

TECNOLÓGICO NACIONAL DE MÉXICO INSTITUTO TECNOLÓGICO DE TIJUANA

SUBDIRECCIÓN ACADÉMICA DEPARTAMENTO DE SISTEMAS Y COMPUTACIÓN

SEMESTRE:
Agosto - Diciembre 2025

CARRERA:
Ingeniería en Sistemas Computacionales

MATERIA:

Patrones de diseño

TÍTULO ACTIVIDAD:
Examen 3

UNIDAD A EVALUAR:
3

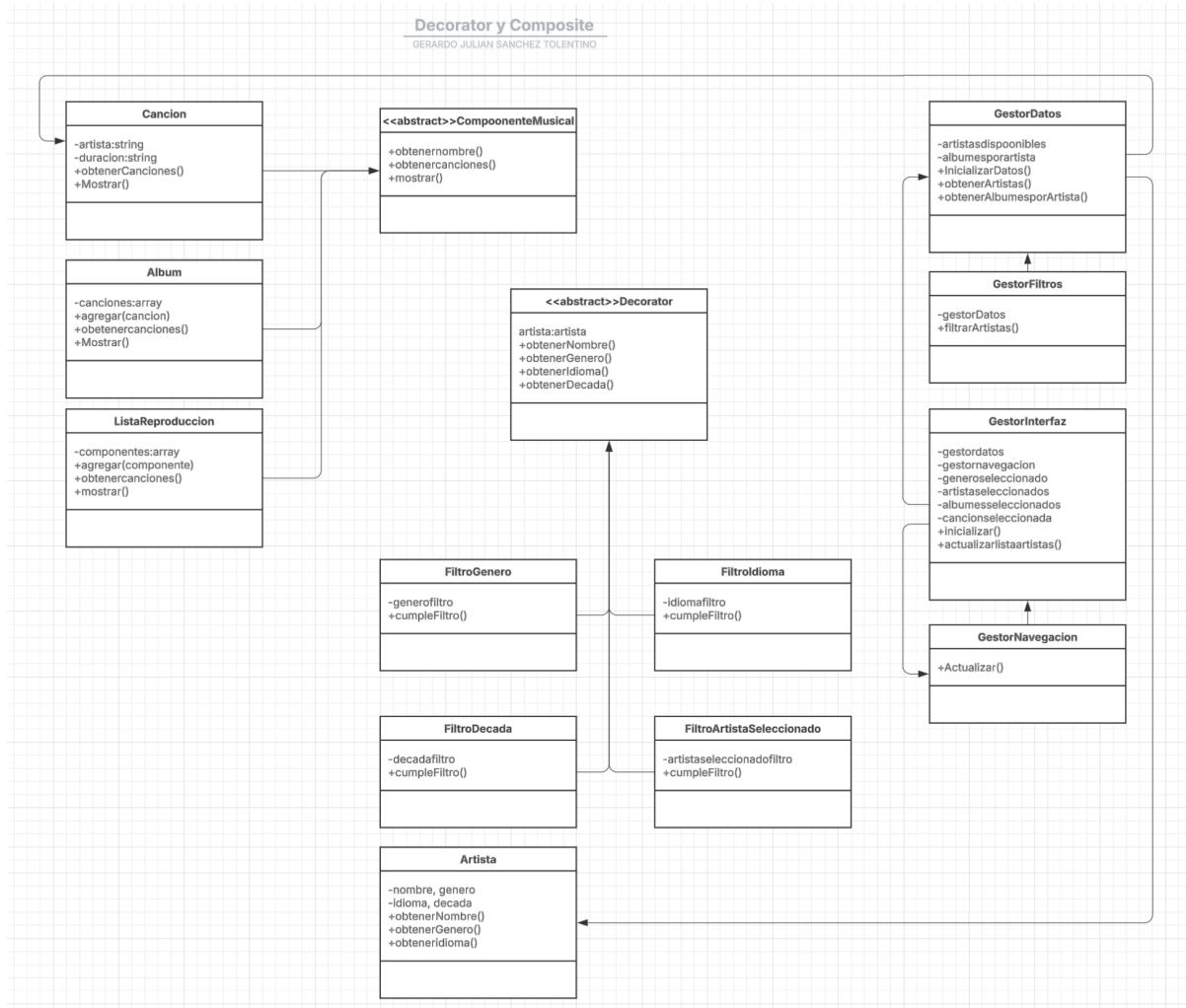
NOMBRE Y NÚMERO DE CONTROL DEL ALUMNO:

Sanchez Tolentino Gerardo Julian 212120248

NOMBRE DEL MAESTRO (A):

Maribel Guerrero Luis

Diagrama UML



Código

[Album.js](#)

```
1  class Album extends ComponenteMusical
2  {
3      constructor(nombre, canciones)
4      {
5          super(nombre);
6          this.canciones = canciones || [];
7      }
8
9      agregar(cancion)
10     {
11         this.canciones.push(cancion);
12     }
13
14     obtenerCanciones()
15     {
16         let todasLasCanciones = [];
17         this.canciones.forEach(cancion =>
18             {
19                 todasLasCanciones = todasLasCanciones.concat(cancion.obtenerCanciones());
20             });
21         return todasLasCanciones;
22     }
23
24     mostrar()
25     {
26         const div = document.createElement('div');
27         div.className = 'album';
28
29         const titulo = document.createElement('div');
30         titulo.className = 'album-titulo';
31         titulo.textContent = `Álbum: ${this.nombre}`;
32         div.appendChild(titulo);
33
34         this.canciones.forEach(cancion =>
35             {
36                 div.appendChild(cancion.mostrar());
37             });
38
39         return div;
40     }
41 }
```

[Artista.js](#)

```
1  class Artista
2  {
3      constructor(nombre, genero, idioma, decada)
4      {
5          this.nombre = nombre;
6          this.genero = genero;
7          this.idioma = idioma;
8          this.decada = decada;
9      }
10
11     obtenerNombre()
12     {
13         return this.nombre;
14     }
15
16     obtenerGenero()
17     {
18         return this.genero;
19     }
20
21     obtenerIdioma()
22     {
23         return this.idioma;
24     }
25
26     obtenerDecada()
27     {
28         return this.decada;
29     }
30 }
```

[Cancion.js](#)

```
1  class Cancion extends ComponenteMusical
2  {
3      constructor(nombre, artista, duracion)
4      {
5          super(nombre);
6          this.artista = artista;
7          this.duracion = duracion;
8      }
9
10     obtenerCanciones()
11     {
12         return [this];
13     }
14
15     mostrar()
16     {
17         const div = document.createElement('div');
18         div.className = 'cancion';
19         div.style.cursor = 'pointer';
20         const artista = this.artista || 'Artista Desconocido';
21         div.textContent = `${this.nombre} - ${artista} (${this.duracion})`;
22
23         div.addEventListener('click', () =>
24         {
25             if (window.gestorInterfaz)
26             {
27                 window.gestorInterfaz.seleccionarCancion(this);
28             }
29         });
30
31         return div;
32     }
33 }
```

[ComponenteMusical.js](#)

```
1  class ComponenteMusical
2  {
3      constructor(nombre)
4      {
5          this.nombre = nombre;
6      }
7
8      obtenerNombre()
9      {
10         return this.nombre;
11     }
12
13     obtenerCanciones()
14     {
15         throw new Error("Método obtenerCanciones debe ser implementado");
16     }
17
18     mostrar()
19     {
20         throw new Error("Método mostrar debe ser implementado");
21     }
22 }
```

[DecoradorFiltro.js](#)

```
1  class DecoradorFiltro
2  {
3      constructor(artista)
4      {
5          this.artista = artista;
6      }
7
8      obtenerNombre()
9      {
10         return this.artista.obtenerNombre();
11     }
12
13     obtenerGenero()
14     {
15         return this.artista.obtenerGenero();
16     }
17
18     obtenerIdioma()
19     {
20         return this.artista.obtenerIdioma();
21     }
22
23     obtenerDecada()
24     {
25         return this.artista.obtenerDecada();
26     }
27
28     cumpleFiltro()
29     {
30         throw new Error("Método cumpleFiltro debe ser implementado");
31     }
32 }
```

FiltroArtistaSeleccionado.js

```
1  class FiltroArtistaSeleccionado extends DecoradorFiltro
2  {
3      constructor(artista, artistasSeleccionados)
4      {
5          super(artista);
6          this.artistasSeleccionados = artistasSeleccionados;
7      }
8
9      cumpleFiltro()
10     {
11         return this.artistasSeleccionados.includes(this.artista.obtenerNombre());
12     }
13 }
```

FiltroDecada.js

```
1  class FiltroDecada extends DecoradorFiltro
2  {
3      constructor(artista, decadaFiltro)
4      {
5          super(artista);
6          this.decadaFiltro = decadaFiltro;
7      }
8
9      cumpleFiltro()
10     {
11         if (!this.decadaFiltro) return true;
12         return this.artista.obtenerDecada() === this.decadaFiltro;
13     }
14 }
```

[FiltroGenero.js](#)

```
1  class FiltroGenero extends DecoradorFiltro
2  {
3      constructor(artista, generoFiltro)
4      {
5          super(artista);
6          this.generoFiltro = generoFiltro;
7      }
8
9      cumpleFiltro()
10     {
11         return this.artista.obtenerGenero() === this.generoFiltro;
12     }
13 }
```

[Filtoldioma.js](#)

```
1  class FiltroIdioma extends DecoradorFiltro
2  {
3      constructor(artista, idiomaFiltro)
4      {
5          super(artista);
6          this.idiomaFiltro = idiomaFiltro;
7      }
8
9      cumpleFiltro()
10     {
11         if (!this.idiomaFiltro) return true;
12         return this.artista.obtenerIdioma() === this.idiomaFiltro;
13     }
14 }
```

[GestorDatos.js](#)

```
1  class GestorDatos
2  {
3      constructor()
4      {
5          this.artistasDisponibles = [];
6          this.albumesPorArtista = {};
7          this.inicializarDatos();
8      }
9
10     inicializarDatos()
11     {
12         this.artistasDisponibles = [
13             // Rock 70s
14             new Artista("The Beatles", "rock", "ingles", "70s"),
15             new Artista("Led Zeppelin", "rock", "ingles", "70s"),
16             new Artista("Pink Floyd", "rock", "ingles", "70s"),
17             new Artista("The Rolling Stones", "rock", "ingles", "70s"),
18             // Rock 80s
19             new Artista("Queen", "rock", "ingles", "80s"),
20             new Artista("AC/DC", "rock", "ingles", "80s"),
21             new Artista("Guns N' Roses", "rock", "ingles", "80s"),
22             new Artista("Joaquín Sabina", "rock", "espanol", "80s"),
23             new Artista("Soda Stereo", "rock", "espanol", "80s"),
24             // Rock 90s
25             new Artista("Nirvana", "rock", "ingles", "90s"),
26             new Artista("Radiohead", "rock", "ingles", "90s"),
27             new Artista("Pearl Jam", "rock", "ingles", "90s"),
28             new Artista("Café Tacvba", "rock", "espanol", "90s"),
29             new Artista("Maná", "rock", "espanol", "90s"),
30             new Artista("Los Fabulosos Cadillacs", "rock", "espanol", "90s"),
31             // Rock 2000
32             new Artista("Linkin Park", "rock", "ingles", "2000"),
33             new Artista("The White Stripes", "rock", "ingles", "2000"),
34             new Artista("Arctic Monkeys", "rock", "ingles", "2000"),
35             new Artista("Juanes", "rock", "espanol", "2000"),
36             new Artista("Molotov", "rock", "espanol", "2000"),
37             new Artista("Zoé", "rock", "espanol", "2000"),
38             // Pop 70s
39             new Artista("ABBA", "pop", "ingles", "70s"),
40             // Pop 80s
41             new Artista("Michael Jackson", "pop", "ingles", "80s"),
42             new Artista("Madonna", "pop", "ingles", "80s"),
43             new Artista("Prince", "pop", "ingles", "80s"),
44             // Pop 90s
45             new Artista("Britney Spears", "pop", "ingles", "90s"),
46             new Artista("Backstreet Boys", "pop", "ingles", "90s"),
47             new Artista("Luis Miguel", "pop", "espanol", "90s"),
48             new Artista("Ricky Martin", "pop", "espanol", "90s"),
49             // Pop 2000
50             new Artista("Shakira", "pop", "espanol", "2000"),
51             new Artista("Enrique Iglesias", "pop", "espanol", "2000"),
52             new Artista("Lady Gaga", "pop", "ingles", "2000"),
53             new Artista("Taylor Swift", "pop", "ingles", "2000"),
54             // Jazz 70s
55             new Artista("Miles Davis", "jazz", "ingles", "70s"),
```

```

56     new Artista("John Coltrane", "jazz", "ingles", "70s"),
57     new Artista("Herbie Hancock", "jazz", "ingles", "70s"),
58     // Jazz 80s
59     new Artista("Pat Metheny", "jazz", "ingles", "80s"),
60     // Jazz 90s
61     new Artista("Diana Krall", "jazz", "ingles", "90s"),
62     // Jazz 2000
63     new Artista("Norah Jones", "jazz", "ingles", "2000"),
64     // Electrónica 80s
65     new Artista("Kraftwerk", "electronica", "ingles", "80s"),
66     // Electrónica 90s
67     new Artista("The Prodigy", "electronica", "ingles", "90s"),
68     new Artista("Chemical Brothers", "electronica", "ingles", "90s"),
69     // Electrónica 2000
70     new Artista("Daft Punk", "electronica", "ingles", "2000"),
71     new Artista("Deadmau5", "electronica", "ingles", "2000"),
72     new Artista("Skrillex", "electronica", "ingles", "2000"),
73     // Clásica 70s
74     new Artista("Mozart", "clasica", "ingles", "70s"),
75     new Artista("Beethoven", "clasica", "ingles", "70s"),
76     new Artista("Bach", "clasica", "ingles", "70s"),
77     // Clásica 80s
78     new Artista("Vivaldi", "clasica", "ingles", "80s"),
79     // Clásica 90s
80     new Artista("Chopin", "clasica", "ingles", "90s"),
81     // Clásica 2000
82     new Artista("Ludovico Einaudi", "clasica", "ingles", "2000")
83 ];
84
85     // Álbumes y canciones
86     this.albumesPorArtista =
87     {
88         "The Beatles": [
89             "Abbey Road": [
90                 new Cancion("Come Together", "The Beatles", "4:20"),
91                 new Cancion("Something", "The Beatles", "3:03"),
92                 new Cancion("Here Comes the Sun", "The Beatles", "3:06")
93             ],
94             "The White Album": [
95                 new Cancion("Hey Jude", "The Beatles", "4:35"),
96                 new Cancion("Let It Be", "The Beatles", "3:50"),
97                 new Cancion("Yesterday", "The Beatles", "2:05")
98             ]
99         },
100         "Michael Jackson": [
101             "Thriller": [
102                 new Cancion("Billie Jean", "Michael Jackson", "4:54"),
103                 new Cancion("Thriller", "Michael Jackson", "5:57"),
104                 new Cancion("Beat It", "Michael Jackson", "4:18")
105             ],
106             "Bad": [

```

```
107     ],
108     new Cancion("Smooth Criminal", "Michael Jackson", "4:17"),
109     new Cancion("The Way You Make Me Feel", "Michael Jackson", "4:58"),
110     new Cancion("Man in the Mirror", "Michael Jackson", "5:19")
111   ],
112   "Miles Davis": {
113     "Kind of Blue": [
114       new Cancion("So What", "Miles Davis", "9:22"),
115       new Cancion("Blue in Green", "Miles Davis", "5:37"),
116       new Cancion("Kind of Blue", "Miles Davis", "9:44")
117     ],
118   },
119   "Daft Punk": {
120     "Discovery": [
121       new Cancion("One More Time", "Daft Punk", "5:20"),
122       new Cancion("Around the World", "Daft Punk", "7:09")
123     ],
124     "Random Access Memories": [
125       new Cancion("Get Lucky", "Daft Punk", "4:08"),
126       new Cancion("Instant Crush", "Daft Punk", "5:37")
127     ],
128   },
129   "Mozart": {
130     "Sinfonías Clásicas": [
131       new Cancion("Eine kleine Nachtmusik", "Mozart", "6:00"),
132       new Cancion("Rondo Alla Turca", "Mozart", "3:30"),
133       new Cancion("Lacrimosa", "Mozart", "3:05")
134     ],
135   },
136   "Queen": {
137     "A Night at the Opera": [
138       new Cancion("Bohemian Rhapsody", "Queen", "5:55"),
139       new Cancion("Love of My Life", "Queen", "3:38")
140     ],
141     "News of the World": [
142       new Cancion("We Will Rock You", "Queen", "2:01"),
143       new Cancion("We Are the Champions", "Queen", "2:59"),
144       new Cancion("Don't Stop Me Now", "Queen", "3:29")
145     ],
146   },
147   "Madonna": {
148     "Like a Virgin": [
149       new Cancion("Like a Virgin", "Madonna", "3:38"),
150       new Cancion("Material Girl", "Madonna", "4:00")
151     ],
152     "The Immaculate Collection": [
153       new Cancion("Vogue", "Madonna", "4:50"),
154       new Cancion("Like a Prayer", "Madonna", "5:40")
155     ],
156   },

```

```
157     "John Coltrane": {
158         "Giant Steps": [
159             new Cancion("Giant Steps", "John Coltrane", "4:45"),
160             new Cancion("Naima", "John Coltrane", "4:22")
161         ],
162         "A Love Supreme": [
163             new Cancion("A Love Supreme", "John Coltrane", "7:50"),
164             new Cancion("My Favorite Things", "John Coltrane", "13:41")
165         ]
166     },
167     "Deadmau5": {
168         "For Lack of a Better Name": [
169             new Cancion("Ghosts 'n' Stuff", "Deadmau5", "5:28"),
170             new Cancion("I Remember", "Deadmau5", "9:07")
171         ],
172         "Random Album Title": [
173             new Cancion("Strobe", "Deadmau5", "10:37"),
174             new Cancion("Faxing Berlin", "Deadmau5", "11:28")
175         ]
176     },
177     "Beethoven": {
178         "Sinfonías": [
179             new Cancion("Symphony No. 9", "Beethoven", "65:00"),
180             new Cancion("Symphony No. 5", "Beethoven", "33:30")
181         ],
182         "Sonatas": [
183             new Cancion("Moonlight Sonata", "Beethoven", "15:00"),
184             new Cancion("Für Elise", "Beethoven", "3:00")
185         ]
186     },
187     "Joaquín Sabina": {
188         "19 días y 500 noches": [
189             new Cancion("19 días y 500 noches", "Joaquín Sabina", "4:15"),
190             new Cancion("Y nos dieron las diez", "Joaquín Sabina", "4:30"),
191             new Cancion("Contigo", "Joaquín Sabina", "3:45")
192         ]
193     },
194     "Shakira": {
195         "Oral Fixation Vol. 2": [
196             new Cancion("Hips Don't Lie", "Shakira", "3:38"),
197             new Cancion("La Tortura", "Shakira", "3:35")
198         ],
199         "Waka Waka": [
200             new Cancion("Waka Waka", "Shakira", "3:22"),
201             new Cancion("Loca", "Shakira", "3:13")
202         ]
203     },
204     "Luis Miguel": {
205         "Romance": [
206             new Cancion("La Incondicional", "Luis Miguel", "4:17"),
207             new Cancion("Hasta Que Me Olvides", "Luis Miguel", "4:41")
```

```
208     ],
209     "Segundo Romance": [
210         new Cancion("Ahora Te Puedes Marchar", "Luis Miguel", "3:11"),
211         new Cancion("El Día Que Me Quieras", "Luis Miguel", "4:15")
212     ]
213 },
214 "Café Tacvba": {
215     "Re": [
216         new Cancion("Eres", "Café Tacvba", "4:27"),
217         new Cancion("La Ingrata", "Café Tacvba", "3:32")
218     ],
219     "Avalancha de Éxitos": [
220         new Cancion("Chilanga Banda", "Café Tacvba", "3:31"),
221         new Cancion("No Controles", "Café Tacvba", "3:05")
222     ]
223 },
224 "Juanes": {
225     "Mi Sangre": [
226         new Cancion("La Camisa Negra", "Juanes", "3:36"),
227         new Cancion("A Dios Le Pido", "Juanes", "3:25")
228     ],
229     "Un Día Normal": [
230         new Cancion("Es Por Ti", "Juanes", "3:58"),
231         new Cancion("Fotografía", "Juanes", "3:58")
232     ]
233 },
234 "Led Zeppelin": {
235     "Led Zeppelin IV": [
236         new Cancion("Stairway to Heaven", "Led Zeppelin", "8:02"),
237         new Cancion("Black Dog", "Led Zeppelin", "4:54")
238     ],
239     "Led Zeppelin II": [
240         new Cancion("Whole Lotta Love", "Led Zeppelin", "5:34")
241     ]
242 },
243 "Pink Floyd": {
244     "The Wall": [
245         new Cancion("Comfortably Numb", "Pink Floyd", "6:22"),
246         new Cancion("Another Brick in the Wall", "Pink Floyd", "3:59")
247     ],
248     "Wish You Were Here": [
249         new Cancion("Wish You Were Here", "Pink Floyd", "5:34")
250     ]
251 },
252 "The Rolling Stones": {
253     "Aftermath": [
254         new Cancion("Paint It Black", "The Rolling Stones", "3:45"),
255         new Cancion("Satisfaction", "The Rolling Stones", "3:44")
256     ],
257     "Goats Head Soup": [
258         new Cancion("Angie", "The Rolling Stones", "4:22"),
259     ]
260 }
```

```
258     ],
259   },
260   "AC/DC": {
261     "Back in Black": [
262       new Cancion("Back in Black", "AC/DC", "4:15"),
263       new Cancion("Thunderstruck", "AC/DC", "4:52")
264     ],
265     "Highway to Hell": [
266       new Cancion("Highway to Hell", "AC/DC", "3:28")
267     ]
268   },
269   "Guns N' Roses": {
270     "Appetite for Destruction": [
271       new Cancion("Sweet Child O' Mine", "Guns N' Roses", "5:56"),
272       new Cancion("Welcome to the Jungle", "Guns N' Roses", "4:31")
273     ],
274     "Use Your Illusion I": [
275       new Cancion("November Rain", "Guns N' Roses", "8:57")
276     ]
277   },
278   "Soda Stereo": {
279     "Canción Animal": [
280       new Cancion("De Música Ligera", "Soda Stereo", "3:32"),
281       new Cancion("Persiana Americana", "Soda Stereo", "4:52")
282     ],
283     "Doble Vida": [
284       new Cancion("En la Ciudad de la Furia", "Soda Stereo", "5:26")
285     ]
286   },
287   "Nirvana": {
288     "Nevermind": [
289       new Cancion("Smells Like Teen Spirit", "Nirvana", "5:01"),
290       new Cancion("Come As You Are", "Nirvana", "3:38"),
291       new Cancion("Lithium", "Nirvana", "4:16")
292     ]
293   },
294   "Radiohead": {
295     "OK Computer": [
296       new Cancion("Karma Police", "Radiohead", "4:24"),
297       new Cancion("Paranoid Android", "Radiohead", "6:23")
298     ],
299     "Pablo Honey": [
300       new Cancion("Creep", "Radiohead", "3:56")
301     ]
302   },
303   "Pearl Jam": {
304     "Ten": [
305       new Cancion("Alive", "Pearl Jam", "5:40"),
306       new Cancion("Jeremy", "Pearl Jam", "5:18"),
307     ]
308   }
309 }
```

```
308             new Cancion("Black", "Pearl Jam", "5:43")
309         ],
310     },
311     "Maná": {
312         "¿Dónde Jugarán los Niños?": [
313             new Cancion("Oye Mi Amor", "Maná", "4:19"),
314             new Cancion("Rayando el Sol", "Maná", "4:10")
315         ],
316         "Sueños Líquidos": [
317             new Cancion("En el Muelle de San Blas", "Maná", "5:52")
318         ]
319     },
320     "Los Fabulosos Cadillacs": {
321         "El León": [
322             new Cancion("Matador", "Los Fabulosos Cadillacs", "5:13"),
323             new Cancion("Vasos Vacíos", "Los Fabulosos Cadillacs", "4:20")
324         ],
325         "Rey Azúcar": [
326             new Cancion("Siguiendo la Luna", "Los Fabulosos Cadillacs", "4:51")
327         ]
328     },
329     "Linkin Park": {
330         "Hybrid Theory": [
331             new Cancion("In the End", "Linkin Park", "3:36"),
332             new Cancion("Numb", "Linkin Park", "3:05")
333         ],
334         "Minutes to Midnight": [
335             new Cancion("What I've Done", "Linkin Park", "3:25")
336         ]
337     },
338     "The White Stripes": {
339         "Elephant": [
340             new Cancion("Seven Nation Army", "The White Stripes", "3:52"),
341             new Cancion("Icky Thump", "The White Stripes", "4:14")
342         ],
343         "White Blood Cells": [
344             new Cancion("Fell in Love with a Girl", "The White Stripes", "1:50")
345         ]
346     },
347     "Arctic Monkeys": {
348         "AM": [
349             new Cancion("Do I Wanna Know?", "Arctic Monkeys", "4:32"),
350             new Cancion("R U Mine?", "Arctic Monkeys", "3:20")
351         ],
352         "Favourite Worst Nightmare": [
353             new Cancion("505", "Arctic Monkeys", "4:13")
354         ]
355     },
356     "Molotov": {
357         "¿Dónde Jugarán las Niñas?": [
358             new Cancion("Gimme The Power", "Molotov", "4:10")
359         ]
360     }
361 }
```

```
359             new Cancion("Puto", "Molotov", "3:05")
360         ],
361         "Dance and Dense Denso": [
362             new Cancion("Frijolero", "Molotov", "3:30")
363         ]
364     },
365     "Zoé": {
366         "Reptilectric": [
367             new Cancion("Soñé", "Zoé", "4:47"),
368             new Cancion("Nada", "Zoé", "4:20")
369         ],
370         "Memo Rex Commander": [
371             new Cancion("Labios Rotos", "Zoé", "4:04")
372         ]
373     },
374     "ABBA": {
375         "Arrival": [
376             new Cancion("Dancing Queen", "ABBA", "3:50"),
377             new Cancion("Mamma Mia", "ABBA", "3:33")
378         ],
379         "ABBA": [
380             new Cancion("Waterloo", "ABBA", "2:45")
381         ]
382     },
383     "Prince": {
384         "Purple Rain": [
385             new Cancion("Purple Rain", "Prince", "8:41"),
386             new Cancion("When Doves Cry", "Prince", "5:52")
387         ],
388         "Parade": [
389             new Cancion("Kiss", "Prince", "3:46")
390         ]
391     },
392     "Britney Spears": {
393         "...Baby One More Time": [
394             new Cancion("Baby One More Time", "Britney Spears", "3:30"),
395             new Cancion("Oops!... I Did It Again", "Britney Spears", "3:31")
396         ],
397         "In the Zone": [
398             new Cancion("Toxic", "Britney Spears", "3:18")
399         ]
400     },
401     "Backstreet Boys": {
402         "Millennium": [
403             new Cancion("I Want It That Way", "Backstreet Boys", "3:33"),
404             new Cancion("As Long As You Love Me", "Backstreet Boys", "3:34")
405         ],
406         "Backstreet Boys": [
407             new Cancion("Everybody", "Backstreet Boys", "3:45")
408         ]
409     }
410 }
```

```
409 },
410     "Ricky Martin": {
411         "Ricky Martin": [
412             new Cancion("Livin' la Vida Loca", "Ricky Martin", "4:03"),
413             new Cancion("María", "Ricky Martin", "4:22")
414         ],
415         "Vuelve": [
416             new Cancion("La Copa de la Vida", "Ricky Martin", "4:30")
417         ]
418     },
419     "Enrique Iglesias": {
420         "Sex and Love": [
421             new Cancion("Bailando", "Enrique Iglesias", "4:03")
422         ],
423         "Escape": [
424             new Cancion("Hero", "Enrique Iglesias", "4:24"),
425             new Cancion("Escape", "Enrique Iglesias", "3:28")
426         ]
427     },
428     "Lady Gaga": {
429         "The Fame": [
430             new Cancion("Poker Face", "Lady Gaga", "3:57"),
431             new Cancion("Bad Romance", "Lady Gaga", "4:54")
432         ],
433         "Born This Way": [
434             new Cancion("Born This Way", "Lady Gaga", "4:20")
435         ]
436     },
437     "Taylor Swift": {
438         "1989": [
439             new Cancion("Shake It Off", "Taylor Swift", "3:39"),
440             new Cancion("Blank Space", "Taylor Swift", "3:51")
441         ],
442         "Fearless": [
443             new Cancion("Love Story", "Taylor Swift", "3:55")
444         ]
445     },
446     "Herbie Hancock": {
447         "Head Hunters": [
448             new Cancion("Chameleon", "Herbie Hancock", "15:41"),
449             new Cancion("Watermelon Man", "Herbie Hancock", "6:28")
450         ],
451         "Empyrean Isles": [
452             new Cancion("Cantaloupe Island", "Herbie Hancock", "5:33")
453         ]
454     },
455     "Pat Metheny": {
456         "Still Life (Talking)": [
457             new Cancion("Last Train Home", "Pat Metheny", "5:43")
458         ],
459         "Pat Metheny Group": [

```

```
460         new Cancion("Phase Dance", "Pat Metheny", "8:25"),
461         new Cancion("Are You Going With Me?", "Pat Metheny", "8:50")
462     ],
463 },
464 "Diana Krall": {
465     "The Look of Love": [
466         new Cancion("The Look of Love", "Diana Krall", "4:42"),
467         new Cancion("Let's Fall in Love", "Diana Krall", "4:18")
468     ],
469     "All for You": [
470         new Cancion("Peel Me a Grape", "Diana Krall", "5:52")
471     ]
472 },
473 "Norah Jones": {
474     "Come Away With Me": [
475         new Cancion("Don't Know Why", "Norah Jones", "3:06"),
476         new Cancion("Come Away With Me", "Norah Jones", "3:18")
477     ],
478     "Feels Like Home": [
479         new Cancion("Sunrise", "Norah Jones", "3:20")
480     ]
481 },
482 "Kraftwerk": {
483     "The Man-Machine": [
484         new Cancion("The Model", "Kraftwerk", "3:40"),
485         new Cancion("Computer Love", "Kraftwerk", "7:15")
486     ],
487     "Autobahn": [
488         new Cancion("Autobahn", "Kraftwerk", "22:43")
489     ]
490 },
491 "The Prodigy": {
492     "The Fat of the Land": [
493         new Cancion("Firestarter", "The Prodigy", "4:40"),
494         new Cancion("Breathe", "The Prodigy", "5:35"),
495         new Cancion("Smack My Bitch Up", "The Prodigy", "5:42")
496     ]
497 },
498 "Chemical Brothers": {
499     "Dig Your Own Hole": [
500         new Cancion("Block Rockin' Beats", "Chemical Brothers", "5:13")
501     ],
502     "Surrender": [
503         new Cancion("Hey Boy Hey Girl", "Chemical Brothers", "4:50")
504     ],
505     "Push the Button": [
506         new Cancion("Galvanize", "Chemical Brothers", "6:33")
507     ]
508 },
509 "Skrillex": {
510     "Bangarang": [
511         new Cancion("Bangarang", "Skrillex", "3:25")
512     ]
513 }
```

```

509         "Skrillex": {
510             "Bangarang": [
511                 new Cancion("Bangarang", "Skrillex", "3:35"),
512                 new Cancion("First of the Year", "Skrillex", "4:22")
513             ],
514             "Scary Monsters and Nice Sprites": [
515                 new Cancion("Scary Monsters and Nice Sprites", "Skrillex", "4:03")
516             ]
517         },
518         "Bach": {
519             "Obras Maestras": [
520                 new Cancion("Toccata and Fugue in D minor", "Bach", "9:20"),
521                 new Cancion("Air on the G String", "Bach", "5:10"),
522                 new Cancion("Jesu, Joy of Man's Desiring", "Bach", "3:52")
523             ]
524         },
525         "Vivaldi": {
526             "The Four Seasons": [
527                 new Cancion("The Four Seasons - Spring", "Vivaldi", "3:30"),
528                 new Cancion("The Four Seasons - Summer", "Vivaldi", "5:40"),
529                 new Cancion("The Four Seasons - Winter", "Vivaldi", "3:25")
530             ]
531         },
532         "Chopin": {
533             "Nocturnes": [
534                 new Cancion("Nocturne Op. 9 No. 2", "Chopin", "4:20")
535             ],
536             "Valses": [
537                 new Cancion("Waltz in D-flat major", "Chopin", "1:48"),
538                 new Cancion("Fantaisie-Impromptu", "Chopin", "5:27")
539             ]
540         },
541         "Ludovico Einaudi": {
542             "Una Mattina": [
543                 new Cancion("Una Mattina", "Ludovico Einaudi", "3:32"),
544                 new Cancion("Nuvole Bianche", "Ludovico Einaudi", "5:57")
545             ],
546             "Divenire": [
547                 new Cancion("Divenire", "Ludovico Einaudi", "6:47")
548             ]
549         }
550     };
551 }
552
553 obtenerArtistas()
554 {
555     return this.artistasDisponibles;
556 }
557
558 obtenerAlbumesPorArtista()
559 {
560     return this.albumesPorArtista;

```

[GestorFiltros.js](#)

```
1  class GestorFiltros
2  {
3      constructor(gestorDatos)
4      {
5          this.gestorDatos = gestorDatos;
6      }
7
8      filtrarArtistas(genero, idioma, decada, artistasSeleccionados)
9      {
10         const artistasDisponibles = this.gestorDatos.obtenerArtistas();
11
12         return artistasDisponibles.filter(artista => {
13             const filtroGenero = new FiltroGenero(artista, genero);
14             const filtroIdioma = new FiltroIdioma(artista, idioma);
15             const filtroDecada = new FiltroDecada(artista, decada);
16             const filtroArtista = new FiltroArtistaSeleccionado(artista, artistasSeleccionados);
17
18             return filtroGenero.cumpleFiltro() && filtroIdioma.cumpleFiltro() && filtroDecada.cumpleFiltro() &&
19                 (artistasSeleccionados.length === 0 || filtroArtista.cumpleFiltro());
20         });
21     }
22 }
```

GestorInterfaz.js

```
1  class GestorInterfaz
2  {
3      constructor(gestorDatos, gestorFiltros, gestorNavegacion)
4      {
5          this.gestorDatos = gestorDatos;
6          this.gestorFiltros = gestorFiltros;
7          this.gestorNavegacion = gestorNavegacion;
8
9          this.generoSeleccionado = '';
10         this.idiomaSeleccionado = '';
11         this.decadaSeleccionada = '';
12         this.artistasSeleccionados = [];
13         this.albumesSeleccionados = [];
14         this.cancionSeleccionada = null;
15     }
16
17     inicializar()
18     {
19         const selectorGenero = document.getElementById('selectorGenero');
20         const selectorIdioma = document.getElementById('selectorIdioma');
21         const selectorDecada = document.getElementById('selectorDecada');
22         const btnMostrarCanciones = document.getElementById('btnMostrarCanciones');
23
24         selectorGenero.addEventListener('change', (e) => [
25             this.generoSeleccionado = e.target.value;
26             this.actualizarListaArtistas();
27             this.limpiarAlbumes();
28             this.cancionSeleccionada = null;
29             this.mostrarCancionActual();
30             this.gestorNavegacion.actualizar(this);
31         ]);
32
33         selectorIdioma.addEventListener('change', (e) => {
34             this.idiomaSeleccionado = e.target.value;
35             this.actualizarListaArtistas();
36             this.limpiarAlbumes();
37             this.cancionSeleccionada = null;
38             this.mostrarCancionActual();
39             this.gestorNavegacion.actualizar(this);
40         });
41
42         selectorDecada.addEventListener('change', (e) => {
43             this.decadaSeleccionada = e.target.value;
44             this.actualizarListaArtistas();
45             this.limpiarAlbumes();
46             this.cancionSeleccionada = null;
47             this.mostrarCancionActual();
48             this.gestorNavegacion.actualizar(this);
49         });
50
51         document.addEventListener('change', (e) => {
52             if (e.target.type === 'checkbox' && e.target.id.startsWith('artista-')) {
53                 this.actualizarListaAlbumes();
54                 this.gestorNavegacion.actualizar(this);
55             }
56         });
57     }
58 }
```

```
56     if (e.target.type === 'checkbox' && e.target.id.startsWith('album-')) {
57         this.gestorNavegacion.actualizar(this);
58     }
59 );
60
61 btnMostrarCanciones.addEventListener('click', () => {
62     this.mostrarCanciones();
63     this.gestorNavegacion.actualizar(this);
64 });
65
66 this.gestorNavegacion.actualizar(this);
67 }
68
69 actualizarListaArtistas()
70 {
71     const listaArtistas = document.getElementById('listaArtistas');
72     listaArtistas.innerHTML = '';
73
74     if (!this.generoSeleccionado)
75     {
76         return;
77     }
78
79     const artistasFiltrados = this.gestorFiltros.filtrarArtistas(
80         this.generoSeleccionado,
81         this.idiomaSeleccionado,
82         this.decadaSeleccionada,
83         []
84     );
85
86     artistasFiltrados.forEach(artista => {
87         const div = document.createElement('div');
88         div.className = 'artista-item';
89
90         const checkbox = document.createElement('input');
91         checkbox.type = 'checkbox';
92         checkbox.value = artista.obtenerNombre();
93         checkbox.id = `artista-${artista.obtenerNombre()}`;
94
95         const label = document.createElement('label');
96         label.htmlFor = `artista-${artista.obtenerNombre()}`;
97         label.textContent = `${artista.obtenerNombre()} (${artista.obtenerIdioma()}, ${artista.obtenerDecada()})`;
98
99         div.appendChild(checkbox);
100        div.appendChild(label);
101        listaArtistas.appendChild(div);
102    });
103 }
```

```
104     actualizarListaAlbumes()
105     {
106         const listaAlbumes = document.getElementById('listaAlbumes');
107         listaAlbumes.innerHTML = '';
108         const checkboxes = document.querySelectorAll('#listaArtistas input[type="checkbox"]:checked');
109         this.artistasSeleccionados = Array.from(checkboxes).map(cb => cb.value);
110
111         if (this.artistasSeleccionados.length === 0) {
112             return;
113         }
114
115
116         const albumesPorArtista = this.gestorDatos.obtenerAlbumesPorArtista();
117         this.artistasSeleccionados.forEach(nombreArtista => {
118             const albumes = albumesPorArtista[nombreArtista];
119
120             if (albumes && typeof albumes === 'object') {
121                 const divArtista = document.createElement('div');
122                 divArtista.style.marginBottom = '15px';
123
124                 const tituloArtista = document.createElement('strong');
125                 tituloArtista.textContent = nombreArtista + ':';
126                 divArtista.appendChild(tituloArtista);
127
128                 Object.keys(albumes).forEach(nombreAlbum => {
129                     const div = document.createElement('div');
130                     div.className = 'album-item';
131                     div.style.marginLeft = '20px';
132                     div.style.marginTop = '5px';
133
134                     const checkbox = document.createElement('input');
135                     checkbox.type = 'checkbox';
136                     checkbox.value = `${nombreArtista}|${nombreAlbum}`;
137                     checkbox.id = `album-${nombreArtista}-${nombreAlbum}`;
138
139                     const label = document.createElement('label');
140                     label.htmlFor = `album-${nombreArtista}-${nombreAlbum}`;
141                     label.textContent = nombreAlbum;
142                     label.style.marginLeft = '5px';
143
144                     div.appendChild(checkbox);
145                     div.appendChild(label);
146                     divArtista.appendChild(div);
147                 });
148
149                 listaAlbumes.appendChild(divArtista);
150             }
151         });
152     }
}
```

```
153     limpiarAlbumes()
154     {
155         const listaAlbumes = document.getElementById('listaAlbumes');
156         listaAlbumes.innerHTML = '';
157         this.albumesSeleccionados = [];
158     }
159
160     mostrarCanciones()
161     {
162         const resultados = document.getElementById('resultados');
163         resultados.innerHTML = '';
164
165         const checkboxes = document.querySelectorAll('#listaAlbumes input[type="checkbox"]:checked');
166         this.albumesSeleccionados = Array.from(checkboxes).map(cb => cb.value);
167
168         if (this.albumesSeleccionados.length === 0) {
169             alert('Por favor selecciona al menos un álbum');
170             return;
171         }
172
173
174         const albumesPorArtista = this.gestorDatos.obtenerAlbumesPorArtista();
175         const listaReproduccion = new ListaReproduccion(`Canciones Seleccionadas`);
176
177         this.albumesSeleccionados.forEach(albumKey => {
178             const [nombreArtista, nombreAlbum] = albumKey.split('|');
179             const albumes = albumesPorArtista[nombreArtista];
180
181             if (albumes && albumes[nombreAlbum]) {
182                 const canciones = albumes[nombreAlbum];
183                 const album = new Album(`${nombreAlbum} - ${nombreArtista}`);
184
185                 canciones.forEach(cancion => {
186                     if (!cancion.artista || cancion.artista === '')
187                     {
188                         cancion.artista = nombreArtista;
189                     }
190                     album.agregar(cancion);
191                 });
192
193                 listaReproduccion.agregar(album);
194             }
195         });
196         resultados.appendChild(listaReproduccion.mostrar());
197     }
198
199     seleccionarCancion(cancion)
200     {
201         this.cancionSeleccionada = cancion;
202         this.gestorNavegacion.actualizar(this);
203         this.mostrarCancionActual();
204     }
```

```
205     mostrarCancionActual()
206     {
207         const reproductorActual = document.getElementById('reproductorActual');
208
209         if (this.cancionSeleccionada) {
210             reproductorActual.innerHTML = '';
211             reproductorActual.style.display = 'block';
212
213             const divCancion = document.createElement('div');
214             divCancion.className = 'cancion-actual';
215
216             const nombreCancion = document.createElement('div');
217             nombreCancion.className = 'nombre-cancion-actual';
218             nombreCancion.textContent = this.cancionSeleccionada.nombre;
219
220             const infoCancion = document.createElement('div');
221             infoCancion.className = 'info-cancion-actual';
222             infoCancion.textContent = `${this.cancionSeleccionada.artista} - ${this.cancionSeleccionada.duracion}`;
223
224             divCancion.appendChild(nombreCancion);
225             divCancion.appendChild(infoCancion);
226             reproductorActual.appendChild(divCancion);
227         } else {
228             reproductorActual.style.display = 'none';
229         }
230     }
231 }
232 }
```

[GestorNavegacion.js](#)

```
1 class GestorNavegacion
2 {
3     constructor() {
4     }
5
6     actualizar(gestorInterfaz)
7     {
8         const contenidoArbol = document.getElementById('contenidoArbol');
9         contenidoArbol.innerHTML = '';
10        const ruta = document.createElement('div');
11        ruta.className = 'ruta-lineal';
12        const elementosRuta = [];
13
14        if (gestorInterfaz.generoSeleccionado) {
15            elementosRuta.push({
16                texto: gestorInterfaz.generoSeleccionado,
17                activo: true
18            });
19        } else {
20            ruta.textContent = 'Selecciona un género...';
21            contenidoArbol.appendChild(ruta);
22            return;
23        }
24
25        if (gestorInterfaz.artistasSeleccionados.length > 0) {
26            if (gestorInterfaz.artistasSeleccionados.length === 1) {
27                elementosRuta.push({
28                    texto: gestorInterfaz.artistasSeleccionados[0],
29                    activo: true
30                });
31            } else {
32                elementosRuta.push({
33                    texto: `${gestorInterfaz.artistasSeleccionados.length} artistas`,
34                    activo: true
35                });
36            }
37        }
38
39        if (gestorInterfaz.albumesSeleccionados.length > 0) {
40            if (gestorInterfaz.albumesSeleccionados.length === 1) {
41                const [, nombreAlbum] = gestorInterfaz.albumesSeleccionados[0].split('|');
42                elementosRuta.push({
43                    texto: nombreAlbum,
44                    activo: true
45                });
46            } else {
47                elementosRuta.push({
48                    texto: `${gestorInterfaz.albumesSeleccionados.length} álbumes`,
49                    activo: true
50                });
51            }
52        }
53
54        if (gestorInterfaz.cancionSeleccionada) {
55            elementosRuta.push({
```

```
56     |         texto: gestorInterfaz.cancionSeleccionada.nombre,
57     |         activo: true
58     |     );
59 }
60
61 elementosRuta.forEach((elemento, index) => {
62     const span = document.createElement('span');
63     if (elemento.activo) {
64         span.className = 'elemento-ruta activo';
65     } else {
66         span.className = 'elemento-ruta';
67     }
68     span.textContent = elemento.texto;
69     ruta.appendChild(span);
70
71     if (index < elementosRuta.length - 1) {
72         const separador = document.createElement('span');
73         separador.className = 'separador-ruta';
74         separador.textContent = ' > ';
75         ruta.appendChild(separador);
76     }
77 });
78
79 contenidoArbol.appendChild(ruta);
80 }
81 }
```

[ListaReproduccion.js](#)

```
1 class ListaReproduccion extends ComponenteMusical
2 {
3     constructor(nombre, componentes)
4     {
5         super(nombre);
6         this.componentes = componentes || [];
7     }
8
9     agregar(componente)
10    {
11        this.componentes.push(componente);
12    }
13
14     obtenerCanciones()
15    {
16        let todasLasCanciones = [];
17        this.componentes.forEach(componente =>
18        {
19            todasLasCanciones = todasLasCanciones.concat(componente.obtenerCanciones());
20        });
21        return todasLasCanciones;
22    }
23
24     mostrar()
25    {
26        const div = document.createElement('div');
27        div.className = 'lista-reproduccion';
28
29        const titulo = document.createElement('h3');
30        titulo.textContent = `Lista: ${this.nombre}`;
31        div.appendChild(titulo);
32
33        this.componentes.forEach(componente =>
34        {
35            div.appendChild(componente.mostrar());
36        });
37
38        return div;
39    }
40 }
```

estilos.css

```
1  * {
2   |     margin: 0;
3   |     padding: 0;
4   |     box-sizing: border-box;
5  }
6
7  body {
8   |     font-family: Arial, sans-serif;
9   |     padding: 20px;
10 }
11
12 .contenedor {
13 |     max-width: 800px;
14 |     margin: 0 auto;
15 |     background-color: #white;
16 |     padding: 20px;
17 }
18
19 h1 {
20 |     margin-bottom: 20px;
21 |     font-size: 24px;
22 }
23
24 h2 {
25 |     margin-bottom: 10px;
26 |     font-size: 18px;
27 }
28
29 .seccion {
30 |     margin-bottom: 20px;
31 }
32
33 select {
34 |     width: 100%;
35 |     padding: 8px;
36 |     font-size: 14px;
37 }
38
39 button {
40 |     padding: 10px 20px;
41 |     font-size: 14px;
42 |     cursor: pointer;
43 }
44
45 #listaArtistas {
46 |     margin-top: 10px;
47 }
48
49 .artista-item {
50 |     margin-bottom: 5px;
51 }
52
53 .artista-item input {
54 |     margin-right: 5px;
55 }
```

```
56
57 #resultados {
58   margin-top: 10px;
59 }
60
61 .cancion {
62   padding: 5px;
63   margin-bottom: 5px;
64   border: 1px solid #ddd;
65   transition: background-color 0.2s;
66 }
67
68 .cancion:hover {
69   background-color: #f0f0f0;
70 }
71
72 .album {
73   margin-bottom: 10px;
74   border: 1px solid #ccc;
75   padding: 10px;
76 }
77
78 .album-titulo {
79   font-weight: bold;
80   margin-bottom: 5px;
81 }
82
83 .arbol-navegacion {
84   position: fixed;
85   bottom: 20px;
86   left: 20px;
87   background-color: white;
88   border: 1px solid #ccc;
89   padding: 15px;
90   max-width: 300px;
91   max-height: 400px;
92   overflow-y: auto;
93 }
94
95 .arbol-navegacion h3 {
96   font-size: 16px;
97   margin-bottom: 10px;
98   border-bottom: 1px solid #ddd;
99   padding-bottom: 5px;
100 }
101
102 #contenidoArbol {
103   font-size: 12px;
104   line-height: 1.6;
105 }
106
107 .ruta-lineal {
108   display: flex;
109   align-items: center;
```

```
110     flex-wrap: wrap;
111     gap: 5px;
112 }
113
114 .elemento-ruta {
115     color: #666;
116     padding: 2px 5px;
117 }
118
119 .elemento-ruta.activo {
120     font-weight: bold;
121     color: #0066cc;
122 }
123
124 .separador-ruta {
125     color: #999;
126     margin: 0 3px;
127 }
128
129 .reproductor-actual {
130     position: fixed;
131     bottom: 20px;
132     left: 50%;
133     transform: translateX(-50%);
134     background-color: white;
135     border: 2px solid #0066cc;
136     border-radius: 8px;
137     padding: 20px 40px;
138     min-width: 300px;
139     text-align: center;
140     display: none;
141     z-index: 1000;
142 }
143
144 .cancion-actual {
145     display: flex;
146     flex-direction: column;
147     align-items: center;
148     gap: 8px;
149 }
150
151 .nombre-cancion-actual {
152     font-size: 20px;
153     font-weight: bold;
154     color: #0066cc;
155 }
156
157 .info-cancion-actual {
158     font-size: 14px;
159     color: #666;
160 }
```

program.html

```
1  <!DOCTYPE html>
2  <html lang="es">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Examen 3</title>
7      <link rel="stylesheet" href="estilos.css">
8  </head>
9  <body>
10     <div class="contenedor">
11         <h1>Reproductor de Música</h1>
12
13         <div class="seccion">
14             <h2>Seleccionar Género</h2>
15             <select id="selectorGenero">
16                 <option value="">--- Selecciona un género ---</option>
17                 <option value="rock">Rock</option>
18                 <option value="pop">Pop</option>
19                 <option value="jazz">Jazz</option>
20                 <option value="electronica">Electrónica</option>
21                 <option value="clasica">Clásica</option>
22             </select>
23         </div>
24
25         <div class="seccion">
26             <h2>Filtros Adicionales</h2>
27             <div>
28                 <label for="selectorIdioma">Idioma:</label>
29                 <select id="selectorIdioma">
30                     <option value="">Todos los idiomas</option>
31                     <option value="espanol">Español</option>
32                     <option value="ingles">Inglés</option>
33                 </select>
34             </div>
35             <div style="margin-top: 10px;">
36                 <label for="selectorDecada">Década:</label>
37                 <select id="selectorDecada">
38                     <option value="">Todas las décadas</option>
39                     <option value="70s">70s</option>
40                     <option value="80s">80s</option>
41                     <option value="90s">90s</option>
42                     <option value="2000">2000 en adelante</option>
43                 </select>
44             </div>
45         </div>
46
47         <div class="seccion">
48             <h2>Seleccionar Artistas</h2>
49             <div id="listaArtistas"></div>
50         </div>
51
52         <div class="seccion">
53             <h2>Seleccionar Álbumes</h2>
54             <div id="listaAlbumes"></div>
55         </div>
```

```
56
57     <div class="seccion">
58         <button id="btnMostrarCanciones">Mostrar Canciones</button>
59     </div>
60
61     <div class="seccion">
62         <h2>Resultados</h2>
63         <div id="resultados"></div>
64     </div>
65 </div>
66
67     <div id="arbolNavegacion" class="arbol-navegacion">
68         <h3>Árbol de Navegación</h3>
69         <div id="contenidoArbol"></div>
70     </div>
71
72     <div id="reproductorActual" class="reproductor-actual">
73     </div>
74
75     <script src="clases/ComponenteMusical.js"></script>
76     <script src="clases/Cancion.js"></script>
77     <script src="clases/Album.js"></script>
78     <script src="clases/ListaReproduccion.js"></script>
79
80
81     <script src="clases/Artista.js"></script>
82     <script src="clases/DecoradorFiltro.js"></script>
83     <script src="clases/FiltroGenero.js"></script>
84     <script src="clases/FiltroIdioma.js"></script>
85     <script src="clases/FiltroDecada.js"></script>
86     <script src="clases/FiltroArtistaSeleccionado.js"></script>
87
88
89     <script src="clases/GestorDatos.js"></script>
90     <script src="clases/GestorFiltros.js"></script>
91     <script src="clases/GestorNavegacion.js"></script>
92     <script src="clases/GestorInterfaz.js"></script>
93
94
95     <script src="programa.js"></script>
96 </body>
97 </html>
```

[Programa.js](#)

```
1  document.addEventListener('DOMContentLoaded', () => {
2      const gestorDatos = new GestorDatos();
3      const gestorFiltros = new GestorFiltros(gestorDatos);
4      const gestorNavegacion = new GestorNavegacion();
5      const gestorInterfaz = new GestorInterfaz(gestorDatos, gestorFiltros, gestorNavegacion);
6
7      window.gestorInterfaz = gestorInterfaz;
8      gestorInterfaz.inicializar();
9  });

});
```

Conclusiones

Con este examen aprendí a cómo implementar correctamente dos patrones de diseño, y a pesar de que lo vimos en clase su uso se puede llevar a cualquier problemática para resolverla de forma distinta gráficamente, pero lógicamente el programa funciona bajo la misma lógica y al incluir otro patrón se modifica un poco la lógica externa sin alterar el patrón principal.