

UmTremdeCinematicFMG / catalogodedvds Public

[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Security](#) [Insights](#)

5dde6a5 ↓

[catalogodedvds / index.html](#)

Go to file

...



UmTremdeCinematicFMG Update index.html ✓

5dde6a5 · 13 minutes ago

History

```
1      <!DOCTYPE html>
2      <html lang="pt-BR">
3      <head>
4          <meta charset="UTF-8">
5          <meta name="viewport" content="width=device-width, initial-scale=1.0">
6          <title>Catálogo de DVDs</title>
7
8          <!-- FONTS E ÍCONES -->
9          <link rel="stylesheet" href="https://static.wixstatic.com/media/727424_6576b9d1efb145568d7ad9ae8c14f923~mv2.jpg">
10
11         <!-- FAVICON TEMPORÁRIO -->
12         <!-- <link rel="icon" href="data:,"> -->
13
14         <!-- SE DIVIDIR O CÓDIGO EM 3 PARTES (HTML+CSS+JS), ISSO AQUI ABAIXO FARÁ REFERÊNCIA AO CSS -->
15         <!-- <link rel="stylesheet" href="styles.css" -->
16
17         <!-- ESTILOS -->
18         <style>
19             /* =====
20                 1. VARIÁVEIS GLOBAIS
21                 ===== */
22         :root {
23             --verde-ifmg: #009a44;
24             --verde-escuro: #007236;
25             --cinza-escuro: #333333;
26             --cinza-medio: #666666;
27             --cinza-claro: #f2f2f2;
28             --branco: #ffffff;
29             --azul: #0066cc;
30             --amarelo: #ffcc00;
31             --laranja: #ff9900;
32             --vermelho: #cc0000;
33             --preto: #000000;
34             --sombra: 0 4px 6px rgba(0, 0, 0, 0.1);
35             --transicao: all 0.3s ease;
36         }
37
38         /* =====
39             2. ESTILOS BASE
40             ===== */
41
42         /* RESET E CONFIGURAÇÕES BÁSICAS */
43         body {
44             font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
45             margin: 0;
46             padding: 0;
47             background-color: var(--cinza-claro);
48             color: var(--cinza-escuro);
49             line-height: 1.6;
50         }
51
52         /* CABEÇALHO */
53         header {
54             background-color: var(--verde-ifmg);
55             color: var(--branco);
56             padding: 25px 0;
57             text-align: center;
58             margin-bottom: 30px;
59             box-shadow: var(--sombra);
60             position: relative;
61             overflow: hidden;
```

```
62         transition: var(--transicao);  
63     }  
64  
65     header::after {  
66         content: '';  
67         position: absolute;  
68         bottom: 0;  
69         left: 0;  
70         width: 100%;  
71         height: 5px;  
72         background: linear-gradient(90deg, var(--verde-ifmg), var(--verde-escuro));  
73     }  
74  
75     h1 {  
76         margin: 0;  
77         font-size: 2.5em;  
78         font-weight: 700;  
79         text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.2);  
80         transition: var(--transicao);  
81     }  
82  
83     /* ======  
84     3. COMPONENTES DE CONTROLE  
85     ====== */  
86  
87     /* CONTAINER PRINCIPAL */  
88     .container {  
89         max-width: 1200px;  
90         margin: 0 auto;  
91         padding: 0 20px;  
92     }  
93  
94     /* WRAPPER DOS CONTROLES */  
95     .controls-wrapper {  
96         display: flex;  
97         flex-direction: column;  
98         gap: 20px;  
99         margin-bottom: 30px;  
100        width: 100%;  
101    }  
102  
103    /* BARRA DE BUSCA */  
104    .search-container {  
105        width: 100%;  
106    }  
107  
108    .search-box {  
109        position: relative;  
110    }  
111  
112    .search-box input {  
113        width: 100%;  
114        padding: 12px 15px 12px 40px;  
115        border: 1px solid var(--cinza-medio);  
116        border-radius: 6px;  
117        font-size: 16px;  
118        transition: var(--transicao);  
119        box-shadow: inset 0 1px 3px rgba(0,0,0,0.1);  
120        box-sizing: border-box;  
121    }  
122  
123    .search-box input:focus {  
124        border-color: var(--verde-ifmg);  
125        outline: none;  
126        box-shadow: 0 0 0 3px rgba(0, 154, 68, 0.2);  
127    }  
128  
129    .search-box i {  
130        position: absolute;  
131        left: 15px;  
132        top: 50%;  
133        transform: translateY(-50%);  
134        color: var(--cinza-medio);  
135    }  
136  
137    /* CONTROLES SECUNDÁRIOS */
```

```
138     .secondary-controls {
139         display: flex;
140         align-items: center;
141         gap: 20px;
142         flex-wrap: wrap;
143         width: 100%;
144     }
145
146     /* ORDENAÇÃO */
147     .sort-container {
148         min-width: 200px;
149         max-width: 300px;
150         margin: 0;
151     }
152
153     .sort-box {
154         position: relative;
155     }
156
157     .sort-box i {
158         position: absolute;
159         left: 15px;
160         top: 50%;
161         transform: translateY(-50%);
162         color: var(--cinza-medio);
163         z-index: 1;
164     }
165
166     .sort-box select {
167         width: 100%;
168         padding: 12px 15px 12px 40px;
169         border: 1px solid var(--cinza-medio);
170         border-radius: 6px;
171         font-size: 16px;
172         transition: var(--transicao);
173         box-shadow: inset 0 1px 3px rgba(0,0,0,0.1);
174         appearance: none;
175         background-color: var(--branco);
176         cursor: pointer;
177     }
178
179     .sort-box select:focus {
180         border-color: var(--verde-ifmg);
181         outline: none;
182         box-shadow: 0 0 0 3px rgba(0, 154, 68, 0.2);
183     }
184
185     /* LINHA DE FILTROS */
186     .filters-row {
187         display: flex;
188         gap: 15px;
189         align-items: center;
190         flex-wrap: wrap;
191     }
192
193     /* CONTAINER DOS FILTROS */
194     .filter-container {
195         flex: 1;
196         min-width: 200px;
197         max-width: 250px;
198     }
199
200     /* ESTILO COMUM PARA TODAS AS CAIXAS DE FILTRO */
201     .filter-box {
202         position: relative;
203     }
204
205     .filter-box i {
206         position: absolute;
207         left: 15px;
208         top: 50%;
209         transform: translateY(-50%);
210         color: var(--cinza-medio);
211         z-index: 1;
212     }
```

```
213  
214     .filter-box select {  
215         width: 100%;  
216         padding: 12px 15px 12px 40px;  
217         border: 1px solid var(--cinza-medio);  
218         border-radius: 6px;  
219         font-size: 16px;  
220         transition: var(--transicao);  
221         box-shadow: inset 0 1px 3px rgba(0,0,0,0.1);  
222         appearance: none;  
223         background-color: var(--branco);  
224         cursor: pointer;  
225         background-image: url("data:image/svg+xml; charset=UTF-8,%3csvg xmlns='http://www.w3.org/2000/svg' viewBox='0 0 24 24' fill='none' stroke='var(--cinza-medio)' stroke-width='2px' style='width: 100%; height: 100%; vertical-align: middle; margin-right: 10px;"/>  
226         background-repeat: no-repeat;  
227         background-position: right 15px center;  
228         background-size: 15px;  
229     }  
230  
231     .filter-box select:focus {  
232         border-color: var(--verde-ifmg);  
233         outline: none;  
234         box-shadow: 0 0 0 3px rgba(0, 154, 68, 0.2);  
235     }  
236  
237     /* CONTADOR DE RESULTADOS */  
238     .results-counter {  
239         background-color: var(--verde-ifmg);  
240         color: var(--branco);  
241         padding: 8px 15px;  
242         border-radius: 20px;  
243         font-size: 14px;  
244         box-shadow: var(--sombra);  
245         white-space: nowrap;  
246         margin-left: auto;  
247     }  
248  
249     .results-counter.sem-resultados {  
250         background-color: var(--vermelho);  
251     }  
252  
253     .results-counter.com-resultados {  
254         background-color: var(--verde-ifmg);  
255     }  
256  
257     #filmsCount {  
258         font-weight: bold;  
259         font-size: 16px;  
260         transition: all 0.3s ease;  
261         display: inline-block;  
262     }  
263  
264     #filmsCount.updated {  
265         transform: scale(1.2);  
266         color: var(--amarelo);  
267     }  
268  
269     /* =====  
270        4. FILTROS DE GÊNERO  
271        ===== */  
272  
273     /* CONTAINER DE FILTROS */  
274     .genre-filter-container {  
275         background-color: var(--branco);  
276         padding: 15px;  
277         border-radius: 8px;  
278         box-shadow: var(--sombra);  
279         margin-bottom: 30px;  
280     }  
281  
282     .genre-filter-title {  
283         font-weight: bold;  
284         font-size: 16px;  
285         color: var(--cinza-escuro);  
286         margin-bottom: 10px;  
287         display: flex;  
288         align-items: center;
```

```
289         gap: 8px;
290     }
291
292     .genre-tags {
293         display: flex;
294         flex-wrap: wrap;
295         gap: 8px;
296     }
297
298     .genre-tag {
299         display: inline-block;
300         padding: 6px 12px;
301         background-color: var(--cinza-claro);
302         color: var(--cinza-escuro);
303         border-radius: 20px;
304         font-size: 14px;
305         cursor: pointer;
306         transition: var(--transicao);
307         border: 1px solid var(--cinza-medio);
308     }
309
310     .genre-tag:hover {
311         background-color: var(--verde-ifmg);
312         color: var(--branco);
313         border-color: var(--verde-escuro);
314     }
315
316     .genre-tag.active {
317         background-color: var(--verde-ifmg);
318         color: var(--branco);
319         border-color: var(--verde-escuro);
320         font-weight: bold;
321     }
322
323     /* =====
324      5. GRID DE FILMES
325      ===== */
326
327     /* GRID PRINCIPAL */
328     .film-grid {
329         display: grid;
330         grid-template-columns: repeat(auto-fill, minmax(250px, 1fr));
331         gap: 30px;
332         margin-bottom: 30px;
333     }
334
335     /* CARTÃO DE FILME */
336     .film-card {
337         background-color: var(--branco);
338         border-radius: 8px;
339         overflow: hidden;
340         box-shadow: var(--sombra);
341         transition: var(--transicao);
342         cursor: pointer;
343         position: relative;
344     }
345
346     .film-card:hover {
347         transform: translateY(-5px);
348         box-shadow: 0 10px 20px rgba(0,0,0,0.15);
349     }
350
351     /* POSTER DO FILME */
352     .film-poster-container {
353         width: 100%;
354         height: 350px;
355         position: relative;
356         overflow: hidden;
357         border-bottom: 3px solid var(--verde-ifmg);
358     }
359
360     .film-poster {
361         width: auto;
362         height: 100%;
363         position: absolute;
364         top: 0;
```

```
304     top: 0,
305     transition: transform 0.5s ease;
306   }
307
308   .film-poster-controls {
309     position: absolute;
310     top: 0;
311     left: 0;
312     width: 100%;
313     height: 100%;
314     display: flex;
315     justify-content: space-between;
316     align-items: center;
317     pointer-events: none;
318   }
319
320   .film-poster-control {
321     background-color: rgba(0, 0, 0, 0.5);
322     color: white;
323     border: none;
324     width: 40px;
325     height: 40px;
326     border-radius: 50%;
327     display: flex;
328     align-items: center;
329     justify-content: center;
330     cursor: pointer;
331     pointer-events: auto;
332     margin: 0 10px;
333     transition: var(--transicao);
334     z-index: 2;
335   }
336
337   .film-poster-control:hover {
338     background-color: rgba(0, 0, 0, 0.8);
339     transform: scale(1.1);
340   }
341
342   .film-card:hover .film-poster {
343     opacity: 0.9;
344   }
345
346   /* INFORMAÇÕES DO FILME */
347   .film-info {
348     padding: 15px;
349   }
350
351   .film-title {
352     font-weight: bold;
353     margin: 0 0 5px 0;
354     color: var(--verde-escuro);
355     font-size: 16px;
356     line-height: 1.3;
357     display: -webkit-box;
358     -webkit-line-clamp: 2;
359     -webkit-box-orient: vertical;
360     overflow: hidden;
361     min-height: 2.6em;
362   }
363
364   .film-meta {
365     display: flex;
366     justify-content: space-between;
367     font-size: 14px;
368     color: var(--cinza-medio);
369   }
370
371   .film-badge {
372     position: absolute;
373     top: 10px;
374     right: 10px;
375     background-color: transparent;
376     padding: 0;
377     z-index: 1;
378   }
```

```
440      /* CLASSIFICAÇÕES */
441      .classification {
442          display: inline-flex;
443          align-items: center;
444          justify-content: center;
445          width: 24px;
446          height: 24px;
447          border-radius: 4px;
448          font-weight: bold;
449          font-size: 12px;
450          margin-right: 10px;
451          color: white;
452          border: 1px solid rgba(0,0,0,0.1);
453      }
454
455      .classification.L {
456          background-color: var(--verde-ifmg);
457      }
458
459      .classification.ten {
460          background-color: var(--azul);
461      }
462
463      .classification.twelve {
464          background-color: var(--amarelo);
465          color: var(--cinza-escuro);
466      }
467
468      .classification.fourteen {
469          background-color: var(--laranja);
470      }
471
472      .classification.sixteen {
473          background-color: var(--vermelho);
474      }
475
476      .classification.eighteen {
477          background-color: var(--preto);
478      }
479
480      /* =====
481          6. MODAL E DETALHES
482      ===== */
483
484      /* BOTÃO DE DETALHES NO MODAL */
485      .modal-actions {
486          margin-top: 20px;
487          padding-top: 20px;
488          border-top: 1px dashed var(--cinza-medio);
489          text-align: center;
490      }
491
492      .btn-details {
493          display: inline-flex;
494          align-items: center;
495          gap: 8px;
496          padding: 12px 24px;
497          background-color: var(--verde-ifmg);
498          color: var(--branco);
499          text-decoration: none;
500          border-radius: 6px;
501          transition: var(--transicao);
502          font-weight: 500;
503      }
504
505      .btn-details:hover {
506          background-color: var(--verde-escuro);
507          transform: translateY(-2px);
508          box-shadow: 0 4px 8px rgba(0,0,0,0.2);
509      }
510
511      .btn-details i {
512          font-size: 0.9em;
513      }
514
515      /* MODAL BASE */
```

```
516     .modal {
517         display: none;
518         position: fixed;
519         z-index: 1000;
520         left: 0;
521         top: 0;
522         width: 100%;
523         height: 100%;
524         background-color: rgba(0,0,0,0.8);
525         overflow: auto;
526         opacity: 0;
527         transition: opacity 0.3s ease;
528     }
529
530     .modal.show {
531         opacity: 1;
532     }
533
534     /* CONTEÚDO DO MODAL */
535     .modal-content {
536         background-color: var(--branco);
537         margin: 5% auto;
538         padding: 30px;
539         width: 70%;
540         max-width: 800px;
541         border-radius: 8px;
542         box-shadow: 0 5px 15px rgba(0,0,0,0.3);
543         position: relative;
544         transform: translateY(-20px);
545         transition: transform 0.3s ease;
546     }
547
548     .modal.show .modal-content {
549         transform: translateY(0);
550     }
551
552     .close {
553         position: absolute;
554         top: 15px;
555         right: 25px;
556         font-size: 28px;
557         font-weight: bold;
558         color: var(--cinza-medio);
559         cursor: pointer;
560         transition: color 0.2s;
561     }
562
563     .close:hover {
564         color: var(--verde-escuro);
565     }
566
567     /* POSTER NO MODAL */
568     .modal-poster-container {
569         float: left;
570         width: 200px;
571         height: 280px;
572         position: relative;
573         overflow: hidden;
574         margin-right: 20px;
575         margin-bottom: 20px;
576         border: 3px solid var(--verde-ifmg);
577         border-radius: 4px;
578         box-shadow: var(--sombra);
579     }
580
581     .modal-poster {
582         width: auto;
583         height: 100%;
584         position: absolute;
585         top: 0;
586         transition: transform 0.5s ease;
587     }
588
589     .modal-poster-controls {
590         position: absolute;
```

```
591     top: 0;
592     left: 0;
593     width: 100%;
594     height: 100%;
595     display: flex;
596     justify-content: space-between;
597     align-items: center;
598     pointer-events: none;
599 }
600
601 .modal-poster-control {
602     background-color: rgba(0, 0, 0, 0.5);
603     color: white;
604     border: none;
605     width: 30px;
606     height: 30px;
607     border-radius: 50%;
608     display: flex;
609     align-items: center;
610     justify-content: center;
611     cursor: pointer;
612     pointer-events: auto;
613     margin: 0 5px;
614     transition: var(--transicao);
615     z-index: 2;
616 }
617
618 .modal-poster-control:hover {
619     background-color: rgba(0, 0, 0, 0.8);
620     transform: scale(1.1);
621 }
622
623 /* DETALHES DO MODAL */
624 .modal-title {
625     color: var(--verde-escuro);
626     margin-top: 0;
627     border-bottom: 2px solid var(--cinza-claro);
628     padding-bottom: 10px;
629     margin-bottom: 15px;
630     font-size: 24px;
631 }
632
633 .modal-details {
634     margin-bottom: 20px;
635 }
636
637 .modal-details p {
638     margin: 8px 0;
639     font-size: 15px;
640     line-height: 1.5;
641 }
642
643 .modal-details strong {
644     color: var(--cinza-escuro);
645 }
646
647 /* SINOPSE E TEMAS */
648 .modal-synopsis {
649     clear: both;
650     line-height: 1.6;
651 }
652
653 .modal-synopsis h3 {
654     color: var(--verde-escuro);
655     margin-bottom: 10px;
656 }
657
658 .modal-synopsis p {
659     font-size: 15px;
660     line-height: 1.6;
661 }
662
663 .modal-themes {
664     margin: 15px 0;
665     padding: 15px;
666     background-color: var(--cinza-claro);
```

```
667         border-radius: 6px;
668     }
669
670     .modal-themes h3 {
671         color: var(--verde-escuro);
672         margin-top: 0;
673         margin-bottom: 10px;
674     }
675
676     .modal-themes .theme-tag {
677         display: inline-block;
678         background-color: var(--verde-ifmg);
679         color: white;
680         padding: 4px 8px;
681         border-radius: 4px;
682         margin-right: 5px;
683         margin-bottom: 5px;
684         font-size: 13px;
685     }
686
687     /* INFORMAÇÕES ADICIONAIS */
688     .modal-additional {
689         margin-top: 20px;
690         padding-top: 15px;
691         border-top: 1px dashed var(--cinza-medio);
692     }
693
694     .modal-additional h3 {
695         color: var(--verde-escuro);
696         margin-top: 0;
697         margin-bottom: 10px;
698     }
699
700     .modal-additional p {
701         margin: 8px 0;
702         font-size: 15px;
703         line-height: 1.5;
704     }
705
706     /* =====
707         7. ELEMENTOS DE PAGINAÇÃO
708         ===== */
709
710     .pagination {
711         display: flex;
712         justify-content: center;
713         gap: 10px;
714         margin-bottom: 30px;
715         flex-wrap: wrap;
716     }
717
718     .pagination button {
719         padding: 8px 12px;
720         border: 1px solid var(--cinza-medio);
721         border-radius: 4px;
722         background-color: var(--branco);
723         cursor: pointer;
724         transition: var(--transicao);
725         min-width: 40px;
726     }
727
728     .pagination button:hover {
729         background-color: var(--cinza-claro);
730     }
731
732     .pagination button.active {
733         background-color: var(--verde-ifmg);
734         color: var(--branco);
735         border-color: var(--verde-escuro);
736     }
737
738     .pagination span {
739         padding: 8px 12px;
740         display: flex;
741         align-items: center;
742     }
```

```
142
143
144     }
145
146     /* =====
147      8. RODAPÉ
148      ===== */
149
150     footer {
151         background-color: var(--verde-escuro);
152         color: var(--branco);
153         text-align: center;
154         padding: 20px;
155         margin-top: 30px;
156         box-shadow: 0 -2px 5px rgba(0,0,0,0.1);
157         transition: var(--transicao);
158         cursor: pointer;
159     }
160
161     footer:hover {
162         background-color: var(--vermelho);
163         transform: scale(1.01);
164     }
165
166     footer p {
167         margin: 0;
168         transition: var(--transicao);
169     }
170
171     footer:hover p {
172         transform: scale(1.02);
173     }
174
175     /* =====
176      9. UTILITÁRIOS E ESTADOS
177      ===== */
178
179     /* LOADING E SKELETON */
180     .loading {
181         text-align: center;
182         padding: 40px;
183         font-size: 1.2em;
184         color: var(--cinza-medio);
185     }
186
187     .skeleton {
188         background-color: #e0e0e0;
189         border-radius: 4px;
190         overflow: hidden;
191         position: relative;
192     }
193
194     .skeleton::after {
195         content: '';
196         position: absolute;
197         top: 0;
198         left: 0;
199         right: 0;
200         bottom: 0;
201         background: linear-gradient(90deg, transparent, rgba(255,255,255,0.5), transparent);
202         animation: shimmer 1.5s infinite;
203     }
204
205     /* ESTADOS SEM RESULTADO */
206     .no-results {
207         grid-column: 1 / -1;
208         text-align: center;
209         padding: 40px;
210         background: var(--branco);
211         border-radius: 8px;
212         box-shadow: var(--sombra);
213     }
214
215     .no-results i {
216         font-size: 40px;
217         color: var(--cinza-medio);
218         margin-bottom: 15px;
219     }
```

```
818  
819     .no-results p {  
820         margin: 0;  
821         font-size: 16px;  
822     }  
823  
824     /* ANIMAÇÕES */  
825     @keyframes shimmer {  
826         0% { transform: translateX(-100%); }  
827         100% { transform: translateX(100%); }  
828     }  
829  
830     /* ====== */  
831     10. MEDIA QUERIES  
832     ====== */  
833  
834     @media (max-width: 900px) {  
835         .modal-content {  
836             width: 90%;  
837             padding: 20px;  
838         }  
839  
840         .modal-poster-container {  
841             float: none;  
842             width: 100%;  
843             height: auto;  
844             max-height: 400px;  
845             margin-right: 0;  
846             margin-bottom: 15px;  
847         }  
848  
849         .film-grid {  
850             grid-template-columns: repeat(auto-fill, minmax(200px, 1fr));  
851             gap: 20px;  
852         }  
853  
854         .film-poster-container {  
855             height: 280px;  
856         }  
857     }  
858  
859     @media (max-width: 768px) {  
860         .secondary-controls {  
861             flex-direction: column;  
862             align-items: stretch;  
863             width: 100%;  
864         }  
865  
866         .sort-container {  
867             max-width: none;  
868             width: 100%;  
869         }  
870  
871         .results-counter {  
872             text-align: center;  
873         }  
874     }  
875  
876     @media (max-width: 600px) {  
877         h1 {  
878             font-size: 2em;  
879         }  
880  
881         .film-grid {  
882             grid-template-columns: repeat(auto-fill, minmax(150px, 1fr));  
883             gap: 15px;  
884         }  
885  
886         .film-poster-container {  
887             height: 220px;  
888         }  
889  
890         .film-title {  
891             font-size: 14px;  
892         }  
893
```

```
894     .film-meta {
895         font-size: 12px;
896     }
897
898     .pagination {
899         gap: 5px;
900     }
901
902     .pagination button,
903     .pagination span {
904         padding: 6px 8px;
905         min-width: 30px;
906     }
907 }
908
909 /* RESPONSIVIDADE */
910 @media (max-width: 768px) {
911     .filters-row {
912         flex-direction: column;
913         gap: 10px;
914     }
915
916     .filter-container {
917         width: 100%;
918         max-width: none;
919     }
920
921     .results-counter {
922         margin-left: 0;
923         text-align: center;
924         width: 100%;
925     }
926 }
927 </style>
928 </head>
929
930 <body>
931     
932     <header>
933         <div class="container">
934             <h1><i class="fas fa-film"></i> Catálogo de DVDs</h1>
935         </div>
936     </header>
937
938     
939     <main class="container">
940         
941         <div class="controls-wrapper">
942             
943             <div class="search-container">
944                 <div class="search-box">
945                     <i class="fas fa-search"></i>
946                     <input
947                         type="text"
948                         id="searchInput"
949                         placeholder="Buscar filmes por título, diretor, elenco, sinopse...">
950                 >
951             </div>
952         </div>
953
954         
955         <div class="filters-row">
956             
957             <div class="filter-container">
958                 <div class="filter-box">
959                     <i class="fas fa-film"></i>
960                     <select id="genreSelect">
961                         <option value="">Todos os Gêneros</option>
962                     </select>
963                 </div>
964             </div>
965
966             
967             <div class="filter-container">
968                 <div class="filter-box">
```

```
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
```

<i class="fas fa-user-shield"></i>

<select id="classificationSelect">

- <option value="">Todas as Classificações</option>
- <option value="L">Livre</option>
- <option value="10">10 anos</option>
- <option value="12">12 anos</option>
- <option value="14">14 anos</option>
- <option value="16">16 anos</option>
- <option value="18">18 anos</option>

</select>

</div>

</div>

<!-- ORDENAÇÃO -->

<div class="filter-container">

<div class="filter-box">

<i class="fas fa-sort"></i>

<select id="sortSelect">

- <option value="title-asc">Título (A-Z)</option>
- <option value="title-desc">Título (Z-A)</option>
- <option value="year-asc">Ano (Antigo-Novo)</option>
- <option value="year-desc">Ano (Novo-Antigo)</option>
- <option value="classification-asc">Classificação (Menor-Maior)</option>
- <option value="classification-desc">Classificação (Maior-Menor)</option>

</select>

</div>

</div>

<!-- CONTADOR DE RESULTADOS -->

<div class="results-counter">

0 filmes encontrados

</div>

</div>

<!-- LOADING STATE -->

<div id="loadingMessage" class="loading">

<div class="skeleton" style="width: 100%; height: 300px; margin-bottom: 20px;"></div>

<div class="skeleton" style="width: 80%; height: 20px; margin: 0 auto 10px;"></div>

<div class="skeleton" style="width: 60%; height: 20px; margin: 0 auto;"></div>

</div>

<!-- GRID DE FILMES -->

<div class="film-grid" id="filmGrid"></div>

<!-- PAGINAÇÃO -->

<div class="pagination" id="pagination"></div>

</main>

<!-- MODAL DE DETALHES DO FILME -->

<div id="filmModal" class="modal">

<div class="modal-content">

×

<div id="modalContent"></div>

</div>

</div>

<!-- RODAPÉ -->

<footer>

<div class="container">

<p><i class="fas fa-train"></i> UM TREM DE CINEMA 2019-2024</p>

</div>

</footer>

<!-- SE DIVIDIR O CÓDIGO EM 3 PARTES (HTML+CSS+JS), ISSO AQUI ABAIXO FARÁ REFERÊNCIA AO JS -->

<!-- SCRIPTS -->

<!-- <script src="script.js"></script> -->

<!-- SCRIPTS -->

<script>

```
/*
=====
1. VARIÁVEIS E CONFIGURAÇÕES GLOBAIS
===== */

let allFilms = []; // ARMAZENA TODOS OS FILMES DO CATÁLOGO
let currentFilms = []; // ARMAZENA OS FILMES FILTRADOS ATUALMENTE
let currentPage = 1; // PÁGINA ATUAL DA PAGINAÇÃO
```

```

1045     let allGenres = [];           // LISTA DE TODOS OS GÊNEROS ÚNICOS
1046     let selectedGenre = '';      // GÊNERO SELECIONADO ATUALMENTE
1047     let debounceTimer;          // TIMER PARA DEBOUNCE DA BUSCA
1048     const itemsPerPage = 20;     // QUANTIDADE DE FILMES POR PÁGINA
1049
1050     /* =====
1051      2. FUNÇÕES DE UTILIDADE E FORMATAÇÃO
1052      ===== */
1053
1054     // LIMPA E FORMATA CAMPOS DE TEXTO
1055     function cleanField(value) {
1056         if (!value) return '';
1057         return String(value).replace(/^\\"|\"$/g, '').trim();
1058     }
1059
1060     // OBTÉM A CLASSE CSS PARA CLASSIFICAÇÃO INDICATIVA
1061     function getClassificationClass(age) {
1062         if (!age || age <= 0) return 'L';
1063
1064         const ageNum = typeof age === 'string' ? parseInt(age) : age;
1065
1066         switch(ageNum) {
1067             case 10: return 'ten';
1068             case 12: return 'twelve';
1069             case 14: return 'fourteen';
1070             case 16: return 'sixteen';
1071             case 18: return 'eighteen';
1072             default: return 'L';
1073         }
1074     }
1075
1076     // FUNÇÃO PARA GERAR CAMINHOS DE IMAGENS CORRETAMENTE
1077     function getbasePath() {
1078         // Para GitHub Pages, o caminho base será /catalogodedvds/
1079         return '/catalogodedvds/';
1080     }
1081

```

5dde6a5 catalogodedvds / index.html

Code Blame 1901 lines (1629 loc) · 82.8 KB

```

1087         if (!filmData || !filmData.imageName) {
1088             console.log("Usando imagem padrão:", DEFAULT_COVER);
1089             return DEFAULT_COVER;
1090         }
1091
1092         const baseName = filmData.imageName.replace(/\.(jpg|jpeg|png|gif)$/i, '');
1093         const imagePath = `${basePath}capas/${baseName}.jpg`;
1094
1095         console.log("Caminho da imagem gerado:", imagePath);
1096         return imagePath;
1097     }
1098
1099     /* =====
1100      3. FUNÇÕES DE TRANSFORMAÇÃO E ORDENAÇÃO
1101      ===== */
1102
1103
1104     // TRANSFORMA DADOS DO JSON PARA O FORMATO DESEJADO
1105     function transformFilmData(originalFilm) {
1106         console.log("Original imageName:", originalFilm["imageName"]);
1107
1108         // TRATAMENTO ESPECIAL PARA NOTA IMDB
1109         let imdbData = { votantes: '' };
1110         if (originalFilm["nota imdb/votantes"]) {
1111             const [nota, votantes] = String(originalFilm["nota imdb/votantes"]).split('/');
1112             imdbData = { votantes: `${nota}/${votantes} || ''`.trim() };
1113         }
1114
1115         return {
1116             title: cleanField(originalFilm["Título do filme"]),
1117             director: cleanField(originalFilm["Direção"]),
1118             cast: cleanField(originalFilm["Elenco"]),
1119             duration: parseInt(originalFilm["Dur.(')"]) || 0,
1120             genres: cleanField(originalFilm["Gên."])
1121         };
1122     }

```

```

catalogodedvds/index.html at 5dde6a5f7969337c842031b9366850ddfd33c346 · UmTremdeCinemaIFMG/catalogodedvds · GitHub
genre: cleanField(originalFilm["Gênero"]),
year: parseInt(originalFilm["Ano"]) || 0,
imdb: imdbData,
country: cleanField(originalFilm["País"]),
state: cleanField(originalFilm["UF"]),
city: cleanField(originalFilm["cidade"]),
audiodescricao: cleanField(originalFilm["Audiodescrição"]),
closedCaption: cleanField(originalFilm["Closed Caption"]),
synopsis: cleanField(originalFilm["Sinopse"]),
tema: cleanField(originalFilm["tema_ProgramadonaBrasil"])\n
tags: cleanField(originalFilm["tags"]),
website: cleanField(originalFilm["website"]),
portaCurta: cleanField(originalFilm["Porta Curta"]),
festivais: cleanField(originalFilm["festivais"]),
premios: cleanField(originalFilm["premios"]),
legendasOutras: cleanField(originalFilm["legendas_outras"]),
materialOutros: cleanField(originalFilm["material_outros"]),
duracaoFormato: cleanField(originalFilm["duracao FORMATO"]),
nossoAcervo: cleanField(originalFilm["Nosso Acervo"]),
pgm: parseInt(originalFilm["PGM"]) || 0,
filmes: parseInt(originalFilm["Filmes"]) || 0,
dvd: cleanField(originalFilm["Nome do Programa"]),
imageName: cleanField(originalFilm["imageName"]),
classification: parseInt(originalFilm["Classificação Indicativa POR PGM"]) || 0
};

}

// ORDENAÇÃO DOS FILMES
function sortFilms(films, sortOption) {
  const sortedFilms = [...films];

  switch(sortOption) {
    case 'title-asc':
      sortedFilms.sort((a, b) => a.title.localeCompare(b.title));
      break;
    case 'title-desc':
      sortedFilms.sort((a, b) => b.title.localeCompare(a.title));
      break;
    case 'year-asc':
      sortedFilms.sort((a, b) => a.year - b.year);
      break;
    case 'year-desc':
      sortedFilms.sort((a, b) => b.year - a.year);
      break;
    case 'classification-asc':
      sortedFilms.sort((a, b) => a.classification - b.classification);
      break;
    case 'classification-desc':
      sortedFilms.sort((a, b) => b.classification - a.classification);
      break;
  }

  return sortedFilms;
}

/*
=====
4. FUNÇÕES DE FILTRO E BUSCA
=====
*/

// FILTRA E RENDERIZA OS FILMES COM DEBOUNCE
function filterAndRenderFilms() {
  clearTimeout(debounceTimer);
  debounceTimer = setTimeout(() => {
    const searchTerm = document.getElementById('searchInput').value.toLowerCase();
    const sortOption = document.getElementById('sortSelect').value;
    const selectedClassification = document.getElementById('classificationSelect').value;
    const selectedGenre = document.getElementById('genreSelect').value;

    // APLICA TODOS OS FILTROS
    currentFilms = allFilms.filter(film => {
      const matchesSearch =
        film.title.toLowerCase().includes(searchTerm) ||
        (film.director && film.director.toLowerCase().includes(searchTerm)) ||
        (film.cast && film.cast.toLowerCase().includes(searchTerm)) ||
        (film.synopsis && film.synopsis.toLowerCase().includes(searchTerm)) ||
        (film.tema && film.tema.toLowerCase().includes(searchTerm)) ||

```

```

1196         (film.tags && film.tags.toLowerCase().includes(searchTerm)) ||
1197         (film.dvd && film.dvd.toLowerCase().includes(searchTerm));
1198
1199     const matchesGenre = !selectedGenre || film.genre === selectedGenre;
1200
1201     const matchesClassification = !selectedClassification ||
1202         film.classification === parseInt(selectedClassification) ||
1203         (selectedClassification === 'L' && film.classification <= 0);
1204
1205     return matchesSearch && matchesGenre && matchesClassification;
1206   });
1207
1208   // ATUALIZA CONTADOR E APLICA ORDENAÇÃO
1209   updateFilmsCounter();
1210   currentFilms = sortFilms(currentFilms, sortOption);
1211
1212   // RESETA PAGINAÇÃO E RENDERIZA
1213   currentPage = 1;           // Reseta para a primeira página
1214   renderPagination();       // Atualiza a paginação
1215   renderFilms();            // Renderiza os filmes
1216 }, 300);
1217 }
1218
1219 /* =====
1220 5. FUNÇÕES DE RENDERIZAÇÃO DA INTERFACE
1221 ===== */
1222
1223 // ATUALIZA O CONTADOR DE FILMES
1224 function updateFilmsCounter() {
1225   const countElement = document.getElementById('filmsCount');
1226   const counterContainer = document.querySelector('.results-counter');
1227
1228   countElement.classList.add('updated');
1229   setTimeout(() => {
1230     countElement.classList.remove('updated');
1231   }, 300);
1232
1233   countElement.textContent = currentFilms.length;
1234
1235   if (currentFilms.length === 0) {
1236     counterContainer.classList.add('sem-resultados');
1237     counterContainer.classList.remove('com-resultados');
1238   } else {
1239     counterContainer.classList.add('com-resultados');
1240     counterContainer.classList.remove('sem-resultados');
1241   }
1242 }
1243
1244 // INICIALIZA OS SELECTS DE FILTRO
1245 function initializeFilters() {
1246   // PREENCHE O SELECT DE GÊNEROS
1247   const genreSelect = document.getElementById('genreSelect');
1248   genreSelect.innerHTML = '<option value="">Todos os Gêneros</option>';
1249   allGenres.forEach(genre => {
1250     const option = document.createElement('option');
1251     option.value = genre;
1252     option.textContent = genre;
1253     genreSelect.appendChild(option);
1254   });
1255
1256   // PREENCHE O SELECT DE CLASSIFICAÇÃO
1257   const classificationSelect = document.getElementById('classificationSelect');
1258   classificationSelect.innerHTML =
1259     `<option value="">Todas as Classificações</option>
1260     <option value="L">Livre</option>
1261     <option value="10">10 anos</option>
1262     <option value="12">12 anos</option>
1263     <option value="14">14 anos</option>
1264     <option value="16">16 anos</option>
1265     <option value="18">18 anos</option>
1266   `;
1267 }
1268
1269 // CRIA CONTROLES DE NAVEGAÇÃO PARA IMAGENS
1270 function createImageControls(container, image) {
1271   const controls = document.createElement('div'):

```

```

1272
1273     controls.className = container.classList.contains('modal-poster-container') ?
1274         'modal-poster-controls' : 'film-poster-controls';
1275
1276     const leftButton = document.createElement('button');
1277     leftButton.className = container.classList.contains('modal-poster-container') ?
1278         'modal-poster-control' : 'film-poster-control';
1279     leftButton.innerHTML = '<i class="fas fa-chevron-left"></i>';
1280     leftButton.addEventListener('click', (e) => {
1281         e.stopPropagation();
1282         const currentTransform = image.style.transform || 'translateX(0)';
1283         const currentX = parseInt(currentTransform.match(/translateX\(([ -\d]+px)\)/)?.[1] || 0);
1284         const newX = Math.min(currentX + 100, 0);
1285         image.style.transform = `translateX(${newX}px)`;
1286     });
1287
1288     const rightButton = document.createElement('button');
1289     rightButton.className = container.classList.contains('modal-poster-container') ?
1290         'modal-poster-control' : 'film-poster-control';
1291     rightButton.innerHTML = '<i class="fas fa-chevron-right"></i>';
1292     rightButton.addEventListener('click', (e) => {
1293         e.stopPropagation();
1294         const currentTransform = image.style.transform || 'translateX(0)';
1295         const currentX = parseInt(currentTransform.match(/translateX\(([ -\d]+px)\)/)?.[1] || 0);
1296         const containerWidth = container.clientWidth;
1297         const imageWidth = image.clientWidth;
1298         const maxX = containerWidth - imageWidth;
1299         const newX = Math.max(currentX - 100, maxX);
1300         image.style.transform = `translateX(${newX}px)`;
1301     });
1302
1303     controls.appendChild(leftButton);
1304     controls.appendChild(rightButton);
1305     container.appendChild(controls);
1306
1307     setTimeout(() => {
1308         const containerWidth = container.clientWidth;
1309         const imageWidth = image.clientWidth;
1310         if (imageWidth > containerWidth) {
1311             const initialX = containerWidth - imageWidth;
1312             image.style.transform = `translateX(${initialX}px)`;
1313         }
1314     }, 50);
1315 }
1316
1317 // CRIA IMAGEM COMFallback
1318 function createSmartPoster(film) {
1319     const img = new Image();
1320     img.className = 'film-poster';
1321     img.alt = film.title || 'Capa do filme';
1322
1323     img.src = getDvdCover(film);
1324     console.log("Imagen definida como:", img.src);
1325
1326     img.onerror = function() {
1327         console.error("Falha ao carregar:", this.src);
1328         this.src = 'capas/progbrasil.png';
1329     };
1330
1331     return img;
1332 }
1333
1334 /* =====
1335    6. FUNÇÕES DE RENDERIZAÇÃO DE FILMES E CARDS
1336 ===== */
1337
1338 // RENDERIZA GRID DE FILMES
1339 function renderFilms() {
1340     const filmGrid = document.getElementById('filmGrid');
1341     const startIndex = (currentPage - 1) * itemsPerPage;
1342     const endIndex = startIndex + itemsPerPage;
1343     const filmsToShow = currentFilms.slice(startIndex, endIndex);
1344
1345     if (filmsToShow.length === 0) {
1346         filmGrid.innerHTML =
1347             <div class="no-results">

```

```

catalogodedvds/index.html at 5dde6a5f7969337c842031b9366850ddfd33c346 · UmTremdeCinemaIFMG/catalogodedvds · GitHub
    <i class="fas fa-search"></i>
        <p>Nenhum filme encontrado com os critérios selecionados.</p>
    </div>;
    return;
}

filmGrid.innerHTML = filmsToShow.map(film => `
    <div class="film-card" onclick="openModal(${JSON.stringify(film).replace(/"/g, '"')})">
        <div class="film-poster-container">
            
        </div>
        <div class="film-badge">
            <span class="classification ${getClassificationClass(film.classification)}">
                ${film.classification <= 0 ? 'L' : film.classification}
            </span>
        </div>
        <div class="film-info">
            <h3 class="film-title">${film.title}</h3>
            <div class="film-meta">
                <span>${film.year || 'Ano N/A'}</span>
                <span>${film.duration ? film.duration + ' min' : 'Duração N/A'}</span>
            </div>
        </div>
    `).join('');

renderPagination();
}

filmsToShow.forEach(film => {
    const filmCard = document.createElement('div');
    filmCard.className = 'film-card';

    // BADGE DE CLASSIFICAÇÃO INDICATIVA
    if (film.classification !== undefined) {
        const classificationClass = getClassificationClass(film.classification);
        const classificationText = film.classification <= 0 ? 'L' : film.classification;

        const classificationBadge = document.createElement('div');
        classificationBadge.className = 'film-badge';
        classificationBadge.innerHTML =
            <span class="classification ${classificationClass}">${classificationText}</span>
        ;
        filmCard.appendChild(classificationBadge);
    }

    // CONTAINER DO POSTER
    const posterContainer = document.createElement('div');
    posterContainer.className = 'film-poster-container';

    const posterImg = createSmartPoster(film);
    posterImg.onload = function() {
        createImageControls(posterContainer, posterImg);
    };

    posterContainer.appendChild(posterImg);

    // INFORMAÇÕES DO FILME
    const filmInfo = document.createElement('div');
    filmInfo.className = 'film-info';

    const title = document.createElement('h3');
    title.className = 'film-title';
    title.textContent = film.title;

    const meta = document.createElement('div');
    meta.className = 'film-meta';

    const duration = document.createElement('span');
    duration.textContent = film.duration ? `${film.duration} min` : '';

```

```
1423
1424     const year = document.createElement('span');
1425     year.textContent = film.year || '';
1426
1427     meta.appendChild(duration);
1428     meta.appendChild(year);
1429
1430     filmInfo.appendChild(title);
1431     filmInfo.appendChild(meta);
1432
1433     filmCard.appendChild(posterContainer);
1434     filmCard.appendChild(filmInfo);
1435
1436     filmCard.addEventListener('click', () => openModal(film));
1437     filmGrid.appendChild(filmCard);
1438 });
1439 }
1440
1441 // RESETA TODOS OS FILTROS
1442 function resetFilters() {
1443     // LIMPA CAMPO DE BUSCA
1444     document.getElementById('searchInput').value = '';
1445
1446     // RESETA SELECTS
1447     document.getElementById('genreSelect').value = '';
1448     document.getElementById('classificationSelect').value = '';
1449     document.getElementById('sortSelect').value = 'title-asc';
1450
1451     // ATUALIZA VISUALIZAÇÃO
1452     filterAndRenderFilms();
1453     updateFilmsCounter();
1454 }
1455
1456 /* =====
1457    7. FUNÇÕES DE CARREGAMENTO DE DADOS
1458 ===== */
1459
1460 // CARREGA DADOS DO CATÁLOGO
1461 async function loadCatalogData() {
1462     try {
1463         await new Promise(resolve => setTimeout(resolve, 800));
1464
1465         const response = await fetch('catalogo.json');
1466         if (!response.ok) throw new Error('Erro ao carregar o arquivo');
1467
1468         const data = await response.json();
1469         allFilms = data.map(transformFilmData);
1470
1471         // EXTRAI GÊNEROS ÚNICOS
1472         allGenres = [...new Set(allFilms.map(film => film.genre)).filter(Boolean)].sort();
1473
1474         // INICIALIZA INTERFACE
1475         initializeFilters();
1476         currentFilms = allFilms;
1477         document.getElementById('loadingMessage').style.display = 'none';
1478         updateFilmsCounter();
1479         renderFilms();
1480
1481     } catch (error) {
1482         console.error('Erro:', error);
1483         document.getElementById('loadingMessage').innerHTML =
1484             `<div class="error-message">
1485                 <i class="fas fa-exclamation-triangle"></i>
1486                 <p>Erro ao carregar o catálogo</p>
1487                 <p>${error.message}</p>
1488                 <button onclick="loadCatalogData()" class="retry-button">
1489                     <i class="fas fa-sync-alt"></i> Tentar novamente
1490                 </button>
1491             </div>
1492             `;
1493     }
1494 }
1495
1496 /* =====
1497    8. FUNÇÕES DO MODAL
1498 ===== */
```

```
catalogodedvds/index.html at 5dde6a5f7969337c842031b9366850ddfd33c346 · UmTremdeCinemaIFMG/catalogodedvds · GitHub
=====
1498
1499
1500 // CRIA LISTA DE TEMAS
1501 function createThemesList(film) {
1502     const themes = [];
1503
1504     if (film.tema) {
1505         themes.push(...film.tema.split(',').map(t => t.trim()));
1506     }
1507
1508     if (film.tags) {
1509         themes.push(...film.tags.split(',').map(t => t.trim()));
1510     }
1511
1512     return [...new Set(themes.filter(t => t))];
1513 }
1514
1515 // ABRE O MODAL COM ANIMAÇÃO
1516 function openModal(film) {
1517     const modal = document.getElementById('filmModal');
1518     const modalContent = document.getElementById('modalContent');
1519
1520     // CRIA URL AMIGÁVEL BASEADA NO TÍTULO DO FILME
1521     const filmSlug = film.title
1522         .toLowerCase()
1523         .normalize('NFD')
1524         .replace(/[\u0300-\u036f]/g, '') // Remove acentos
1525         .replace(/[^a-z0-9]+/g, '-') // Substitui caracteres especiais por hífen
1526         .replace(/^+|-+$g, ''); // Remove hífens do início e fim
1527
1528     // ATUALIZA A URL SEM RECARREGAR A PÁGINA
1529     const baseUrl = '/catalogodedvds'; // Nome do seu repositório
1530     const url = `${baseUrl}#${film.id}`; // Usa hash (#) em vez de path
1531     history.pushState({ filmId: film.dvd }, film.title, url);
1532
1533     const classification = film.classification || 0;
1534     const classificationClass = getClassificationClass(classification);
1535     const classificationText = classification <= 0 ? 'L' : classification;
1536
1537     const themes = createThemesList(film);
1538     const hasThemes = themes.length > 0;
1539
1540     const hasAdditionalInfo = film.audiodescricao || film.closedCaption || film.website ||
1541                             film.portaCurta || film.festivais || film.premios ||
1542                             film.legendasOutras || film.materialOutros;
1543
1544     // Adiciona logs para debug
1545     console.log("Filme:", film);
1546     console.log("Caminho da imagem:", getDvdCover(film));
1547
1548     modalContent.innerHTML =
1549         `

1550             
1556         </div>
1557         <h2 class="modal-title">
1558             <span class="classification ${getClassificationClass(film.classification || 0)}">
1559                 ${film.classification <= 0 ? 'L' : film.classification}
1560             </span>
1561             ${film.title}
1562         </h2>
1563         <div class="modal-details">
1564             ${film.director ? `<p><strong><i class="fas fa-user"></i> Direção:</strong> ${film.director}</p>` : ''}
1565             ${film.cast ? `<p><strong><i class="fas fa-users"></i> Elenco:</strong> ${film.cast}</p>` : ''}
1566             ${film.duration ? `<p><strong><i class="fas fa-clock"></i> Duração:</strong> ${film.duration} min</p>` : ''}
1567             ${film.genre ? `<p><strong><i class="fas fa-tag"></i> Gênero:</strong> ${film.genre}</p>` : ''}
1568             ${film.year ? `<p><strong><i class="fas fa-calendar-alt"></i> Ano:</strong> ${film.year}</p>` : ''}
1569             ${film.imdb.votantes ? `<p><strong><i class="fab fa-imdb"></i> IMDb:</strong> ${film.imdb.votantes}</p>` : ''}
1570             ${film.country ? `<p><strong><i class="fas fa-globe-americas"></i> País:</strong> ${film.country}</p>` : ''}
1571             ${film.state ? `<p><strong><i class="fas fa-map-marker-alt"></i> UF:</strong> ${film.state}</p>` : ''}
1572             ${film.dvd ? `<p><strong><i class="fas fa-compact-disc"></i> DVD:</strong> ${film.dvd}</p>` : ''}
1573         </div>


```

```
1574  
1575      ${hasThemes ? `  
1576          <div class="modal-themes">  
1577              <h3><i class="fas fa-tags"></i> Temas</h3>  
1578              ${themes.map(theme => `<span class="theme-tag">${theme}</span>`).join('')}  
1579          </div>  
1580      ` : '')  
1581  
1582      ${film.synopsis ? `  
1583          <div class="modal-synopsis">  
1584              <h3><i class="fas fa-align-left"></i> Sinopse</h3>  
1585              <p>${film.synopsis}</p>  
1586          </div>  
1587      ` : '')  
1588  
1589      ${hasAdditionalInfo ? `  
1590          <div class="modal-additional">  
1591              <h3><i class="fas fa-info-circle"></i> Informações Adicionais</h3>  
1592              ${film.audiodescricao ? `<p><strong><i class="fas fa-assistive-listening-systems"></i> Audiodescrição:</strong>` : ''}  
1593              ${film.closedCaption ? `<p><strong><i class="fas fa-closed-captioning"></i> Closed Caption:</strong>` : ''}  
1594              ${film.website ? `<p><strong><i class="fas fa-globe"></i> Website:</strong> <a href="${film.website.startsWith('`  
1595                  ${film.portaCurta ? `<p><strong><i class="fas fa-film"></i> Porta Curtas:</strong> <a href="${film.portaCurta.startsWith('`  
1596                  ${film.festivais ? `<p><strong><i class="fas fa-trophy"></i> Festivais:</strong>` : ''}  
1597                  ${film.premios ? `<p><strong><i class="fas fa-award"></i> Prêmios:</strong>` : ''}  
1598                  ${film.legendasOutras ? `<p><strong><i class="fas fa-language"></i> Outras Legendas:</strong>` : ''}  
1599                  ${film.materialOutros ? `<p><strong><i class="fas fa-box-open"></i> Outros Materiais:</strong>` : ''}  
1600          </div>  
1601      ` : '')  
1602  
1603      <!-- NOVO BOTÃO DE DETALHES -->  
1604      <div class="modal-actions">  
1605          <button class="btn-details" title="Página individual em desenvolvimento"  
1606              onclick="alert('Página individual do filme em desenvolvimento!')">  
1607              <i class="fas fa-external-link-alt"></i>  
1608              Ver página completa (Em breve)  
1609          </button>  
1610      </div>  
1611  
1612      `;  
1613  
1614      const modalPosterContainer = modalContent.querySelector('.modal-poster-container');  
1615      const modalPoster = modalContent.querySelector('.modal-poster');  
1616      if (modalPosterContainer && modalPoster) {  
1617          createImageControls(modalPosterContainer, modalPoster);  
1618      }  
1619  
1620      modal.style.display = 'block';  
1621      setTimeout(() => modal.classList.add('show'), 10);  
1622  
1623      document.addEventListener('keydown', handleKeyDown);  
1624  }  
1625  
1626      // FECHA O MODAL  
1627      function closeModal() {  
1628          const modal = document.getElementById('filmModal');  
1629          modal.classList.remove('show');  
1630  
1631          // Atualiza para a URL base correta  
1632          const baseUrl = '/catalogodedvds/'; // Nome do seu repositório  
1633          history.pushState(null, '', baseUrl);  
1634  
1635          setTimeout(() => {  
1636              modal.style.display = 'none';  
1637          }, 300);  
1638  
1639          document.removeEventListener('keydown', handleKeyDown);  
1640      }  
1641  
1642      // HANDLER PARA TECLA ESC  
1643      function handleKeyDown(e) {  
1644          if (e.key === 'Escape') {  
1645              closeModal();  
1646          }  
1647      }  
1648  
1649      /* ===== */
```

```
1650         9. INICIALIZAÇÃO E EVENTOS
1651         ===== */
1652
1653         // CONFIGURA TODOS OS EVENT LISTENERS
1654         function setupEventListeners() {
1655             // EVENTOS DE BUSCA E FILTROS
1656             document.getElementById('searchInput').addEventListener('input', filterAndRenderFilms);
1657             document.getElementById('genreSelect').addEventListener('change', filterAndRenderFilms);
1658             document.getElementById('classificationSelect').addEventListener('change', filterAndRenderFilms);
1659             document.getElementById('sortSelect').addEventListener('change', filterAndRenderFilms);
1660
1661             // EVENTOS DO MODAL
1662             document.querySelector('.close').addEventListener('click', closeModal);
1663             window.addEventListener('click', function(event) {
1664                 if (event.target === document.getElementById('filmModal')) {
1665                     closeModal();
1666                 }
1667             });
1668
1669             // EVENTO PARA O BOTÃO VOLTAR DO NAVEGADOR
1670             window.addEventListener('popstate', function(event) {
1671                 const modal = document.getElementById('filmModal');
1672                 if (modal.style.display === 'block') {
1673                     closeModal();
1674                 }
1675             });
1676
1677             // Verifica hash na URL
1678             window.addEventListener('load', function() {
1679                 const hash = window.location.hash;
1680                 if (hash && hash.startsWith('#filme/')) {
1681                     // TODO: Implementar carregamento do filme específico
1682                     window.location.href = '/catalogodedvds/'; // Por enquanto, redireciona para a página principal
1683                 }
1684             });
1685
1686             // EVENTO DO FOOTER
1687             document.querySelector('footer').addEventListener('click', function() {
1688                 window.location.href = 'https://umtremdecinema.wixsite.com/umtremdecinema';
1689             });
1690         }
1691
1692         // INICIALIZAÇÃO DA APLICAÇÃO
1693         window.addEventListener('DOMContentLoaded', function() {
1694             setupEventListeners();
1695             loadCatalogData();
1696         });
1697
1698         /* =====
1699             FUNÇÕES DE PAGINAÇÃO
1700             ===== */
1701
1702         // FUNÇÃO PARA RENDERIZAR FILMES NA GRID
1703
1704         function renderFilms() {
1705             const filmGrid = document.getElementById('filmGrid');
1706             const startIndex = (currentPage - 1) * itemsPerPage;
1707             const endIndex = startIndex + itemsPerPage;
1708             const filmsToShow = currentFilms.slice(startIndex, endIndex);
1709
1710             if (filmsToShow.length === 0) {
1711                 filmGrid.innerHTML = `
1712                     <div class="no-results">
1713                         <i class="fas fa-search"></i>
1714                         <p>Nenhum filme encontrado com os critérios selecionados.</p>
1715                     </div>`;
1716             }
1717             return;
1718         }
1719
1720         filmGrid.innerHTML = filmsToShow.map(film => `
1721             <div class="film-card" onclick="openModal(${JSON.stringify(film).replace(/\"/g, '"')})">
1722                 <div class="film-poster-container">
1723                     
1727                 </div>
1728             </div>
1729         `);
1730     
```

```

1725         class="film-poster"
1726             onerror="this.onerror=null; this.src='capas/progbrasil.png';"
1727             loading="lazy"
1728         >
1729     </div>
1730     <div class="film-badge">
1731         <span class="classification ${getClassificationClass(film.classification)}>
1732             ${film.classification <= 0 ? 'L' : film.classification}
1733         </span>
1734     </div>
1735     <div class="film-info">
1736         <h3 class="film-title">${film.title}</h3>
1737         <div class="film-meta">
1738             <span>${film.year || 'Ano N/A'}</span>
1739             <span>${film.duration ? film.duration + ' min' : 'Duração N/A'}</span>
1740         </div>
1741     </div>
1742   </div>
1743 `).join('');
1744
1745   renderPagination();
1746 }
1747
1748 // FUNÇÃO PARA RENDERIZAR A PAGINAÇÃO
1749 function renderPagination() {
1750   const totalPages = Math.ceil(currentFilms.length / itemsPerPage);
1751   const pagination = document.getElementById('pagination');
1752
1753   if (totalPages <= 1) {
1754     pagination.innerHTML = '';
1755     return;
1756   }
1757
1758   let paginationHTML = '';
1759
1760   // Botão Anterior
1761   paginationHTML += `
1762     <button onclick="changePage(${currentPage - 1})" ${currentPage === 1 ? 'disabled' : ''}>
1763       <i class="fas fa-chevron-left"></i>
1764     </button>
1765   `;
1766
1767   // Números das páginas
1768   for (let i = 1; i <= totalPages; i++) {
1769     if (i === 1 || i === totalPages || (i >= currentPage - 2 && i <= currentPage + 2)) {
1770       paginationHTML += `
1771         <button onclick="changePage(${i})" class="${i === currentPage ? 'active' : ''}>
1772           ${i}
1773         </button>
1774       `;
1775     } else if (i === currentPage - 3 || i === currentPage + 3) {
1776       paginationHTML += '<span>...</span>';
1777     }
1778   }
1779
1780   // Botão Próximo
1781   paginationHTML += `
1782     <button onclick="changePage(${currentPage + 1})" ${currentPage === totalPages ? 'disabled' : ''}>
1783       <i class="fas fa-chevron-right"></i>
1784     </button>
1785   `;
1786
1787   pagination.innerHTML = paginationHTML;
1788 }
1789
1790 // FUNÇÃO PARA MUDAR DE PÁGINA
1791 function changePage(page) {
1792   const totalPages = Math.ceil(currentFilms.length / itemsPerPage);
1793   if (page < 1 || page > totalPages) return;
1794   currentPage = page;
1795   renderFilms();
1796   window.scrollTo({ top: 0, behavior: 'smooth' });
1797 }
1798
1799 /*  // RENDERIZA A PAGINAÇÃO
1800   function renderPagination() {

```

```
1801 const pagination = document.getElementById('pagination');
1802 pagination.innerHTML = '';
1803
1804 const totalPages = Math.ceil(currentFilms.length / itemsPerPage);
1805 if (totalPages <= 1) return;
1806
1807 // BOTÃO ANTERIOR
1808 const prevButton = document.createElement('button');
1809 prevButton.innerHTML = '<i class="fas fa-chevron-left"></i> Anterior';
1810 prevButton.disabled = currentPage === 1;
1811 prevButton.addEventListener('click', () => {
1812     if (currentPage > 1) {
1813         currentPage--;
1814         renderFilms();
1815         renderPagination();
1816         window.scrollTo({top: 0, behavior: 'smooth'});
1817     }
1818 });
1819 pagination.appendChild(prevButton);
1820
1821 // PÁGINAS
1822 const maxVisiblePages = 5;
1823 let startPage = Math.max(1, currentPage - Math.floor(maxVisiblePages / 2));
1824 let endPage = Math.min(totalPages, startPage + maxVisiblePages - 1);
1825
1826 if (endPage - startPage + 1 < maxVisiblePages) {
1827     startPage = Math.max(1, endPage - maxVisiblePages + 1);
1828 }
1829
1830 // PRIMEIRA PÁGINA
1831 if (startPage > 1) {
1832     const firstButton = document.createElement('button');
1833     firstButton.textContent = '1';
1834     firstButton.addEventListener('click', () => {
1835         currentPage = 1;
1836         renderFilms();
1837         renderPagination();
1838         window.scrollTo({top: 0, behavior: 'smooth'});
1839     });
1840     pagination.appendChild(firstButton);
1841
1842     if (startPage > 2) {
1843         const ellipsis = document.createElement('span');
1844         ellipsis.textContent = '...';
1845         pagination.appendChild(ellipsis);
1846     }
1847 }
1848
1849 // PÁGINAS INTERMEDIÁRIAS
1850 for (let i = startPage; i <= endPage; i++) {
1851     const pageButton = document.createElement('button');
1852     pageButton.textContent = i;
1853     if (i === currentPage) {
1854         pageButton.classList.add('active');
1855     }
1856     pageButton.addEventListener('click', () => {
1857         currentPage = i;
1858         renderFilms();
1859         renderPagination();
1860         window.scrollTo({top: 0, behavior: 'smooth'});
1861     });
1862     pagination.appendChild(pageButton);
1863 }
1864
1865 // ÚLTIMA PÁGINA
1866 if (endPage < totalPages) {
1867     if (endPage < totalPages - 1) {
1868         const ellipsis = document.createElement('span');
1869         ellipsis.textContent = '...';
1870         pagination.appendChild(ellipsis);
1871     }
1872
1873     const lastButton = document.createElement('button');
1874     lastButton.textContent = totalPages;
1875     lastButton.addEventListener('click', () => {
1876         currentPage = totalPages;
1877     });
1878 }
```

```
1877     currentPage = totalPages;
1878     renderFilms();
1879     renderPagination();
1880     window.scrollTo({top: 0, behavior: 'smooth'});
1881   });
1882 
1883   // BOTÃO PRÓXIMO
1884   const nextButton = document.createElement('button');
1885   nextButton.innerHTML = 'Próximo <i class="fas fa-chevron-right"></i>';
1886   nextButton.disabled = currentPage === totalPages;
1887   nextButton.addEventListener('click', () => {
1888     if (currentPage < totalPages) {
1889       currentPage++;
1890       renderFilms();
1891       renderPagination();
1892       window.scrollTo({top: 0, behavior: 'smooth'});
1893     }
1894   });
1895   pagination.appendChild(nextButton);
1896 }
1897 } */
1898 
1899 </script>
1900 </body>
1901 </html>
```