

```
<!DOCTYPE html>

<html lang="pt-BR">

<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>CineBusca - Informações de Filmes</title>

  <script src="https://cdn.tailwindcss.com"></script>

  <script src="https://unpkg.com/marked@4.0.0/marked.min.js"></script>

  <script>

    tailwind.config = {

      darkMode: 'class',

      theme: {

        extend: {

          colors: {

            primary: '#5D5CDE',

            secondary: '#4F4EC0',

          }

        }

      }

    }

  </script>

  <style>

    .dark body {

      background-color: #181818;

      color: #e0e0e0;

    }

  </style>

</head>

<body>
```

```

.movie-card {
    transition: transform 0.3s ease, box-shadow 0.3s ease;
}

.movie-card:hover {
    transform: translateY(-5px);
    box-shadow: 0 10px 15px -3px rgba(0, 0, 0, 0.1), 0 4px 6px -2px rgba(0, 0, 0, 0.05);
}

.dark .movie-card {
    background-color: #2a2a2a;
}

.loading-dots:after {
    content: '.';
    animation: dots 1.5s steps(5, end) infinite;
}

@keyframes dots {
    0%, 20% { content: '.'; }
    40% { content: '..'; }
    60% { content: '...'; }
    80%, 100% { content: ''; }
}

</style>

</head>

<body class="bg-white dark:bg-gray-900 text-gray-800 dark:text-gray-200 min-h-

```

screen">

<div class="container mx-auto px-4 py-8 max-w-6xl">

<header class="text-center mb-10">

<h1 class="text-4xl font-bold text-primary mb-2">CineBusca</h1>

<p class="text-xl text-gray-600 dark:text-gray-400">Encontre informações sobre seus filmes favoritos</p>

</header>

<div class="search-container mb-10">

<div class="flex flex-col md:flex-row gap-2">

<input

type="text"

id="movieSearch"

placeholder="Digite o nome de um filme..."

class="flex-grow p-3 border border-gray-300 dark:border-gray-700 rounded-lg text-base focus:outline-none focus:ring-2 focus:ring-primary dark:bg-gray-800"

>

<button

id="searchButton"

class="bg-primary hover:bg-secondary text-white font-medium py-3 px-6 rounded-lg transition-colors min-w-fit"

>

Buscar

</button>

</div>

</div>

<div id="loadingState" class="hidden">

```
<div class="text-center py-10">

  <div class="inline-block animate-spin rounded-full h-12 w-12 border-t-2
border-b-2 border-primary mb-4"></div>

  <p class="text-xl font-medium">Buscando informações<span
class="loading-dots"></span></p>

</div>

</div>

<div id="results" class="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-3 gap-6">

  <!-- Os resultados da pesquisa aparecerão aqui -->

</div>

<div id="movieDetails" class="hidden mt-8 p-6 bg-white dark:bg-gray-800
rounded-xl shadow-lg">

  <!-- Detalhes do filme aparecerão aqui -->

</div>

<div id="errorMessage" class="hidden mt-6 p-4 bg-red-100 dark:bg-red-900 text-
red-700 dark:text-red-200 rounded-lg">

  <!-- Mensagens de erro aparecerão aqui -->

</div>

<footer class="mt-16 text-center text-sm text-gray-500 dark:text-gray-400">

  <p>CineBusca © 2023 - Todas as informações são para fins educacionais</p>

  <p class="mt-2">Este aplicativo usa a API do Poe para buscar informações
sobre filmes através do Claude-3.7-Sonnet.</p>

</footer>

</div>
```

```
<script>

    // Detectar preferência de modo escuro

    if (window.matchMedia && window.matchMedia('(prefers-color-scheme:
dark)').matches) {

        document.documentElement.classList.add('dark');

    }

    window.matchMedia('(prefers-color-scheme: dark)').addEventListener('change',
event => {

        if (event.matches) {

            document.documentElement.classList.add('dark');

        } else {

            document.documentElement.classList.remove('dark');

        }

    });

    // Elementos do DOM

    const searchInput = document.getElementById('movieSearch');
    const searchButton = document.getElementById('searchButton');
    const resultsContainer = document.getElementById('results');
    const movieDetailsContainer = document.getElementById('movieDetails');
    const loadingState = document.getElementById('loadingState');
    const errorMessage = document.getElementById('errorMessage');

    // Manipulador para busca

    searchButton.addEventListener('click', searchMovies);

    searchInput.addEventListener('keypress', (e) => {

        if (e.key === 'Enter') {
```

```

        searchMovies();
    }
});

// ID do manipulador para as respostas do bot
const handlerId = 'movie-search-handler';

// Registrar manipulador para respostas
window.Poe.registerHandler(handlerId, (result) => {
    const response = result.responses[0];

    if (response.status === 'error') {
        showError('Ocorreu um erro ao buscar informações. Por favor, tente novamente.');
```

hideLoading();

return;

```
    }

    if (response.status === 'complete') {
        try {
            displayMovieResults(response.content);
            hideLoading();
        } catch (err) {
            showError('Não foi possível processar os resultados. Por favor, tente novamente.');
```

hideLoading();

console.error('Erro ao processar resultados:', err);

```
    }
}
```

```
}  
});
```

```
function searchMovies() {  
    const query = searchInput.value.trim();  
  
    if (!query) {  
        showError('Por favor, digite o nome de um filme para buscar.');        return;  
    }  
  
    // Limpar resultados anteriores  
    resultsContainer.innerHTML = "";  
    movieDetailsContainer.classList.add('hidden');  
    errorMessage.classList.add('hidden');
```

```
    // Limpar resultados anteriores  
    resultsContainer.innerHTML = "";  
    movieDetailsContainer.classList.add('hidden');  
    errorMessage.classList.add('hidden');
```

```
    // Mostrar carregamento  
    showLoading();
```

```
    // Construir prompt para o Claude  
    const prompt = `@Claude-3.7-Sonnet
```

Busque informações sobre o filme "\${query}". Forneça APENAS um JSON no seguinte formato, sem texto adicional ou formatação de bloco de código:

```
{  
  "results": [  
    {  
      "title": "Título do Filme",  
      "year": "Ano",
```

```

    "director": "Diretor",
    "poster": "Descrição breve do cartaz",
    "plot": "Sinopse curta",
    "rating": "Nota (0-10)",
    "whereToWatch": ["Netflix", "Prime Video", etc.],
    "cast": ["Ator 1", "Ator 2", "Ator 3"]
  },
  ... até 5 filmes mais relevantes ...
]
}

```

Inclua 3 a 5 filmes nos resultados. Se a consulta for muito específica, retorne apenas o filme exato. Se não encontrar o filme, retorne um array vazio.`;

```

// Enviar mensagem para o bot

try {
  window.Poe.sendMessage(prompt, {
    handler: handlerId,
    stream: false,
    openChat: false
  });
} catch (err) {
  showError('Erro ao se comunicar com a API. Por favor, tente novamente.');
```

```

  hideLoading();
  console.error('Erro ao enviar mensagem:', err);
}
}

```



```

function displayMovieResults(content) {

  try {

    // Tentar extrair JSON da resposta

    const jsonMatch = content.match(/\s*\{.*\}/);

    const jsonContent = jsonMatch ? jsonMatch[0] : content;

    // Analisar o JSON

    const data = JSON.parse(jsonContent);

    if (!data.results || data.results.length === 0) {

      resultsContainer.innerHTML = `

        <div class="col-span-full text-center py-10">

          <svg class="w-16 h-16 mx-auto text-gray-400" fill="none"
stroke="currentColor" viewBox="0 0 24 24">

            <path stroke-linecap="round" stroke-linejoin="round" stroke-
width="2" d="M9.172 16.172a4 4 0 015.656 0M9 10h.01M15 10h.01M21 12a9 9 0 11-
18 0 9 9 0 0118 0z"></path>

          </svg>

          <h3 class="mt-4 text-xl font-medium">Nenhum filme
encontrado</h3>

          <p class="mt-2 text-gray-500 dark:text-gray-400">Tente buscar com
palavras-chave diferentes.</p>

        </div>

      `;

      return;

    }

    // Renderizar resultados

    resultsContainer.innerHTML = data.results.map(movie => `

```

```
<div class="movie-card bg-white dark:bg-gray-800 rounded-xl overflow-hidden shadow-md hover:shadow-lg">
```

```
<div class="p-5">
```

```
<h3 class="text-xl font-bold mb-1">${movie.title}</h3>
```

```
<p class="text-gray-500 dark:text-gray-400 mb-3">${movie.year} •  
Direção: ${movie.director}</p>
```

```
<div class="mt-2 mb-4">
```

```
<div class="flex items-center">
```

```
<svg class="w-5 h-5 text-yellow-500" fill="currentColor"  
viewBox="0 0 20 20">
```

```
<path d="M9.049 2.927c.3-.921 1.603-.921 1.902 0l1.07  
3.292a1 1 0 00.95.69h3.462c.969 0 1.371 1.24.588 1.81l-2.8 2.034a1 1 0 00-.364  
1.118l1.07 3.292c.3.921-.755 1.688-1.54 1.118l-2.8 2.034a1 1 0 00-1.175 0l-2.8  
2.034c-.784.57-1.838-.197-1.539-1.118l1.07-3.292a1 1 0 00-.364-1.118l-2.8 2.034c  
-.783-.57-.38-1.81.588-1.81h3.461a1 1 0 00.951-.69l1.07-3.292z"></path>
```

```
</svg>
```

```
<span class="ml-1 font-medium">${movie.rating}/10</span>
```

```
</div>
```

```
</div>
```

```
<p class="text-sm mb-4 line-clamp-3">${movie.plot}</p>
```

```
<div class="mt-4">
```

```
<h4 class="font-medium mb-2">Onde assistir:</h4>
```

```
<div class="flex flex-wrap gap-2">
```

```
  ${movie.whereToWatch.map(platform =>
```

```
    `<span class="px-2 py-1 bg-gray-100 dark:bg-gray-700 rounded  
-full text-xs">${platform}</span>`
```

```
  ).join("")}
```

```
</div>
```

```
</div>
```

```
<button class="view-details mt-5 w-full bg-primary hover:bg-secondary text-white font-medium py-2 px-4 rounded-lg transition-colors"
```

```
data-movie='${JSON.stringify(movie).replace(/'/g, "&#39;")}'>
```

```
Ver detalhes
```

```
</button>
```

```
</div>
```

```
</div>
```

```
`.join(");
```

```
// Adicionar manipuladores de eventos para os botões de detalhes
```

```
document.querySelectorAll('.view-details').forEach(button => {
```

```
button.addEventListener('click', () => {
```

```
const movieData = JSON.parse(button.getAttribute('data-movie'));
```

```
showMovieDetails(movieData);
```

```
});
```

```
});
```

```
} catch (err) {
```

```
console.error('Erro ao exibir resultados:', err);
```

```
showError('Não foi possível processar os resultados. Tente novamente ou refine sua busca.');
```

```
}
```

```
}
```

```
function showMovieDetails(movie) {
```

```
window.scrollTo({
```

```

    top: 0,

    behavior: 'smooth'

  });

movieDetailsContainer.innerHTML = `

  <button id="backToResults" class="mb-6 flex items-center text-primary">

    <svg class="w-5 h-5 mr-1" fill="none" stroke="currentColor" viewBox="0 0
24 24">

      <path stroke-linecap="round" stroke-linejoin="round" stroke-width="2"
d="M15 19l-7-7 7-7"></path>

    </svg>

    Voltar aos resultados

  </button>

  <div class="flex flex-col md:flex-row gap-8">

    <div class="md:w-1/3">

      <div class="bg-gray-200 dark:bg-gray-700 rounded-lg p-4 text-center h-
64 flex items-center justify-center">

        <p class="text-gray-500 dark:text-gray-400
italic">${movie.poster}</p>

      </div>

    </div>

    <div class="md:w-2/3">

      <h2 class="text-3xl font-bold mb-2">${movie.title}</h2>

      <p class="text-xl text-gray-600 dark:text-gray-400 mb-4">${movie.year} •
Direção: ${movie.director}</p>

      <div class="flex items-center mb-6">

```

```
20 20">
    <svg class="w-6 h-6 text-yellow-500" fill="currentColor" viewBox="0 0
```

```
    <path d="M9.049 2.927c.3-.921 1.603-.921 1.902 0l1.07 3.292a1 1
    0 00.95.69h3.462c.969 0 1.371 1.24.588 1.81l-2.8 2.034a1 1 0 00-.364 1.118l1.07
    3.292c.3.921-.755 1.688-1.54 1.118l-2.8-2.034a1 1 0 00-1.175 0l-2.8 2.034c-.784.57-
    1.838-.197-1.539-1.118l1.07-3.292a1 1 0 00-.364-1.118l-2.8-2.034c-.783-.57-.38-
    1.81.588-1.81h3.461a1 1 0 00.951-.69l1.07-3.292z"></path>
```

```
</svg>
```

```
<span class="ml-2 text-lg font-bold">${movie.rating}/10</span>
```

```
</div>
```

```
<div class="mb-6">
```

```
<h3 class="text-xl font-semibold mb-2">Sinopse</h3>
```

```
<p>${movie.plot}</p>
```

```
</div>
```

```
<div class="mb-6">
```

```
<h3 class="text-xl font-semibold mb-2">Elenco</h3>
```

```
<div class="flex flex-wrap gap-2">
```

```
    ${movie.cast.map(actor =>
```

```
        `<span class="px-3 py-1 bg-gray-100 dark:bg-gray-700 rounded-
full">${actor}</span>`
```

```
    ).join(" ")
```

```
</div>
```

```
</div>
```

```
<div>
```

```
<h3 class="text-xl font-semibold mb-2">Onde assistir</h3>
```

```
<div class="flex flex-wrap gap-2">
```

```
    ${movie.whereToWatch.length > 0 ?
```

```

        movie.whereToWatch.map(platform =>
            `

```

```

movieDetailsContainer.classList.remove('hidden');

```

```

resultsContainer.classList.add('hidden');

```

```

document.getElementById('backToResults').addEventListener('click', () => {
    movieDetailsContainer.classList.add('hidden');
    resultsContainer.classList.remove('hidden');
});
}

```

```

function showLoading() {
    loadingState.classList.remove('hidden');
}

```

```

function hideLoading() {
    loadingState.classList.add('hidden');
}

```

```
}
```

```
function showError(message) {  
    errorMessage.textContent = message;  
    errorMessage.classList.remove('hidden');
```

```
  
    setTimeout(() => {  
        errorMessage.classList.add('hidden');  
    }, 5000);  
}
```

```
</script>
```

```
</body>
```

```
</html>
```