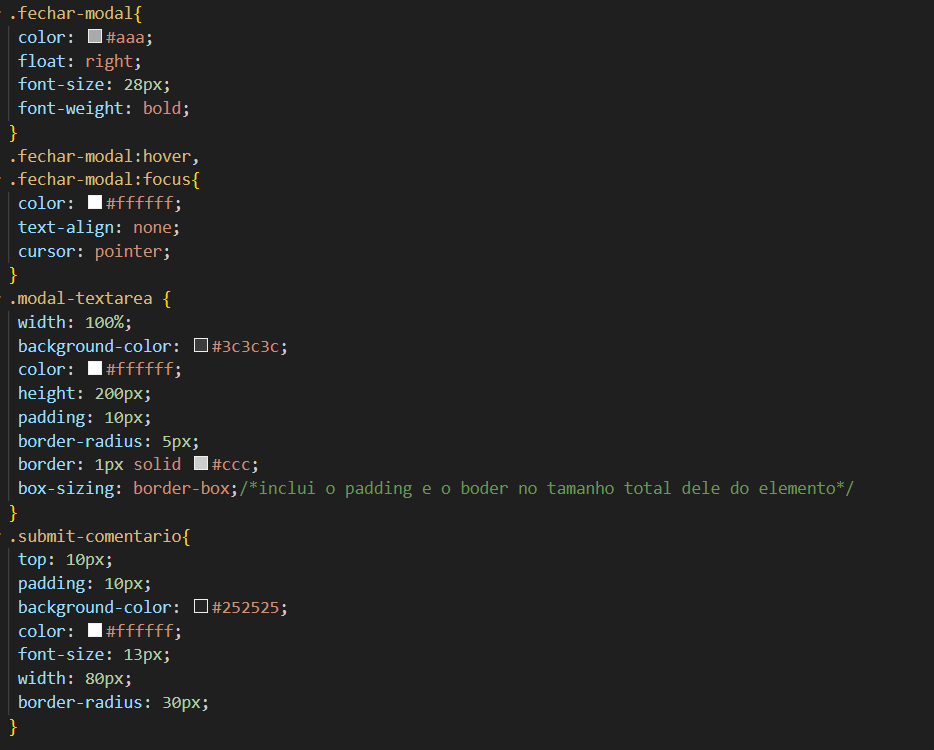
Fonte: Autores (2024)

* **17. .menu**
* **Propriedades:**
* **position:** Posiciona o menu de forma absoluta, removendo-o do fluxo normal da página.
* **top, left, width, height:** Controlam o posicionamento e as dimensões do menu.
* **background:** Aplica um fundo escuro ao menu.
* **18. .abrir-btn**
* **Propriedades:**
* **background:** Define o fundo do botão de abrir o menu.
* **border-radius:** Arredonda os cantos do botão.
* **color:** Define a cor do texto.

****

Fonte: Autores (2024)

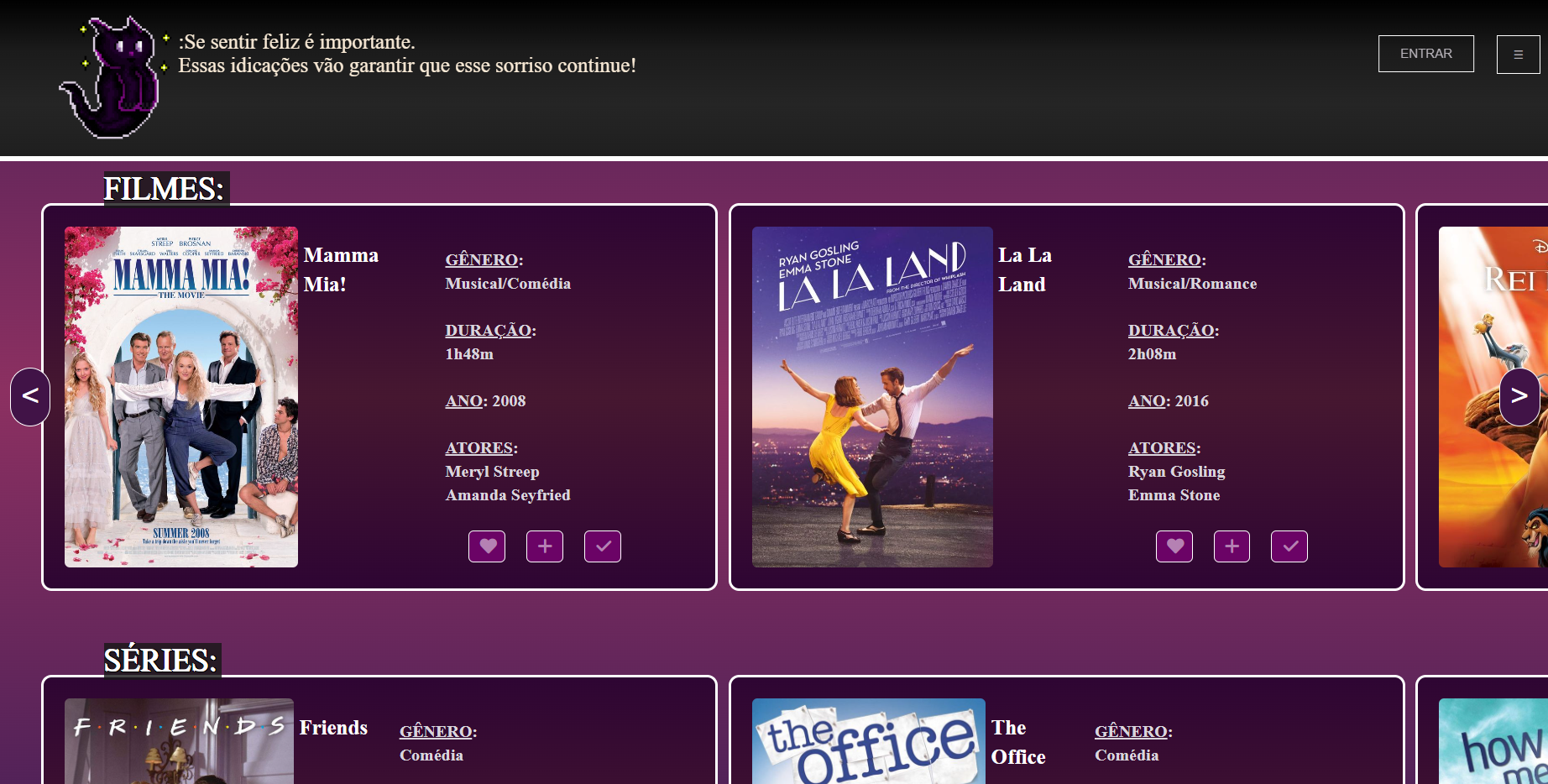
* **19. .modal**
* **Propriedades:**
* **display:** Define o valor none, ou seja, a janela modal inicialmente não será visível.
* **position:** Posiciona a modal de forma absoluta e centraliza na tela.
* **background:** Aplica um fundo semitransparente sobre o conteúdo da página.
* **20. .modal-comentario**
* **Propriedades:**
* **background:** Define um fundo com gradiente na área de comentário.
* **padding:** Aplica espaçamento interno.
* **border-radius:** Arredonda os cantos da área de texto.



Fonte: Autores (2024)

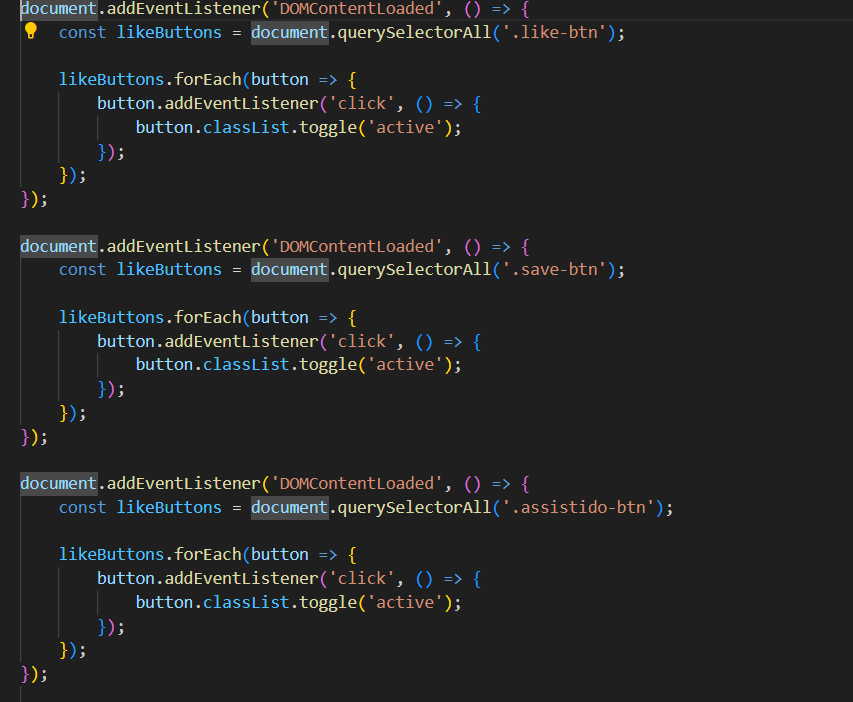
* **21. .fechar-modal**
* **Propriedades:**
* **position:** Posiciona o botão no canto superior direito da modal.
* **background:** Define o fundo do botão de fechar.
* **22. .modal-textarea**
* **Propriedades:**
* **resize:** Permite que o campo de texto seja redimensionado.
* **background:** Define a cor de fundo do campo de texto.
* **23. .submit-comentario**
* **Propriedades:**
* **background:** Define a cor de fundo do botão de envio de comentário.
* **color:** Define a cor do texto dentro do botão.
* **padding:** Adiciona espaçamento interno ao botão.

**RESULTADO:**

****

Fonte: Autores (2024)

**JAVASCRIPT**



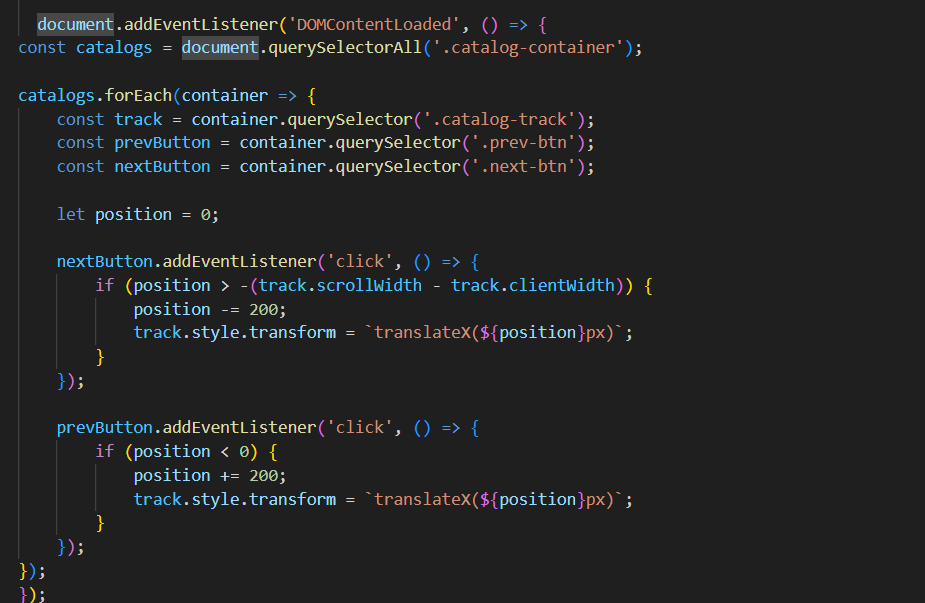
Fonte: Autores (2024)

* **1. <script>:** A tag <script> é usada para incluir código JavaScript em uma página HTML. Dentro dessa tag, você coloca o código que controla a interação do usuário, animações, e funcionalidades dinâmicas.
* **2. document.addEventListener('DOMContentLoaded', () => {...});:** document.addEventListener('DOMContentLoaded', ...): Isso define um evento para garantir que o código dentro da função seja executado somente depois que todo o conteúdo HTML da página for carregado.
* **'DOMContentLoaded':** Um evento que dispara quando o HTML da página foi completamente carregado e analisado, sem esperar o carregamento completo de imagens ou outros recursos.
* **3. document.querySelectorAll('.like-btn'):**
* **querySelectorAll('.like-btn'):** Seleciona todos os elementos da página que possuem a classe like-btn. A classe é usada para identificar botões "Curtir" na página.
* **4. button.addEventListener('click', () => {...});:**
* **button.addEventListener('click', ...):** Adiciona um ouvinte de evento para cada botão selecionado. Quando o botão for clicado, a função dentro do addEventListener será executada.
* **'click':** O tipo de evento que estamos ouvindo (clicar no botão).
* **5. button.classList.toggle('active'):**
* **button.classList.toggle('active'):** Essa linha adiciona ou remove a classe active do botão, dependendo do seu estado atual. Se o botão já tiver a classe, ela será removida, caso contrário, será adicionada.

**RESULTADO:**

****

****



Fonte: Autores (2024)

* **6. Função do Carrossel de Catálogo:**
* **Dentro do código de carrossel:**
* **catalogs.forEach(container => {...});:** Seleciona todos os containers de catálogo e aplica uma função a cada um deles.
* **track.style.transform = 'translateX(...)':** Altera a posição do carrossel para a esquerda ou direita, mudando a posição do container de imagens (ou elementos do catálogo) no eixo X.
* **prevButton.addEventListener('click', () => {...}) e nextButton.addEventListener('click', () => {...}):** Esses eventos são atribuídos aos botões de navegação "anterior" e "próximo", que alteram a posição do carrossel.

**RESULTADO:**

****



Fonte: Autores (2024)

* **7. Função openMenu():** A função openMenu() altera o estilo do elemento com o id "menu", definindo sua largura para 250px. Isso faz com que o menu lateral se abra na tela.
* **8. Função closeMenu():** A função closeMenu() altera o estilo do elemento com o id "menu", definindo sua largura para 0px, fazendo com que o menu lateral se feche.
* 9. **Função openModal():** A função openModal() altera o estilo do elemento com o id "modal", definindo seu estilo display para 'block'. Isso faz com que o modal (uma janela pop-up) apareça na tela.
* **10. Função closeModal():** A função closeModal() altera o estilo do elemento com o id "modal", definindo seu estilo display para 'none'. Isso faz com que o modal desapareça da tela.
* **11. document.getElementById('openModal').addEventListener('click', ...): getElementById('openModal'):** Seleciona o elemento com o id openModal, que é um link ou botão que abrirá o modal.
* **.addEventListener('click', ...):** Adiciona um ouvinte de evento que executa a função quando o link ou botão é clicado.
* **event.preventDefault():** Impede que o link faça sua ação padrão (por exemplo, redirecionar para outra página).
* **openModal():** Chama a função que abrirá o modal de comentários.
* **12. window.onclick = function(event) {...}: window.onclick:** Define um evento de clique para qualquer lugar na janela do navegador. A função verifica se o clique foi na área de fundo do modal (ou seja, fora do conteúdo do modal). Se foi, chama closeModal() para fechar o modal.

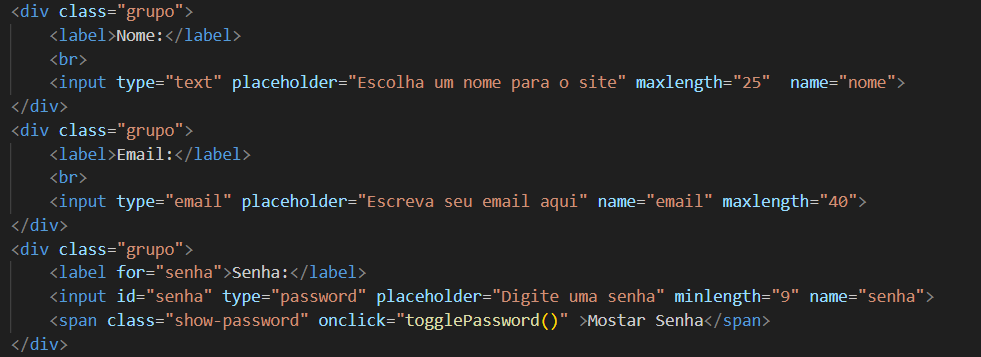
**RESULTADO:**

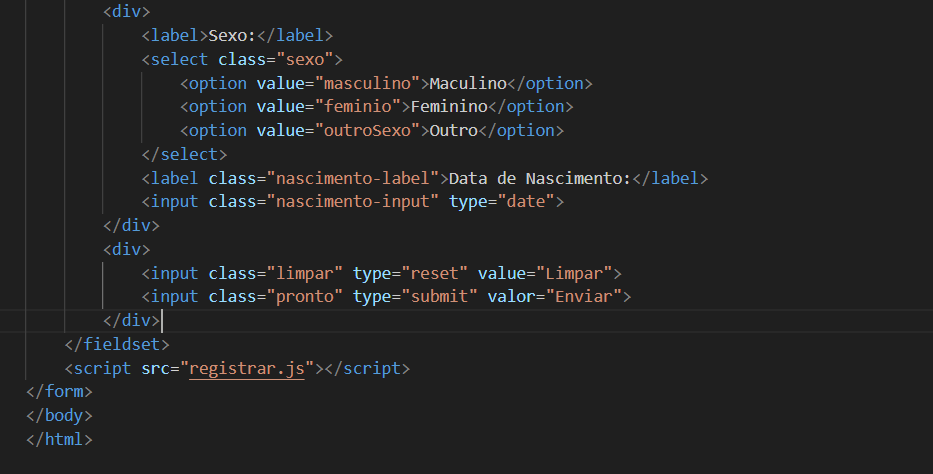
****

Fonte: Autores (2024)

**HTML**



****

****

Fonte: Autores (2024)

* **<html>:** Define o início do documento HTML.
* **<head>:** Contém metadados e links externos do documento.
* <**meta charset="UTF-8" />:** Define a codificação de caracteres para UTF-8, permitindo o uso de caracteres especiais.
* **<link href="entrar.css" rel="stylesheet">:** Linka o arquivo CSS externo "entrar.css" para estilização da página.
* **<title>Registrar</title>:** Define o título da página que aparece na aba do navegador.
* **Formulário de Cadastro**
* **<form>:** Inicia o formulário, englobando todos os campos de entrada e botões relacionados ao cadastro.
* **<fieldset class="box-registrar">:** Agrupa os campos do formulário dentro de uma caixa, aplicando a classe "box-registrar" para estilização.
* **<legend>:** Define uma legenda para o <fieldset>, neste caso, com uma imagem (GIF) inserida dentro da legenda.
* **<div>:** Usado para dividir seções do formulário, facilitando a aplicação de estilos.
* **Elementos de Entrada**
* **Título de Cadastro**
* **<a class="titulo"><h2>Cadastrar</h2></a>:** Define um cabeçalho para o formulário com o texto “Cadastrar”.
* **Campo Nome**
* **<label>Nome:</label>:** Rótulo para o campo de nome, indicando o propósito do campo de entrada.
* **<input type="text" placeholder="Escolha um nome para o site" maxlength="25" name="nome">:** Campo de texto para inserir o nome, com um texto guia, limite de caracteres, e o nome do campo para envio.
* **Campo Email**
* **<label>Email:</label>:** Rótulo para o campo de email.
* **<input type="email" placeholder="Escreva seu email aqui" name="email" maxlength="40">:** Campo de entrada para email, com validação automática para endereços de email.
* **Campo Senha**
* **<label for="senha">Senha:</label>:** Rótulo vinculado ao campo de senha por meio do atributo for, que corresponde ao id="senha" do campo de entrada.
* **<input id="senha" type="password" placeholder="Digite uma senha" minlength="9" name="senha">:** Campo para inserção de senha, com ocultação automática dos caracteres e um mínimo de 9 caracteres exigido.
* **<span class="show-password" onclick="togglePassword()">Mostrar Senha</span>:** Texto interativo que, ao ser clicado, chama a função togglePassword() para alternar a visibilidade da senha.
* **Seleção de Sexo**
* **<label>Sexo:</label>:** Rótulo indicando o propósito do próximo campo de seleção.
* **<select class="sexo">:** Menu suspenso para selecionar o sexo, com três opções definidas por **<option>:**
* **<option value="masculino">Masculino</option>**
* **<option value="feminino">Feminino</option>**
* **<option value="outroSexo">Outro</option>**
* **Data de Nascimento**
* **<label class="nascimento-label">Data de Nascimento:</label>:** Rótulo para o campo de data.
* **<input class="nascimento-input" type="date">:** Campo de entrada para selecionar uma data, utilizando um seletor de data padrão do navegador.
* **Botões do Formulário**
* **<input class="limpar" type="reset" value="Limpar">:** Botão que limpa todos os campos preenchidos no formulário.
* **<input class="pronto" type="submit" value="Enviar">:** Botão para enviar o formulário.
* **Script Externo**
* **<script src="registrar.js"></script>:** Importa o arquivo JavaScript "registrar.js", possivelmente contendo a função togglePassword() e outras funcionalidades adicionais para o formulário.

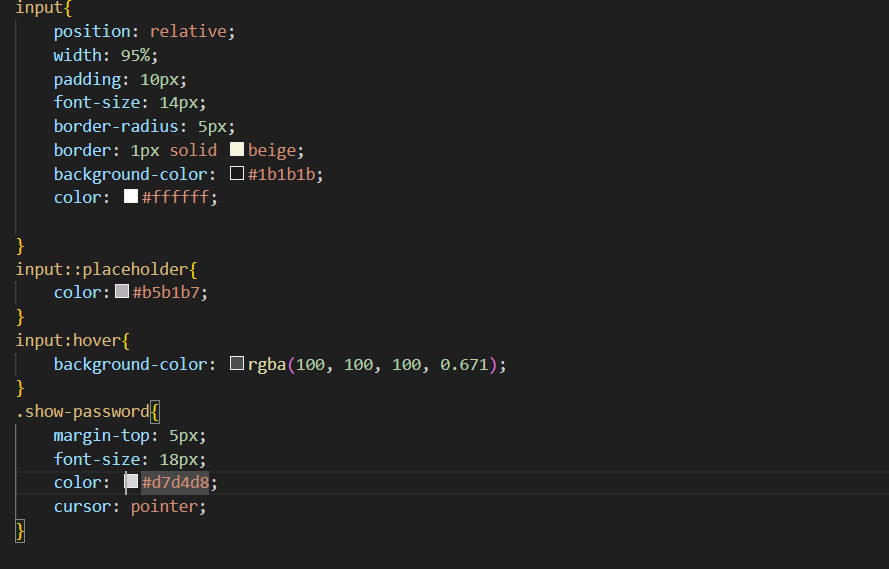
**RESULTADO:**



Fonte: Autores (2024)

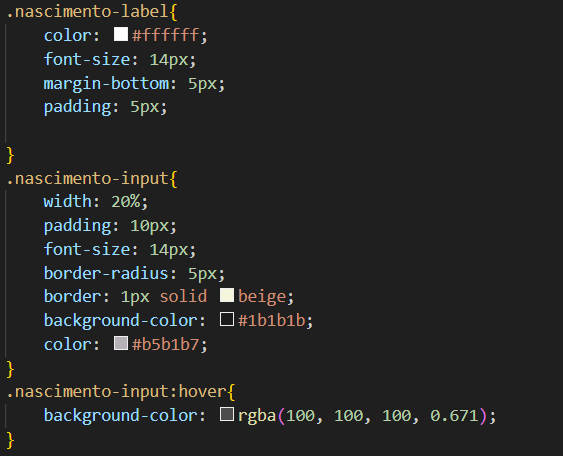
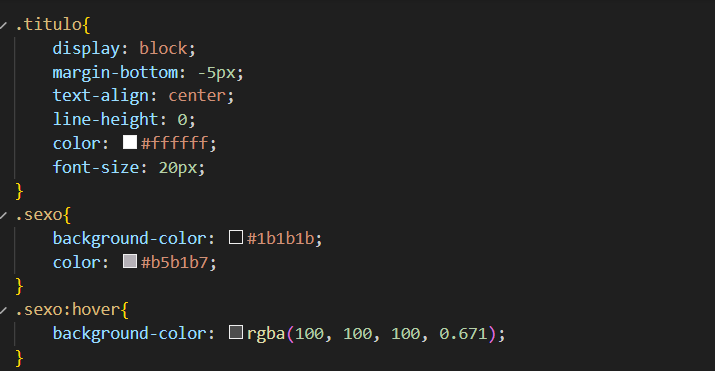
**CSS**



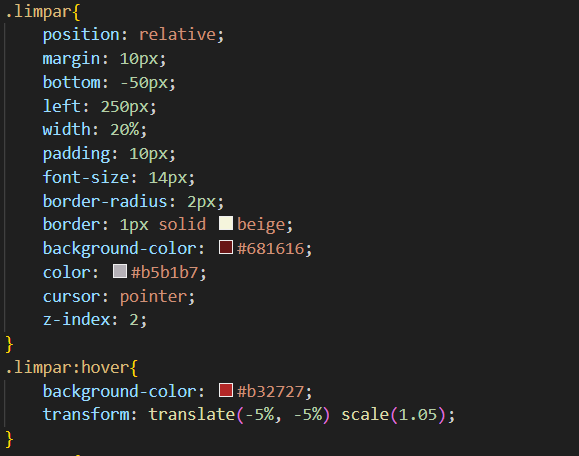


Fonte: Autores (2024)

* **Estilo Geral**
* **body:** Define o layout básico da página, com height: 100vh para ocupar toda a altura da viewport, margin: 0 para remover margens padrão, e background-image com um gradiente linear nas cores (#522258, #8C3061, #3a2734).
* **position:** relative e **overflow:** hidden ajudam a controlar a exibição e o layout da página.
* **Elementos do Formulário de Cadastro**
* .**box-registrar:** Caixa principal do formulário, posicionada a 10% do topo e a 30% da esquerda, com dimensões de 500px x 500px. A cor de fundo é um gradiente linear, com uma borda arredondada e color: #ffffff para deixar o texto branco. A opacidade total (opacity: 1.0) assegura que o elemento seja completamente visível.
* **.grupo:** Margem inferior aplicada aos campos para criar espaçamento entre os elementos do formulário.
* **Estilo dos Rótulos e Campos**
* **label:** Define o estilo do texto dos rótulos com color: #ffffff e font-size: 14px para manter a aparência consistente e legível.
* **input:** Aplica estilo a todos os campos de entrada, com largura de 95%, padding de 10px, borda arredondada, e fundo escuro (background-color: #1b1b1b). O texto é branco para contraste, e o placeholder herda color: #ffffff.
* **input:hover:** Altera a cor de fundo para um cinza mais claro (rgba) ao passar o mouse, dando uma dica visual interativa ao usuário.
* **.show-password:** Estiliza o botão “Mostrar Senha” com uma fonte de 18px, cor de texto #d7d4d8, e cursor pointer para indicar que é clicável.

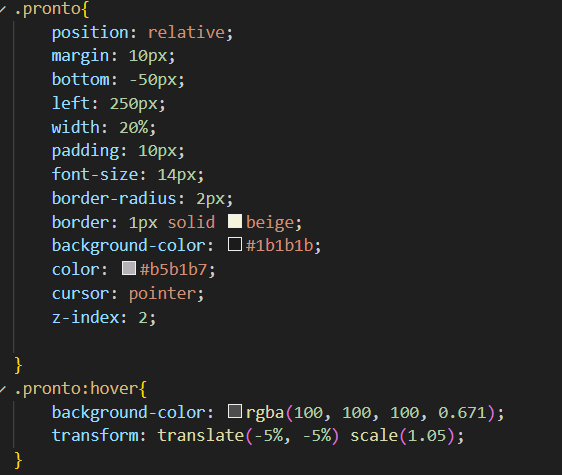


Fonte: Autores (2024)

* **Estilo para o Título**
* **.titulo:** Centraliza o título com text-align: center e remove a altura da linha (line-height: 0) para aproximar o título do campo abaixo.
* **Estilo do Campo de Seleção e Data de Nascimento**
* **.sexo:** Estiliza o menu de seleção com fundo escuro (#1b1b1b) e cor do texto #b5b1b7. No hover, muda para um cinza mais claro para indicar interação.
* **.nascimento-label e .nascimento-input:** Aplica estilo ao rótulo e campo de data, ambos com fundo e cor escuros, com o input recebendo padding, borda arredondada, e uma cor mais clara ao passar o mouse (hover).

Fonte: Autores (2024)

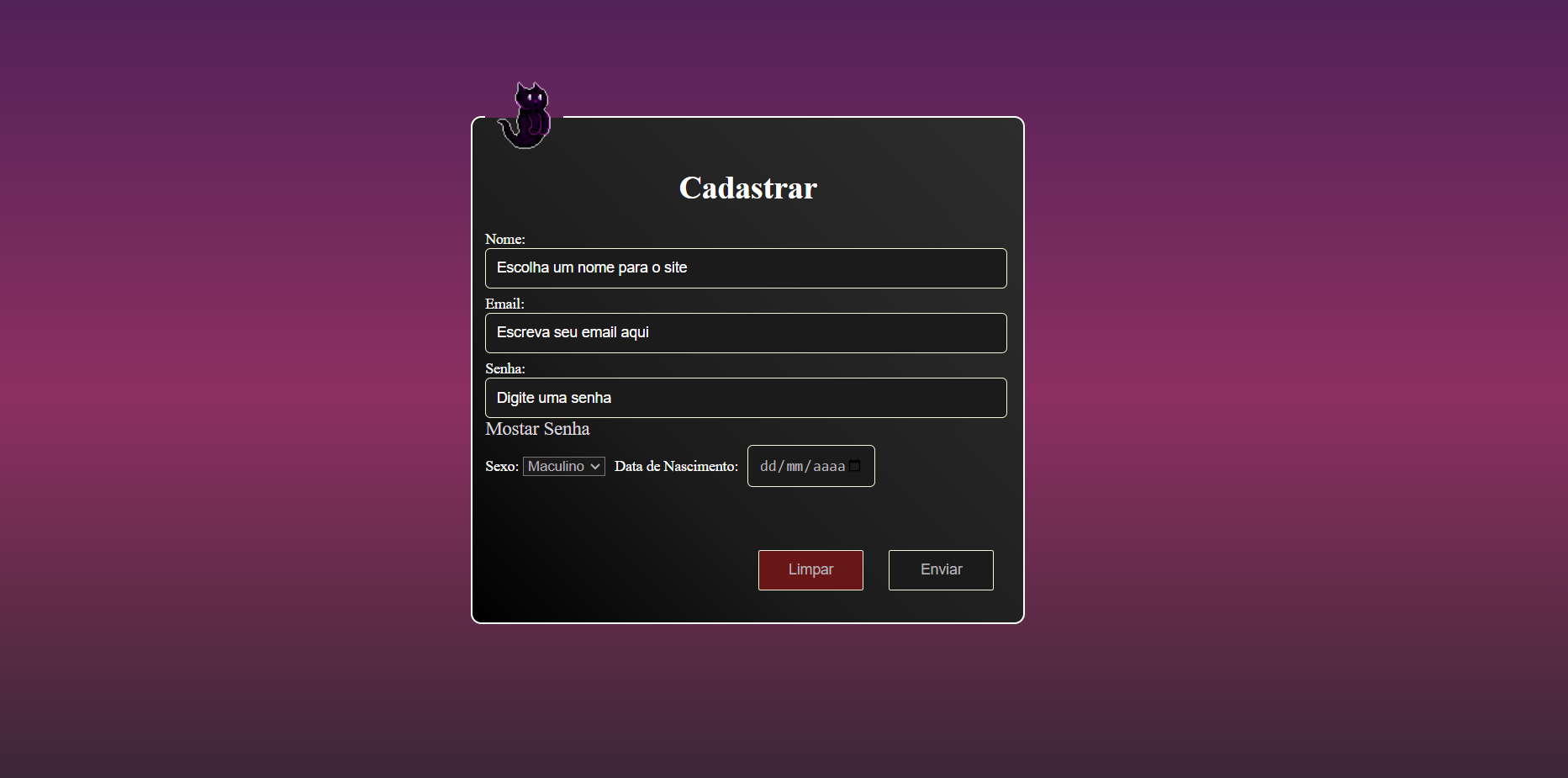
* **Botões de Limpar e Enviar**
* **.limpar:** Estiliza o botão "Limpar" com fundo vermelho escuro (#681616), texto claro, e efeito hover que muda para #b32727 com transformação para um leve aumento, criando um efeito de realce.



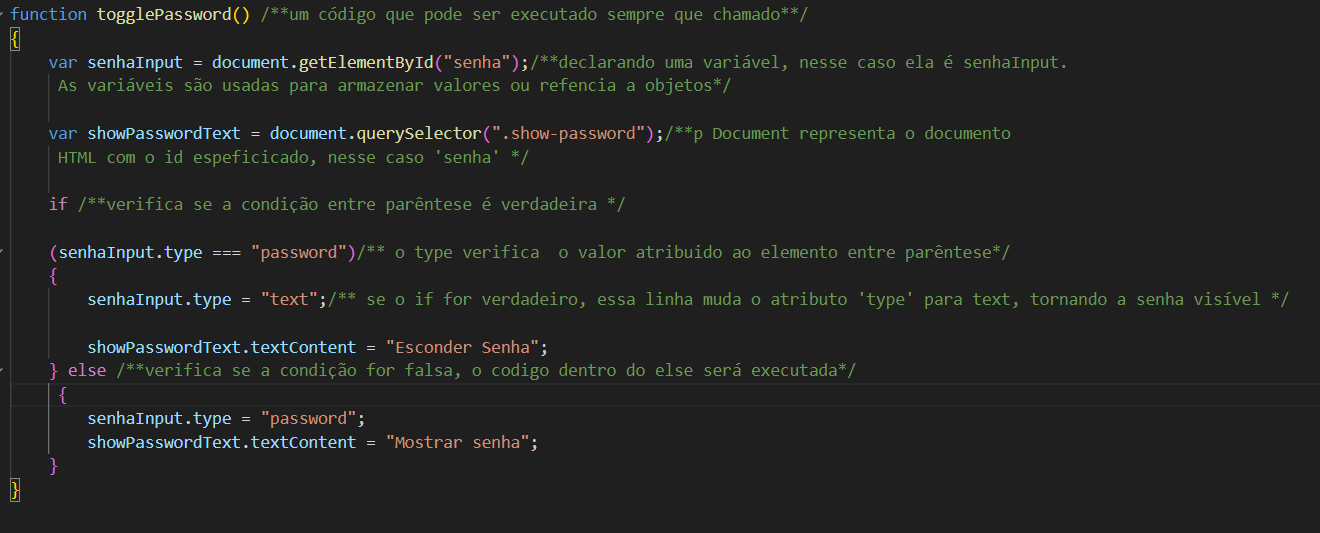
Fonte: Autores (2024)

* **.pronto:** Estiliza o botão "Enviar" de forma semelhante, mas com fundo preto e efeito hover que aumenta o botão e o deixa mais claro para realçar a interação.

**RESULTADO:**



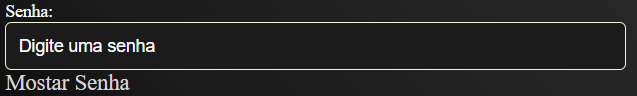
Fonte: Autores (2024)

**JAVASCRIPT**

Fonte: Autores (2024)

* **function togglePassword():** Define uma função chamada togglePassword, que é executada sempre que é chamada. A função alterna a visibilidade da senha no campo de entrada.
* **var senhaInput = document.getElementById("senha");:** Cria uma variável chamada senhaInput que armazena a referência ao elemento HTML com o id="senha". Essa variável é usada para manipular diretamente o campo de entrada da senha.
* **var showPasswordText = document.querySelector(".show-password");:** Cria uma variável chamada showPasswordText que armazena a referência ao elemento HTML com a classe show-password. Esse elemento exibe o texto que informa o usuário sobre a visibilidade da senha.
* **if (senhaInput.type === "password"):** Inicia uma estrutura de decisão if para verificar se o atributo type do elemento senhaInput é igual a "password". Essa condição determina se a senha está atualmente oculta.
* **senhaInput.type = "text";:** Se a condição if for verdadeira, altera o tipo do campo de password para text, tornando a senha visível.
* **showPasswordText.textContent = "Esconder Senha";:** Altera o texto do elemento showPasswordText para "Esconder Senha", informando ao usuário que a senha está visível e pode ser escondida.
* **else:** Define uma ação alternativa que será executada se a condição if for falsa (ou seja, se o campo de senha já estiver visível).
* **senhaInput.type = "password";:** Redefine o tipo do campo de text para password, ocultando a senha novamente.
* **showPasswordText.textContent = "Mostrar senha";:** Atualiza o texto do elemento showPasswordText para "Mostrar senha", indicando que o usuário pode clicar para ver a senha.

**RESULTADO:**

****

Fonte: Autores (2024)

## 2.6 Desenvolvimento do Personagem Tef

O personagem Tef foi concebido como o mascote interativo do site Yourself, representando a união entre tecnologia e empatia. Tef é um gato fantasma preto, com tons de roxo em sua pelagem que brilham sutilmente, refletindo um mistério acolhedor e uma presença mágica. Sua aparência foi cuidadosamente projetada para criar uma sensação de conexão emocional e conforto, garantindo que ele fosse cativante e amigável a todos os públicos.

## 2.7 Personalidade e Função no Site

Tef é mais do que um guia; ele atua como um companheiro emocional e cultural do usuário. Desde o primeiro momento de interação, Tef pergunta como o usuário está se sentindo, oferecendo três opções principais: feliz, triste ou entediado. Com base na escolha, ele adapta sua expressão facial e postura para refletir a emoção do usuário, além de sugerir mídias personalizadas para aquele momento.

Por exemplo:

Se o usuário estiver feliz, Tef exibe brilhos radiantes, e diz:

"Se sentir feliz é importante. Essas indicações vão garantir que esse sorriso continue!"

Se o usuário estiver triste, ele apresenta uma expressão de tristeza, com lágrimas nos olhos, e diz:

"Você deve estar passando por um momento chato... Essas indicações devem ajudar!"

Se o usuário estiver entediado, Tef com uma expressão apática e olhar entendido, comenta:

"Estar entediado é sempre angustiante, mas estou aqui pra indicar algo para sair desse tédio."

Essas falas foram criadas para gerar empatia e transmitir apoio, enquanto as sugestões de filmes, séries, livros e músicas são cuidadosamente escolhidas para atender ao estado emocional indicado.

## 2.8 Tef como Elemento Motivador e Curador

Além de refletir as emoções dos usuários, Tef desempenha o papel de curador de conteúdos. Ele sugere mídias que não apenas combinam com o estado emocional atual, mas que também podem influenciar positivamente o humor e proporcionar uma experiência enriquecedora. Por exemplo:

Para felicidade, Tef pode recomendar comédias leves, músicas animadas ou livros inspiradores, garantindo que o bom humor do usuário seja mantido.

Para tristeza, as indicações incluem histórias reconfortantes, músicas tranquilas ou séries que proporcionam uma sensação de acolhimento.

Para tédio, Tef sugere conteúdos dinâmicos, com histórias envolventes e músicas ou filmes que despertam interesse e energia.

## 2.9 A Estética e o Simbolismo de Tef

A escolha do design de Tef como um gato fantasma carrega significados que se alinham com a proposta do Yourself. O gato simboliza intuição e conforto, enquanto a ideia de ser um fantasma reforça a sensação de que ele está sempre presente para ajudar. Os tons de roxo representam criatividade e calma, promovendo uma atmosfera tranquila e inspiradora.

## 2.10 Impacto na Experiência do Usuário

Tef é a personificação do objetivo central do Yourself: transformar a interação digital em uma experiência humanizada e acolhedora. Ao acompanhar as emoções do usuário e oferecer conteúdos personalizados, ele cria um vínculo emocional que vai além do simples uso da plataforma. Sua presença amigável e sua curadoria cuidadosa de mídias ajudam a promover bem-estar, motivação e momentos de conexão pessoal.



# 3. CONCLUSÃO

Ao longo deste estudo, exploramos os princípios que fundamentam o funcionamento do Yourself, destacando a importância da coleta de dados emocionais, do uso de algoritmos computacionais e de técnicas de análise de comportamento na recomendação personalizada de filmes, séries, livros e músicas.

Com base na revisão realizada, podemos concluir que o Yourself se apresenta como uma ferramenta interdisciplinar, que combina áreas como a psicologia, a ciência da computação e a curadoria de conteúdo cultural.

O Yourself tem o potencial de aperfeiçoar ainda mais a maneira como as emoções influenciam a escolha de conteúdos de entretenimento e bem-estar. Além disso, a constante evolução da inteligência artificial e do aprendizado de máquina promete ampliar o impacto positivo desta ferramenta, tornando-a ainda mais útil para atender às necessidades de um público diversificado.

Por fim, é fundamental reconhecer o papel que o Yourself desempenha em nossa sociedade ao proporcionar um momento de autocuidado e reflexão por meio do entretenimento. Ele não apenas oferece opções culturais personalizadas, mas também promove um melhor entendimento do estado emocional de seus usuários, contribuindo para decisões mais conscientes no dia a dia.

# 4. REFERÊNCIA

“**HTML Reference**.” *W3Schools*, https://www.w3schools.com/tags/. Accessed 19 November 2024.

“**Elementos HTML** - HTML: Linguagem de Marcação de Hipertexto | MDN.” *MDN Web Docs*, 3 December 2023, https://developer.mozilla.org/pt-BR/docs/Web/HTML/Element. Accessed 19 November 2024.

“**Tags HTML**: as principais tags para criar sua página HTML.” *Homehost*, 13 January 2024, https://www.homehost.com.br/blog/tutoriais/tags-html/. Accessed 19 November 2024.

“**Custom code in head and body tags – Webflow Help Center.**” *Webflow Help Center*, 13 November 2024, https://help.webflow.com/hc/en-us/articles/33961357265299-Custom-code-in-head-and-body-tags#supported-code. Accessed 19 November 2024.

**Thomas, Jeremy. “Tags** | Bulma: Free, open source, and modern CSS framework based on Flexbox.” *Bulma CSS*, https://bulma.io/documentation/elements/tag/. Accessed 19 November 2024.

“**HTML JavaScript.**” *W3Schools*, https://www.w3schools.com/html/html\_scripts.asp. Accessed 19 November 2024.

“**Javascript Tags 101**: A Simple (Yes, Simple) Guide.” *Zendata*, https://www.zendata.dev/post/javascript-tags-101-a-simple-yes-simple-guide. Accessed 19 November 2024.

“**<script> - HTML: Linguagem de Marcação de Hipertexto** | MDN.” *MDN Web Docs*, 29 July 2024, https://developer.mozilla.org/pt-BR/docs/Web/HTML/Element/script. Accessed 19 November 2024.