

# PROGRAMACIÓN EN INTERNET



---

Universidad  
de Cádiz

## DOCUMENTACIÓN DEL PROYECTO

Autor:  
Jesús Gabriel Gómez Tocino

FEBRERO 2024

# Índice

	Página
<b>I Hito 1</b>	<b>3</b>
<b>1. Introducción</b>	<b>3</b>
1.1. Competidores potenciales . . . . .	3
1.1.1. Cines Yelmo . . . . .	3
1.1.2. Cinesa . . . . .	4
<b>2. Prototipado</b>	<b>5</b>
2.0.1. Títulos e información importante . . . . .	5
2.0.2. Subtítulos, cabeceras de formulario . . . . .	5
2.0.3. Texto . . . . .	5
2.1. Paleta de colores . . . . .	5
2.1.1. Header y footer . . . . .	5
2.1.2. Recuadros de información . . . . .	5
2.1.3. Color de fondo . . . . .	5
2.2. Landing page . . . . .	6
2.2.1. Espacio principal . . . . .	6
2.2.2. Espacio secundario . . . . .	6
2.2.3. Footer . . . . .	7
2.3. Cartelera . . . . .	7
2.3.1. Todas las películas . . . . .	7
2.4. Película . . . . .	8
2.4.1. Sesiones . . . . .	8
2.4.2. Detalles . . . . .	8
2.5. Comprar entradas . . . . .	10
2.6. Salas . . . . .	13
2.7. Precios y menús . . . . .	16
2.8. Ubicación . . . . .	16
<b>3. Desarrollo e implementación</b>	<b>17</b>
3.1. Consideraciones iniciales . . . . .	17
3.1.1. Enfoques de diseño . . . . .	17
3.1.2. Frameworks CSS . . . . .	17
3.1.3. Elementos comunes a todas las páginas . . . . .	17
3.1.4. Estilos comunes a todas las páginas . . . . .	20
3.1.5. Validaciones W3C HTML & CSS . . . . .	21
3.1.6. Validaciones Accesibilidad . . . . .	21
3.2. Página de inicio: index.html . . . . .	22
3.2.1. Espacio principal . . . . .	22
3.2.2. Espacio secundario . . . . .	23
3.3. Cartelera . . . . .	24
3.4. Película . . . . .	28
3.4.1. Selector de fecha . . . . .	31
3.4.2. Validaciones WAVE . . . . .	31

<b>4. Accesibilidad</b>	<b>34</b>
4.1. Criterio de accesibilidad . . . . .	34
4.2. Aspecto general de accesibilidad . . . . .	34
<b>II Hito 2</b>	<b>35</b>
<b>5. Introducción</b>	<b>35</b>
<b>6. Modificaciones respecto al Hito 1</b>	<b>35</b>
6.1. Cambios comunes . . . . .	35
6.1.1. Imágenes . . . . .	35
6.1.2. Cambios visuales en el footer . . . . .	35
6.1.3. Protección de las URLs . . . . .	35
6.1.4. Títulos de las secciones . . . . .	36
6.2. Página: Precios y menús . . . . .	37
6.3. Página: ¿Cómo llegar? . . . . .	37
6.4. Página: Reserva . . . . .	38
6.4.1. Validaciones JavaScript . . . . .	39
<b>7. Páginas no documentadas en el anterior hito</b>	<b>41</b>
7.1. Página: Pre-Reserva . . . . .	41
7.1.1. Determinación de sesiones disponibles . . . . .	41
7.1.2. Mapa de asientos . . . . .	41
7.1.3. Resultado . . . . .	43
<b>8. API RESTful</b>	<b>44</b>
<b>9. Operaciones RESTful</b>	<b>45</b>
9.1. Documento <i>users</i> . . . . .	45
9.1.1. Formato del documento . . . . .	46
9.1.2. Operaciones . . . . .	46
9.1.3. Gestión usuarios . . . . .	50
9.2. Documento <i>movies</i> . . . . .	51
9.2.1. Formato del documento . . . . .	51
9.3. Documento <i>pelicula-sesiones</i> . . . . .	52
9.3.1. Formato del documento . . . . .	52

# **Parte I**

## **Hito 1**

### **1. Introducción**

Para el proyecto de Programación en Internet, se ha optado por crear una empresa ficticia de cine, para la que desarrollaremos una página web y una aplicación móvil para realizar las siguientes acciones:

- Consultar la cartelera.
- Consultar novedades, próximos estrenos y películas en cartelera relevantes.
- Para una película seleccionada:
  - Consultar las sesiones disponibles para un día dado.
  - Consultar la sinopsis de esta y visualizar su tráiler.
  - Realizar el proceso de compra de entradas para un día y hora seleccionados.\*
- Consultar los distintos menús disponibles, incorporables a la entrada en el proceso de compra.
- Consultar el equipamiento y las facilidades del cine para todos los públicos.
- Consultar cómo llegar a las instalaciones, a través de una dirección y un mapa de Google que permita la navegación de forma sencilla.

\* El sistema manejará de forma correcta el intento de compra paralela de una misma butaca.

#### **1.1. Competidores potenciales**

##### **1.1.1. Cines Yelmo**

###### **Fortalezas:**

- Página bien estructurada
- Accesibilidad a cualquier página de forma sencilla

###### **Debilidades:**

- Mala presentación
- Espacios desaprovechados

### **1.1.2. Cinesa**

#### **Fortalezas:**

- Página bien estructurada
- Excelente presentación

#### **Debilidades:**

- Demasiada información
- Barra de navegación mal distribuida

## **2. Prototipado**

Para trabajar sobre algo visual y establecer una base para el posterior desarrollo del sistema, se ha realizado el siguiente prototipado. Recalcar que el prototipado no debe reflejar necesariamente el producto final, sino establecer las directrices de diseño al desarrollador.

### **2.0.1. Títulos e información importante**

Deben escribirse en MAYÚSCULA y en **negrita**.

### **2.0.2. Subtítulos, cabeceras de formulario**

Deben escribirse en **negrita**.

### **2.0.3. Texto**

Se utiliza la fuente establecida y en un color que facilite la correcta lectura.

## **2.1. Paleta de colores**

El color predominante en esta paleta es el rojo en muchas de sus tonalidades.

### **2.1.1. Header y footer**

La tonalidad seleccionada para estas secciones se identifica mediante el siguiente código de color hexadecimal: #61141b

### **2.1.2. Recuadros de información**

La tonalidad seleccionada para estas secciones se identifica mediante el siguiente código de color hexadecimal: #240101

### **2.1.3. Color de fondo**

La tonalidad seleccionada para estas secciones se identifica mediante el siguiente código de color hexadecimal: #31090B

## 2.2. Landing page

Página principal del sitio web (*landing page*). En esta figura el header, la sección principal con un botón que redirecciona a la cartelera actual, un espacio secundario (aside) con un carrusel con estrenos imperdibles y favoritos actuales y, finalmente, el footer con el *sitemap*, el logo y la información de contacto.

### 2.2.1. Espacio principal



Figura 1: Header y landing page

### 2.2.2. Espacio secundario

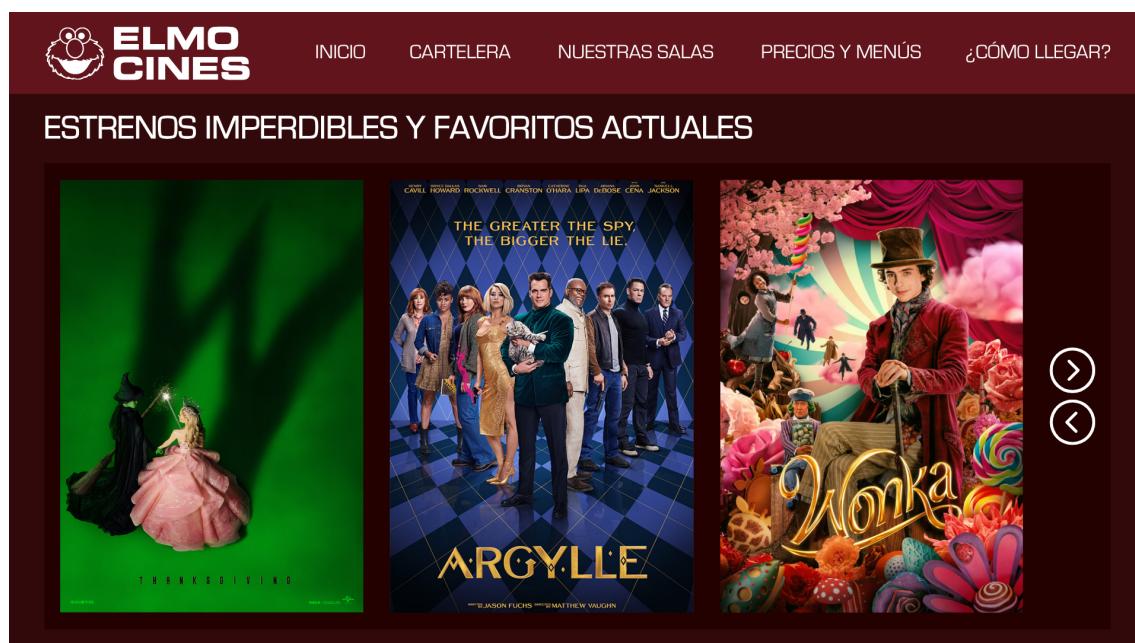


Figura 2: Espacio secundario (aside)

### 2.2.3. Footer



Figura 3: Footer

Este footer figurará en el final de cada página del sitio web, aunque solo se incluya en esta sección del documento.

### 2.3. Cartelera

Nuestra empresa ofrece equipamiento y facilidades comunes a todas las salas del cine, por lo que no procede hacer una página dedicada a describir las características de cada sala. En su lugar, hemos optado por considerar todas las películas (cartelera) como nuestras salas, ya que estas son las que difieren en temática, duración y otros aspectos y sobre las que los clientes decidirán realizar su reserva.

#### 2.3.1. Todas las películas

En primer lugar, se muestran todas las películas, sin proporcionar información más que el nombre. Al pulsar en cada una de ellas, se detallará la información relevante a la película: sesiones, duración, temática, sinopsis, etc.

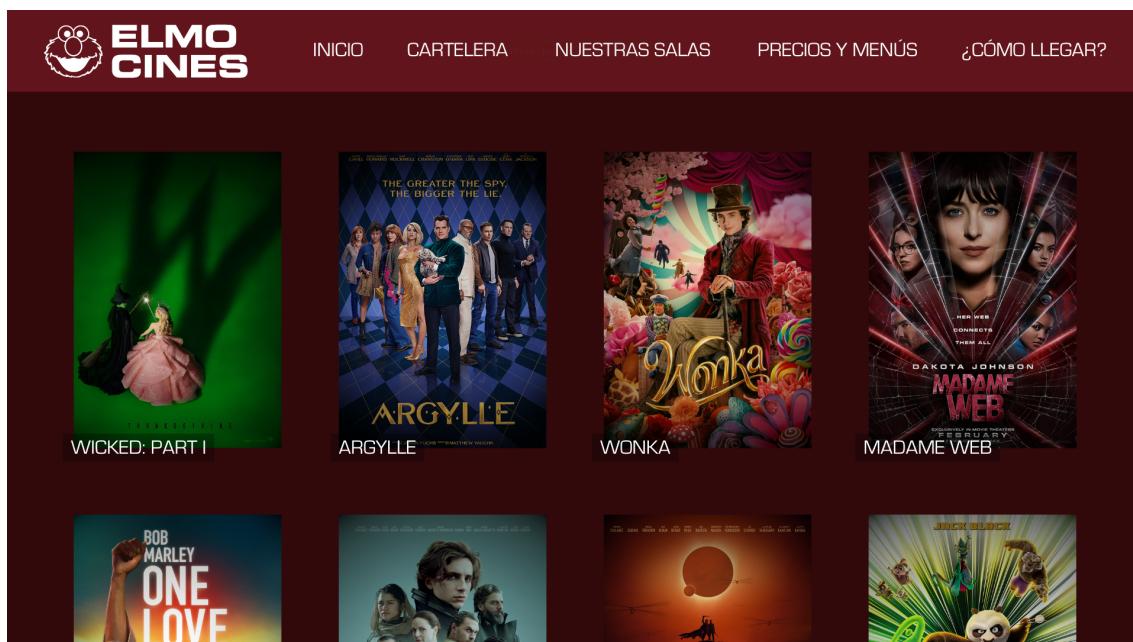


Figura 4: Cartelera

## 2.4. Película

### 2.4.1. Sesiones

Al acceder a una película específica, se mostrarán las distintas fechas y sesiones programadas, habilitando un botón para continuar a la reserva de entradas si el cliente así lo decide.

The screenshot shows the Elmo Cines website interface. At the top, there's a dark header with the logo 'ELMO CINES' featuring a cartoon character. Below the logo are navigation links: INICIO, CARTELERA, NUESTRAS SALAS, PRECIOS Y MENÚS, and ¿CÓMO LLEGAR?. The main content area has a dark background. A section titled 'SESIONES' is displayed, with a dropdown menu showing '14 de febrero de 2024'. Below this, six session boxes are arranged in two rows of three. The first row contains sessions at 16:00 hs. (Sala 1), 17:00 hs. (Sala 2), and 18:00 hs. (Sala 3). The second row contains sessions at 19:00 hs. (Sala 4), 20:00 hs. (Sala 5), and 21:00 hs. (Sala 6). To the right of these boxes is a promotional message: '¡No te quedes sin tus entradas!' and a red button labeled 'COMPRAR ENTRADAS'. Below this, a horizontal bar with the text 'INFORMACIÓN DE LA PELÍCULA' spans the width of the screen. Underneath the bar, the title 'Wicked: Part I' is shown in white text on a dark green background.

Figura 5: Sesiones de una película, dada una fecha

### 2.4.2. Detalles

Para cada película, se mostrará la siguiente información:

- Todas las sesiones disponibles, seleccionada una fecha.
- Información relevante a la película, como:
  - Título
  - Directores
  - Actores
  - Duración
  - Sinopsis
  - Tráiler



## Wicked: Part I

**Directores**

Jon M. Chu

**Actores**

Cynthia Erivo, Ariana Grande, Michelle Yeoh, Jeff Goldblum, Jonathan Bailey, Ethan Slater, Marissa Bode, Bowen Yang, Bronwyn James, Keala Settle

**Duración**

2 h 45 min.

**Sinopsis**

Una amistad improbable surge entre Elphaba y Glinda, dos brujas enfrentadas en la mágica tierra de Oz. Elphaba es injustamente catalogada como una bruja malvada mientras intenta exponer al farsante y corrupto Mago que gobierna la tierra.

Figura 6: Información relevante de la película



Figura 7: Tráiler de la película

## 2.5. Comprar entradas

Se accederá a esta página tras pulsar sobre el botón de *COMPRAR ENTRADAS* que se muestra en la figura 5.

Primeramente, se mostrará un desplegable para seleccionar la fecha a la que se desea asistir a la película. Este input es de tipo **date**.

A continuación, se mostrará un desplegable con las sesiones disponibles para el día seleccionado. En caso contrario, se indicará que no existen sesiones para el día indicado.

Una vez seleccionados los dos datos anteriores, se mostrará un mapa de asientos sobre el que el usuario podrá seleccionar los asientos, que estén disponibles, para su reserva. Los asientos seleccionados se mostrarán simultáneamente de forma numérica y textual. Se habilita un botón de borrado (X) por si el usuario desease des-seleccionar todos los asientos. Una vez seleccionadas las butacas deseadas y comprobado la fecha y



Figura 8: Reserva: Selección de fecha, hora y asientos

sesión por parte del usuario, se solicitará al usuario que confirme los datos y continúe la compra pulsando sobre el botón **CONTINUAR**. Esto redirigirá a la segunda página de reserva, donde se solicitará al usuario toda la información procedente para realizar la compra.

Esta división resulta necesaria para garantizar que el usuario no está realizando la reserva sobre unos asientos que otro usuario ha seleccionado paralelamente en otra sesión. Al pulsar sobre el botón, el sistema realizará una petición al servidor para comprobar si los asientos han sido seleccionados y confirmados por otro usuario previamente. En caso afirmativo, el sistema mostrará al usuario el mensaje de excepción. De lo contrario, la compra continuará sin problema.

En esta segunda parte, se solicitará la siguiente información:

- Nombre y apellidos del comprador
- Correo electrónico del comprador
- Número de teléfono del comprador
- Número de espectadores menores de 18 años, a los que se le aplicará un descuento a la hora de realizar la compra.
- Para cada entrada, se solicitará los menús a incluir (pueden ser más de uno → checkboxes).

The screenshot shows a dark-themed reservation form for Elmo Cines. At the top, there is a navigation bar with the Elmo Cines logo, followed by links for INICIO, CARTELERA, NUESTRAS SALAS, PRECIOS Y MENÚS, and ¿CÓMO LLEGAR?. The main form area contains fields for customer information and movie details. On the left, there are input fields for 'Nombre y apellidos del comprador' (Name and last name), 'Correo electrónico\*' (Email address\*), 'Número de teléfono' (Phone number), and 'Número de espectadores menores de 18 años' (Number of spectators under 18 years old). On the right, there are fields for 'Fecha' (Date) set to '14 de febrero', 'Sesión' (Session) set to '21:30 hs.', and 'Butacas seleccionadas' (Selected seats) showing '[3] A11, A12, A13'. Below these fields is a note: '\* Proporcione una dirección de correo válida, ya que deberá confirmar la compra de entradas en dicha dirección.' (Please provide a valid email address, as you will need to confirm the purchase of tickets at this address.) At the bottom left, there is a dropdown menu for 'Entrada 1: Menú' (Entry 1: Menu) with the option 'Sin menú' (No menu) selected.

Figura 9: Reserva: Cumplimentación de información

Asimismo, se mostrará paralelamente y en tiempo real el resumen de la compra (importe total y desglose e información del comprador). De esta manera, el usuario podrá comprobar y verificar todos los datos antes de confirmar su compra.



Entrada 2: Menú

Menú Grande ▾

Entrada 3: Menú

Menú Infantil ▾

## RESUMEN

2x - Entradas adultos [10 €]  
1x - Entradas menores [6 €]  
1x - Menú grande [8 €]  
1x - Menú infantil [5 €]

Total: 29 €

## INFORMACIÓN

Nombre: Sergio García  
Correo electrónico: canekr@gmail.com  
Teléfono: 612345678

Revisa los datos antes de  
confirmar la compra

CONFIRMAR

Figura 10: Reserva: Resumen y confirmación de la compra

Una vez confirmada la compra, se enviará un correo al cliente para confirmar de nuevo la compra.



¡Compra realizada!

Revisa tu correo electrónico para confirmar y pagar la compra :)

VOLVER AL INICIO

Figura 11: Reserva: Compra realizada (Pendiente de confirmación)

## 2.6. Salas

En esta página se especificarán las características comunes a todas las salas, tanto tecnológicas como de accesibilidad. Las distintas secciones alternarán su posición entre derecha e izquierda.

La siguiente secuencia de imágenes conforman una única página. El espaciado dispuesto en las distintas imágenes es de carácter únicamente ilustrativo, este será ajustado de manera lineal/continua.

The screenshot shows the Elmocines website with a dark red header featuring the logo 'ELMO CINES'. Below the header, there are five navigation links: INICIO, CARTELERA, NUESTRAS SALAS, PRECIOS Y MENÚS, and ¿CÓMO LLEGAR?. The main content area contains text about the design of their movie theaters and a large image of an IMAX screen displaying the word 'IMAX'.

En Elmocines, cada una de nuestras salas está diseñada para ofrecer una experiencia cinematográfica sin precedentes, marcando el nuevo estándar en entretenimiento audiovisual. Con un compromiso firme hacia la innovación y la accesibilidad, garantizamos que cada visita sea memorable.

### PROYECCIÓN Y PANTALLAS

Nuestras salas están equipadas con la tecnología de proyección láser IMAX, proporcionando una claridad de imagen y una profundidad de campo que redefine lo que significa ver una película.

Las pantallas de tamaño gigante, curvas para una experiencia envolvente, aseguran que cada asiento sea el mejor de la casa.

La tecnología IMAX, combinada con proyectores láser de última generación, ofrece colores más vivos y un contraste sin igual, llevando cada detalle a la vida con una precisión excepcional.

Figura 12: Salas: Introducción y Proyección y pantallas

The screenshot shows the Elmocines website with a dark red header featuring the logo 'ELMO CINES'. Below the header, there are five navigation links: INICIO, CARTELERA, NUESTRAS SALAS, PRECIOS Y MENÚS, and ¿CÓMO LLEGAR?. The main content area contains text about the Dolby Atmos sound system and a large image of a movie theater interior with red seats and speakers visible above the screen.

### SISTEMA DE SONIDO

Para complementar nuestra experiencia visual de vanguardia, contamos con el sistema de sonido Dolby Atmos.

Esta tecnología permite un audio tridimensional que fluye a tu alrededor, creando un ambiente sonoro completo y rico en matices.

Con altavoces distribuidos estratégicamente en toda la sala, incluyendo el techo, te sumergirás en un mundo de sonido donde el audio proviene de todas las direcciones, incluso de arriba.

Figura 13: Salas: Sistema de sonido

## SILLONES PREMIUM Y FACILIDADES

Priorizamos tu comodidad por encima de todo.

Por ello, nuestros sillones reclinables premium ofrecen ajustes personalizados que permiten encontrar la posición ideal para disfrutar de tu película.

Con amplio espacio para las piernas, soporte lumbar, y la opción de calor y masaje, aseguramos el máximo confort durante tu estancia.

Cada asiento cuenta con cargadores USB y portavasos, manteniendo tus dispositivos cargados y tus bebidas seguras.



Figura 14: Salas: Sillones premium y facilidades



## ACCESIBILIDAD UNIVERSAL

Elmocines se enorgullece de ser un espacio completamente accesible.

Todas las áreas, incluidas nuestras salas de cine, están diseñadas para ser accesibles para personas con movilidad reducida.

Ofrecemos audífonos para personas con dificultades auditivas y descripciones de audio para visitantes con discapacidad visual.

Nuestro personal está siempre disponible para asistir y garantizar una experiencia inclusiva y confortable para todos.

Figura 15: Salas: Accesibilidad Universal



INICIO

CARTELERA

NUESTRAS SALAS

PRECIOS Y MENÚS

¿CÓMO LLEGAR?

La combinación de tecnología de punta en proyección y sonido, junto con nuestros sillones de lujo y compromiso con la accesibilidad, hacen de Elmocines el destino definitivo para los amantes del cine.

Cada visita a nuestras salas es una promesa de inmersión total en el mundo del cine, donde la realidad supera la imaginación.



Figura 16: Salas: Conclusión

## 2.7. Precios y menús

En esta página se indican los precios de cada tipo de entrada, así como los distintos menús y sus respectivos precios.

The screenshot shows the Elmo Cines website with a dark red header. The header features the Elmo Cines logo on the left and navigation links for 'INICIO', 'CARTELERA', 'NUESTRAS SALAS', 'PRECIOS Y MENÚS', and '¿CÓMO LLEGAR?' on the right. Below the header, the page title 'PRECIOS Y MENÚS' is displayed in large white capital letters. To the right of the title is a photograph of a movie theater seat with a red cup and a white bowl filled with popcorn. On the left side of the main content area, there are four menu options with their descriptions and prices:

- Better Together - 10 €**  
Incluye un bol de palomitas extra grande, una bolsa de gominolas y 2 bebidas grandes a elegir.
- Menú grande - 8 €**  
Incluye un bol de palomitas grande y una bebida grande a elegir.
- Menú mediano - 6 €**  
Incluye un bol de palomitas mediano y una bebida mediana a elegir.
- Menú infantil - 4 €**  
Incluye un bol de palomitas pequeño y una botella de agua.

Figura 17: Precios y menú: Menús y sus respectivos precios

## 2.8. Ubicación

En esta última página del sitio web, se indicará al cliente donde encontrar las instalaciones de la empresa de cine de forma textual (dirección) y de forma gráfica mediante la herramienta de mapas incrustados de Google Maps.

The screenshot shows the '¿Cómo llegar?' (How to get there) section of the Elmo Cines website. The header is identical to the previous page, featuring the Elmo Cines logo and navigation links. The main content area has a dark red background with the heading '¿COMO LLEGAR?' in large white capital letters. Below the heading is a text block providing directions:

Nos puedes encontrar en Puerto Real, en la Avenida Universidad de Cádiz n.<sup>o</sup> 10, junto a la Escuela Superior de Ingeniería.

En los alrededores del establecimiento podrá encontrar un aparcamiento amplio y suficiente para albergar un aforo completo en nuestras instalaciones.

Interactúe con el mapa si necesita situarse u obtener una visión más amplia de la zona :)

On the right side of the page is a Google Map showing the location of Elmo Cines. The map includes coordinates (36°32'13.6"N 6°12'05.3"W), an address (11519 Puerto Real, Cádiz), and a 'Cómo llegar' (Get there) button. The map also shows surrounding landmarks like 'UNIVERSIDAD', 'Puerta de las Lagunas', 'Mirador Matagorda', 'Cementerio Municipal San Roque Puerto Real', and 'Residencia de Estudiantes Bahía'. A red pin marks the exact location of the cinema.

Figura 18: ¿Cómo llegar?: Indicaciones y mapa de Google

### 3. Desarrollo e implementación

#### 3.1. Consideraciones iniciales

##### 3.1.1. Enfoques de diseño

En esta memoria se referirá recurrentemente a dos tipos de pantallas: pantallas grandes y pantallas pequeñas, en específico la de los dispositivos móviles.

Esta categorización se realiza en base a dos dimensiones de anchura:

- **Pantallas grandes:** Aquellas con anchura superior a 767.98px.
- **Pantallas pequeñas:** Aquellas con anchura inferior a 767.98px.

Aunque el modelado de las páginas esté principalmente enfocado en estos dos dispositivos, se ha cuidado la responsividad intermedia lo máximo posible.

##### 3.1.2. Frameworks CSS

Para la estilización del sitio web, se ha hecho uso del framework CSS Bootstrap, delegando algunos aspectos de formateo y visualización a esta herramienta.

No obstante, determinados puntos de ruptura y algunos ajustes fueron desarrollados a medida.

Cabe destacar, y resulta evidente, la evolución y familiarización con esta herramienta a lo largo del desarrollo del sitio web. En las primeras páginas implementadas el uso de Bootstrap era mínimo, mientras que en las últimas páginas es predominante a la hora de aplicar estilos.

La motivación detrás de la integración de Bootstrap en nuestro sitio web principalmente reside en la popularidad de este, además de la variada y completa documentación de la que dispone y la activa comunidad que tiene.

##### 3.1.3. Elementos comunes a todas las páginas

Se ha respetado en todo momento los siguientes requerimientos especificados para la superación de este hito:

- Cada página deberá tener un título y las etiquetas *<meta>* más relevantes.
- Cada página deberá disponer de una cabecera con un logotipo con un hipervínculo.
- Cada página deberá tener menú principal para navegar entre las páginas.
- Cada página deberá tener un pie de página que deberá tener el símbolo © y correo electrónico de contacto (enlace con mailto).

## Título y etiquetas <meta>

Además de establecer el título correspondiente a cada página, se han establecido las siguientes etiquetas:

```
<meta name="keywords" content="cine, película, películas, cartelera, entradas, entretenimiento, cultura">
<meta name="description" content="ElmoCines - Cine premium">
<meta name="subject" content="Cartelera de cine y venta de entradas">
<meta name="copyright" content="ElmoCines">
<meta name="author" content="Jesús Gómez Tocino, jesus.gomeztocino@alum.uca.es">
<meta name="designer" content="Jesús Gómez Tocino, jesus.gomeztocino@alum.uca.es">
<meta charset="UTF-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
```

## Header

La cabecera se ha diseñado haciendo uso de Bootstrap, fusionando las etiquetas semánticas *header* y *nav* y facilitando la visualización en distintos dispositivos de distintos tamaños. Este está presente en todo momento en la pantalla (posicionamiento fixed) y contiene un sitemap del sitio web, y contiene al logotipo con un hipervínculo que redirige a la página principal, además de los distintos elementos de navegación entre las distintas páginas del sitio web.

En dispositivos con pantalla grande (superior a 767px):



Figura 19: Header en dispositivos de pantalla grande

En dispositivos con pantalla pequeña (inferior a 767px):



Figura 20: Header en dispositivos de pantalla pequeña

El pliegue de los elementos, mostrado al pulsar sobre el botón del desplegable, es realizado de manera automática por Bootstrap, siendo esto un punto importantísimo a favor para su aplicación.

## Footer

El pie de página se ha diseñado haciendo uso de Bootstrap, facilitando la visualización en distintos dispositivos de distintos tamaños. Este facilita la navegación entre las páginas, además de incluir una dirección de correo (con un enlace *mailto*) y el símbolo de *copyright* requerido.

**En dispositivos con pantalla grande:**



Figura 21: Footer en dispositivos de pantalla grande

**En dispositivos con pantalla pequeña:**



Figura 22: Footer en dispositivos de pantalla pequeña

Como puede verse, el footer en dispositivos móviles ha sido reordenado para mostrarse en primer lugar, en lugar del *Sitemap*. Esto se consigue mediante el siguiente media query:

```
@media (max-width: 767px) {  
    .container .logo{  
        order: 1;  
    }  
    .container .sitemap{  
        order: 2;  
    }  
    .container .contacto{  
        order: 3;  
    }  
}
```

### 3.1.4. Estilos comunes a todas las páginas

Siguiendo las buenas prácticas de programación con CSS y para facilitar el diseño de las hojas de estilos, respetando la distribución y colores especificados en el prototipado, al comienzo de todas las hojas de estilos se ha incluido el siguiente código:

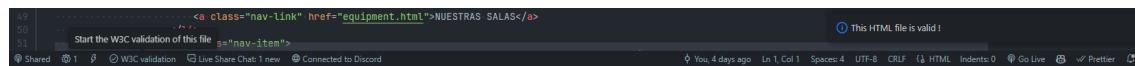
```
:root {  
    --color-principal: #61141b;  
    --color-secundario: #31090b;  
    --color-terciario: #240101;  
    --color-texto: white;  
}  
  
/* Estilos para toda la página */  
body {  
    background-color: var(--color-secundario);  
    color: var(--color-texto);  
    min-height: 100vh;  
    display: flex;  
    flex-direction: column;  
}  
  
main {  
    flex-grow: 1;  
}
```

Esto especifica:

- Variables de color para usar a lo largo de la hoja de estilos y evitar discrepancias en la paleta de colores.
- Que todo texto sea de color blanco.
- Que el tamaño del main siempre ocupe, como mínimo, el tamaño completo de la pantalla donde se visualiza.

### 3.1.5. Validaciones W3C HTML & CSS

Todos los archivos HTML y CSS han pasado las validaciones W3C. Para garantizar este objetivo, se ha hecho uso de la extensión *W3C Web Validator*, disponible en Visual Studio Marketplace.



### 3.1.6. Validaciones Accesibilidad

Todas las páginas pasaron las validaciones de accesibilidad, a excepción de una (movie.html) cuyos motivos se explicarán en su correspondiente sección. Para garantizar este objetivo, se ha hecho uso de la extensión *WAVE*, disponible en las correspondientes tiendas de extensiones para el navegador usado en cuestión.

A screenshot of the WAVE web accessibility evaluation tool. On the left, there's a summary panel with counts for Errors (0), Contrast Errors (0), Alerts (3), Features (5), Structural Elements (16), and ARIA (4). It also includes a "View details" button and a message about no errors detected. The main part of the screen shows the Elmo Cines website with two sections: "ENTRADAS" and "MENÚS". The "ENTRADAS" section contains information about general and discounted ticket purchases, accompanied by images of movie tickets. The "MENÚS" section shows a menu item for "Better Together - 10 €". The WAVE logo and "powered by WebAIM" are visible at the top.

### 3.2. Página de inicio: index.html

En contraste con las figuras 1 y 2, a continuación se muestran los resultados finales de los espacios principal (sección main) y secundario (sección aside).

#### 3.2.1. Espacio principal



Figura 23: Espacio principal: Pantallas grandes

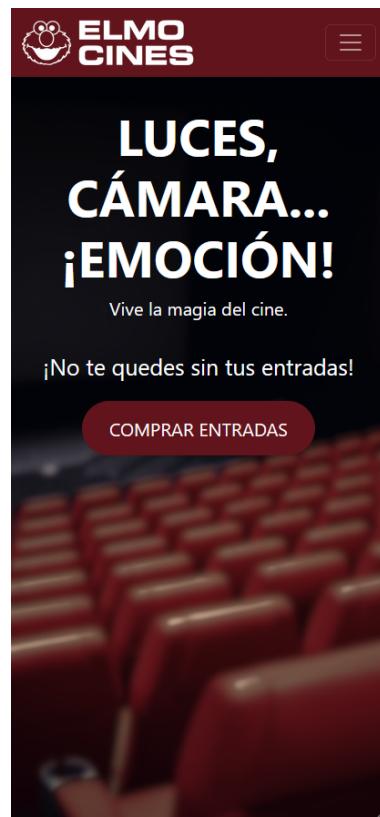


Figura 24: Espacio principal: Pantallas móviles

Con ligeras modificaciones respecto al prototipado original, la sección principal de la página de bienvenida muestra eslogan de la compañía, acompañado de un botón que redirige a la cartelera.

### 3.2.2. Espacio secundario

Para el espacio secundario, tal y como se planteó en el prototipado, se ha implementado un carrusel que muestra, y alterna automáticamente, distintas películas.

Dicho carrusel ha sido implementado haciendo uso de la librería *Slick*, que permite incorporar elementos de este tipo.

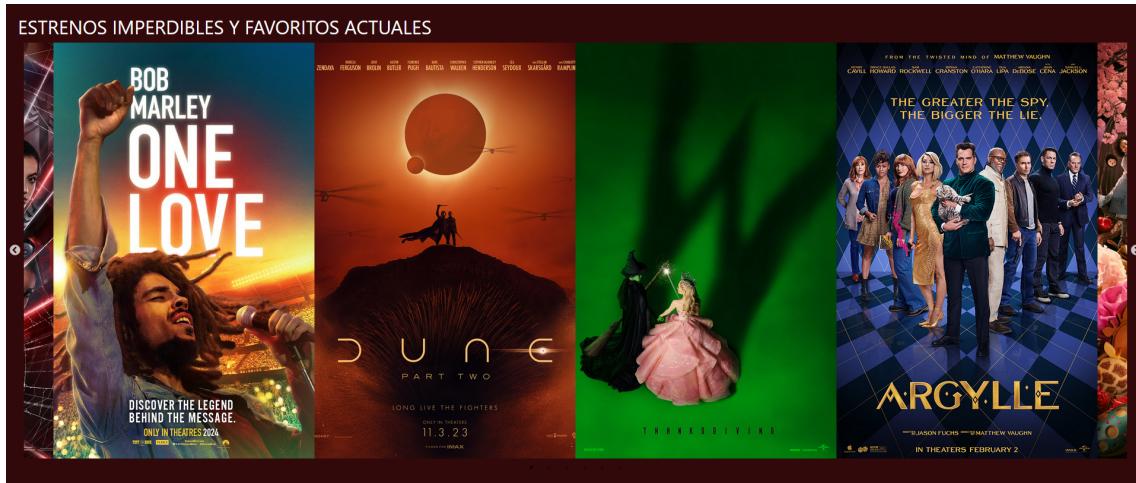


Figura 25: Espacio secundario: Pantallas grandes



Figura 26: Espacio secundario: Pantallas móviles

### 3.3. Cartelera

Para esta página, se pretendía simular un comportamiento real sin implementar, todavía, una API para ello.

El comportamiento buscado se basa en la carga dinámica de contenidos, haciendo uso del DOM. En este caso, por cada película se genera, se ajusta y se completa con una película en concreto, manteniendo en todo momento una altura y anchura común.

Para lograr esto, se ha diseñado un archivo JSON con la siguiente estructura:

```
{  
  "movies": {  
    "wicked": {  
      "title": "Wicked: Part I",  
      "path_to_cover": "wicked.png"  
    },  
    "argylle": {  
      "title": "Argylle",  
      "path_to_cover": "argylle.png"  
    },  
    "wonka": {  
      "title": "Wonka",  
      "path_to_cover": "wonka.png"  
    },  
    "dune": {  
      "title": "Dune",  
      "path_to_cover": "dune1.png"  
    },  
    "dune2": {  
      "title": "Dune: Part II",  
      "path_to_cover": "dune2.png"  
    },  
    "madame-web": {  
      "title": "Madame Web",  
      "path_to_cover": "madame-web.png"  
    },  
    "one-love": {  
      "title": "One Love",  
      "path_to_cover": "one-love.png"  
    }  
  }  
}
```

Donde se mantiene un diccionario de películas (*movies*) para cada cual se almacena:

- El ID
- El título oficial
- El nombre del archivo de la carátula

En base a esta información, se generarán los correspondientes contenidos a memoria.

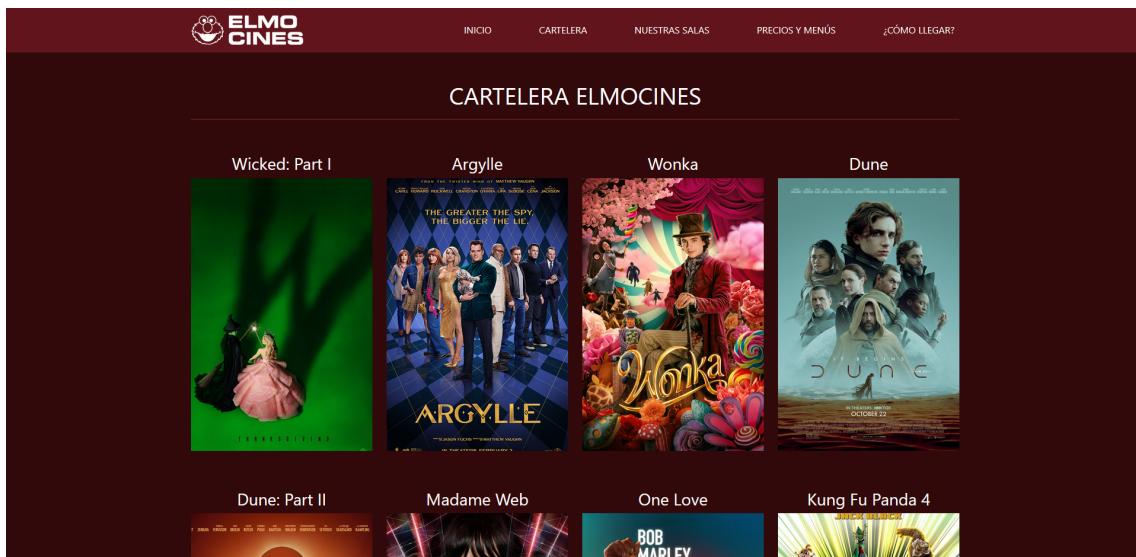


Figura 27: Cartelera: Pantallas grandes



Figura 28: Cartelera: Pantallas móviles

Para conseguir esto, se dispone en el main de un contenedor, con la siguiente estructura:

```
<main>
  <div class = "container my-5 billboardContainer">
    <div class = "row py-5 text-center" id = "billboard">

      </div>
    </div>
  </main>
```

Donde en el div con ID *billboard* se irán introduciendo las distintas películas, mediante la siguiente función JavaScript:

```
// Crea un nuevo elemento de película
// y lo añade al DOM bajo el elemento con ID 'billboard'
function newMovie(movie, movieId) {
  const billboard = document.querySelector("#billboard");
  const column = document.createElement('div');
  column.classList.add('col-md-6');
  column.classList.add('col-sm-6');
  column.classList.add('col-lg-3');
  column.classList.add('col-12');
  billboard.appendChild(column);

  const movieDiv = document.createElement('div');
  movieDiv.classList.add('movie');

  const movieTitle = document.createElement('h3');
  movieTitle.textContent = movie.title;
  movieDiv.appendChild(movieTitle);

  column.appendChild(movieDiv);

  const imgDiv = document.createElement('div');
  imgDiv.classList.add('imgDiv');
  imgDiv.classList.add('mb-3');
  movieDiv.appendChild(imgDiv);

  const movieLink = document.createElement('a');
  movieLink.href = `movie.html?movie=${movieId}`;
  imgDiv.appendChild(movieLink);

  const movieCover = document.createElement('img');
  movieCover.classList.add('img-fluid');
  movieCover.classList.add('imgCover')
  movieCover.src = `/images/movies/${movie.path_to_cover}`;
  movieCover.alt = movie.title;
  movieLink.appendChild(movieCover);
}
```

El parámetro que recibe la función (*movie*), es la película en cuestión cargada desde el JSON, cuya metodología de extracción es la siguiente:

```
// Carga el contenido de la cartelera de películas desde un archivo JSON
// y lo muestra en el DOM
function loadBillboard() {
    fetch('/json/billboard.json')
        .then(response => response.json())
        .then(data => {
            Object.entries(data.movies).forEach(([movieId, movie], index)
                => { newMovie(movie, movieId); })
        });
    }
    .catch(error =>
        console.error('Error al cargar la cartelera:', error));
}
```

Cada vez que se menciona la carga de datos a memoria desde un archivo JSON, se procederá de la misma manera que la ilustrada en este último fragmento de código.

### 3.4. Película

De la misma manera que en la cartelera, esta página carga de forma dinámica, desde archivos JSON, la película seleccionada en la cartelera. Dichos archivos contienen tanto las sesiones asociadas como la información propia a esta. Esto se hace con la intención de mantener una única página HTML que sirva para cualquier película que exista o se incorpore en un futuro, facilitando la depuración, la flexibilidad y el ahorro de costes, e incluso el adelantar trabajo para el siguiente hito.

El archivo JSON de información de la película presenta la siguiente estructura:

```
{  
    "cover": "wicked.png",  
    "movie": "Wicked Part I",  
    "directors": "Jon M. Chu",  
    "actors": [  
        "Cynthia Erivo",  
        "Ariana Grande",  
        "Michelle Yeoh",  
        "Jeff Goldblum",  
        "Jonathan Bailey",  
        "Ethan Slater",  
        "Marissa Bode",  
        "Bowen Yang",  
        "Bronwyn James",  
        "Keala Settle"  
    ],  
    "duration": 165,  
    "synopsis": "Una amistad improbable surge entre Elphaba y Glinda,  
    dos brujas enfrentadas en la mágica tierra de Oz.  
    Elphaba es injustamente catalogada como una bruja malvada  
    mientras intenta exponer al farsante y corrupto Mago que gobierna la tierra.",  
    "trailer": "https://www.youtube-nocookie.com/  
    embed/F1dvX9Vs0ns?si=T201ceF16vLVLArM"  
}
```

Mientras que el archivo JSON de las sesiones se estructura de la siguiente manera:

```
{  
    "2024-03-02": {  
        "15:20": {  
            "butacas": 2,  
            "sala": 5,  
            "butacas_detalles": [  
                {  
                    "numero": "A1",  
                    "estado": "ocupado"  
                },  
                {  
                    "numero": "A2",  
                    "estado": "ocupado"  
                },  
                {  
                    "numero": "A3",  
                    "estado": "ocupado"  
                },  
                ...  
                {  
                    "numero": "F3",  
                    "estado": "ocupado"  
                },  
                {  
                    "numero": "F4",  
                    "estado": "ocupado"  
                },  
                {  
                    "numero": "F5",  
                    "estado": "ocupado"  
                }  
            ]  
        }  
    }  
}
```

Debido a la extensión del documento, y para hacer pruebas con casos prácticos y automatizar la generación de estos archivos, se ha optado por desarrollar un script en Python que genere estos archivos:

```
import random
from datetime import datetime, timedelta
import json

def generar_butacas(max_butacas):
    filas = ['A', 'B', 'C', 'D', 'E', 'F']
    columnas = [1, 2, 3, 4, 5]
    butacas = []
    for fila in filas:
        for columna in columnas:
            butacas.append({"numero": f"{fila}{columna}", "estado": "libre"})

    butacas_ocupadas = random.sample(butacas, max_butacas)
    for butaca in butacas_ocupadas:
        butaca["estado"] = "ocupado"

    return sorted(butacas, key=lambda x: x["numero"])

def generar_sesiones(fecha_inicio, fecha_fin, max_sala, total_butacas,
sesiones_por_dia, hora_inicio=15, probabilidad_dia_vacio=0.1):
    sesiones = {}
    fecha_actual = fecha_inicio

    while fecha_actual <= fecha_fin:
        if random.random() > probabilidad_dia_vacio:
            horario_sesion = hora_inicio
            sesiones_del_dia = {}
            for _ in range(sesiones_por_dia):
                if horario_sesion >= 24:
                    break
                hora = f"{horario_sesion:02d}:{random.randint(0, 59):02d}"
                butacas_libres = random.randint(0, total_butacas)
                sesiones_del_dia[hora] = {
                    "butacas": butacas_libres,
                    "sala": random.randint(1, max_sala),
                    "butacas_detalles": generar_butacas(total_butacas - butacas_libres)
                }
            horario_sesion += random.randint(1, (24 - hora_inicio) // sesiones_por_dia)

            sesiones[fecha_actual.strftime("%Y-%m-%d")] =
                {hora: sesiones_del_dia[hora] for hora in
                 sorted(sesiones_del_dia)}

        fecha_actual += timedelta(days=1)
    return sesiones
```

```

fecha_inicio = datetime(2024, 3, 1)
fecha_fin = datetime(2024, 3, 31)
max_sala = 15
total_butacas = 30 # Total de butacas por sesión
sesiones_por_dia = 5
hora_inicio_sesiones = 15 # Hora de inicio de las sesiones (15:00)
probabilidad_dia_vacio = 0.1 # Probabilidad de que un día no tenga sesiones

```

Este script funciona en base a los siguientes argumentos:

- Fecha de inicio del tiempo de proyección
- Fecha de finalización del tiempo de proyección
- Número máximo de salas en el cine
- Número máximo de butacas por sala
- Número de sesiones por día
- Hora de inicio de las sesiones programadas
- Probabilidad de que un día no tenga ninguna sesión programada

### 3.4.1. Selector de fecha

Este selector de fecha, que por defecto selecciona la primera fecha disponible y desmarca los días no seleccionables (anteriores a la fecha de hoy y aquellos sin sesiones programadas), cambiará en el momento de su selección las sesiones mostradas en pantalla, que corresponderán a las especificadas en el diccionario de fechas JSON.

En este caso, y siguiendo los requisitos y lo planteado en el prototipo, era requerido un input de tipo *date* para seleccionar la fecha de las sesiones a mostrar. Para una mayor flexibilidad en la personalización del input y la limitación física de meses y días seleccionables, se ha incorporado la librería *Flatpickr*.

Debido a las limitaciones que presenta los inputs de tipo *date*, *Flatpickr* modela un selector de fechas propio sobre un input de tipo textual (*text*).

### 3.4.2. Validaciones WAVE

Esta página genera un error a la hora de pasar el validador *WAVE*, en concreto en el selector fecha. Aun teniendo un label especificado para el selector, *Flatpickr* genera internamente otro *form-control* que no tiene ningún label asociado, de ahí que el validador marque como error la ausencia de dicho elemento. Al ser una librería externa, y a pesar de haber intentado solucionarlo, se ha optado por dejar este error a tratarse de un elemento externo.

## SESIONES

Seleccione un día:

Jueves 21 de Marzo

Marzo 2024

Lun Mar Miér Jue Viér Sáb Dom

19	20	21	22	23	24	25
26	27	28	29	30	31	

Hora: 17:20

Sala: 6

Bandas restantes: 26

¡No te quedes sin tus entradas!

[Comprar Entradas](#)

## INFORMACIÓN DE LA PELÍCULA



### Wicked Part I

#### DIRECTORES:

Jon M. Chu

#### ACTORES:

Cynthia Erivo, Ariana Grande, Michelle Yeoh, Jeff Goldblum, Jonathan Bailey, Ethan Slater, Marissa Roldán, Bowen Yang, Broadway James, Keala Settle

#### DURACIÓN:

165 min

#### SINOPSIS:

Una amistad improbable surge entre Elphaba y Glinda, dos brujas enfrentadas en la mágica tierra de Oz. Elphaba es injustamente catalogada como una bruja malvada mientras intenta explorar el fantástico y corrupto Mago que gobierna la tierra.



Figura 29: Película: Pantallas grandes

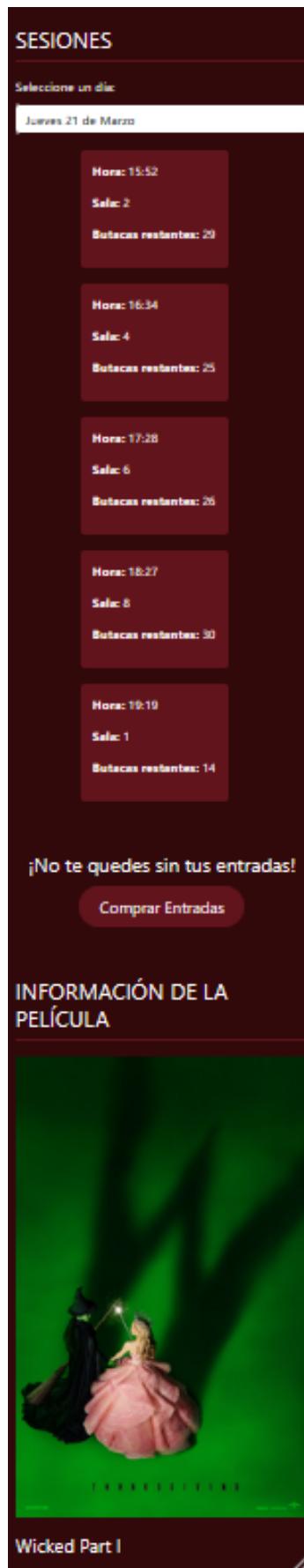


Figura 30: Película: Pantallas móviles

## 4. Accesibilidad

### 4.1. Criterio de accesibilidad

El criterio de accesibilidad del estándar WCAG 2.0 que se ha tenido en consideración a la hora de desarrollar la página web es el de contraste.

El contraste es una medida que normalmente suele pasar muy desapercibida por la gente, pero puede beneficiar a gran cantidad de personas. Las personas que tienen una baja visión o las personas mayores, debido al deterioro de la visión que sufren debido al paso de los años, pueden encontrar dificultades a la hora de visualizar el contenido de una página web. Por tanto, este criterio es muy beneficioso para ese tipo de personas.

De la misma manera, personas sin ningún tipo de discapacidad visual pueden sacar provecho de este aspecto, ya que en entornos con poca iluminación es primordial que los textos y las imágenes sean legibles.

En el desarrollo de nuestra página web no nos hemos encontrado ningún tipo de problema para ajustarnos a este criterio debido a que el prototipo ya cumple con el nivel más alto de contraste, relativo al nivel AAA del estándar WCAG 2.0.

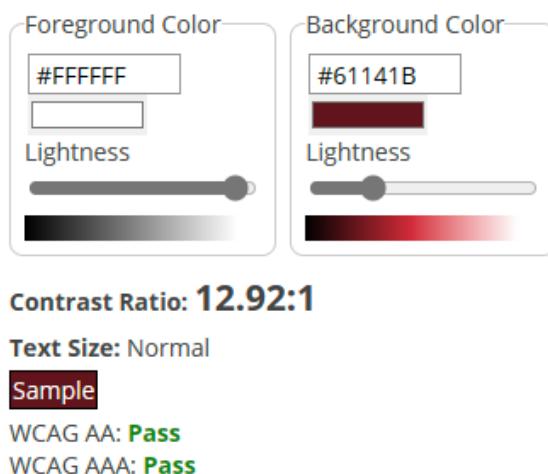


Figura 31: Validación WAVE: Resultado satisfactorio (AAA)

### 4.2. Aspecto general de accesibilidad

Es aspecto general de accesibilidad que se ha considerado al desarrollar nuestra página web es el de la estructuración.

Una buena estructuración es fundamental a la hora de desarrollar una buena página web, ya que, aparte de ser una buena práctica, ayuda a posicionar la web de mejor manera en los motores de búsqueda.

En nuestro caso, hemos estructurado de forma correcta nuestra web desde el comienzo de su creación, habiendo hecho uso de etiquetas tales como `<header>`, `<footer>`, `<nav>` o `<main>`.

# Parte II

## Hito 2

### 5. Introducción

Este hito 2, además de intentar cumplir con los requerimientos establecidos para esta parte, intenta sufragar las carencias de documentación y desarrollo existentes en el anterior hito.

A razón de la decisión motivada por la falta de trabajo equilibrada, se decidió continuar el proyecto de forma individual.

### 6. Modificaciones respecto al Hito 1

En esta sección se detallan algunos cambios mínimos visuales y funcionales respecto al anterior hito.

#### 6.1. Cambios comunes

##### 6.1.1. Imágenes

Todas las imágenes han sido eliminadas del almacenamiento local, para alojarlas en el servicio web *imgur*.

##### 6.1.2. Cambios visuales en el footer

Se ha resaltado los títulos de ambas secciones del footer, mejorando así la visibilidad de estas.



##### 6.1.3. Protección de las URLs

Toda página que necesite de parámetros pasados por la URL para mostrar información, seguirá esta metodología:

Si el usuario introduce manualmente los parámetros y estos son incorrectos, que es el escenario más común debido a la complejidad de estos, mostrará un mensaje por pantalla indicando que no introduzca la URL manualmente.



#### 6.1.4. Títulos de las secciones

Los títulos de las secciones han sido retocados visualmente, para conseguir un estilo común y un acercamiento mayor al prototipado original. El cambio en cuestión es introducir un elemento lineal a cada lado del título, además de establecer un formato en negrita para cada uno de ellos.

The screenshot shows the 'CARTELERA ELMOCINES' section of the Elmo Cines website. The page has a dark red header with the 'ELMO CINES' logo and navigation links for 'INICIO', 'CARTELERA', 'NUESTRAS SALAS', 'PRECIOS Y MENÚS', and '¿CÓMO LLEGAR?'. Below the header is a horizontal line with the text 'CARTELERA ELMOCINES' centered above it. The main content area displays a grid of movie posters. The top row includes 'Wicked: Part I', 'Argylle', 'Wonka', and 'Dune'. The bottom row includes 'Dune: Part II', 'Madame Web', 'One Love', and 'Kung Fu Panda 4'. Each poster is labeled with its title and a small description below it. The website has a dark red background color.

**SITEMAP**

- [Inicio](#)
- [Cartelera](#)
- [Nuestras salas](#)
- [Precios y menús](#)
- [¿Cómo llegar?](#)

**CONTACTO**

Elmo Cines ©  
info@elmocines.es  
Avda. Universidad de Cádiz,  
Campus Pto. Real, n.º 10, 11519

## 6.2. Página: Precios y menús

Esta página no figuraba en la documentación del anterior hito, por lo que a continuación se detalla su contenido: Precios para las distintas entradas y menús disponibles en el cine.

Respecto a la anterior versión, se ha solucionado la errónea alineación de los elementos en el contenedor. Además, se ha añadido los precios para cada tipo de entradas, todo esto consiguiendo responsividad en los dispositivos de pantalla pequeña.

### ENTRADAS

**Entrada general: 6 €**  
Podrás hacer la reserva de tus entradas en nuestra página web o comprárlas en taquilla.

**Entrada para menores de 18 años: 4 €**  
Deberás presentar tu DNI o algún documento que acredite tu edad en taquilla para poder comprar la entrada con descuento.

Si realiza la reserva online, deberá presentar el documento en taquilla para validar la entrada. De lo contrario, deberá abonar la diferencia en taquilla.



### MENÚS

**Better Together - 10 €**  
Incluye un bol de palomitas extra grande, una bolsa de gominolas y 2 bebidas grandes a elegir.

**Menú grande - 8 €**  
Incluye un bol de palomitas grande y una bebida grande a elegir.

**Menú mediano - 6 €**  
Incluye un bol de palomitas mediano y una bebida mediana a elegir.

**Menú infantil - 4 €**  
Incluye un bol de palomitas pequeño y una botella de agua.



## 6.3. Página: ¿Cómo llegar?

Esta página no figuraba en la documentación del anterior hito, por lo que a continuación se detalla su contenido: Indicaciones para el usuario de cómo llegar a las instalaciones de ElmoCines, acompañado un mapa de Google Maps.

Respecto a la anterior versión, se ha solucionado la errónea alineación de los elementos en el contenedor, manteniendo la correcta visualización en los dispositivos de pantalla pequeña.

### ¿CÓMO LLEGAR?

Nos puedes encontrar en Puerto Real, en la Avenida Universidad de Cádiz n.º 10, junto a la Escuela Superior de Ingeniería.

En los alrededores del establecimiento podrá encontrar un aparcamiento amplio y suficiente para albergar un aforo completo en nuestras instalaciones.

Interactúe con el mapa si necesita situarse u obtener una visión más amplia de la zona :)



## 6.4. Página: Reserva

### RESERVA DE ENTRADAS: Wicked: Part I

Nombre y apellidos del comprador  
Introduce tu nombre aquí  
El nombre debe contener al menos un nombre y un apellido, ambos empezando por mayúscula y sin números.

Correo electrónico \*  
Introduce tu correo electrónico aquí  
El correo debe tener el formato correo@dominio.com/es/...

Número de teléfono  
Introduce tu número de teléfono aquí  
El número de teléfono debe empezar por 6, 7, 8 o 9 y contener 9 dígitos.

Número de espectadores menores de 18 años \*\*  
0

Entrada 1: Menú  
Sin menú

Fecha  
2024-04-03

Sesión  
15:07

Butacas seleccionadas  
[1] A2

\* Proporcione una dirección de correo válida, ya que deberá confirmar la compra de entradas en dicha dirección. En caso de no confirmar la compra, las entradas serán liberadas para su venta en un plazo de 1 hora.

\*\* A la hora de acudir a la sala, se solicitará la documentación acreditativa de la edad de los menores de 18 años. En caso de no poder acreditar la edad, deberá abonar la diferencia de precio en taquilla.

---

**RESUMEN**  
1x - Entradas adultas (6 €)  
Total: 6 €

**INFORMACIÓN**  
Nombre: Jesus Gomez  
Correo: jggomeztocino@gmail.com  
Teléfono: 612345678

Revisa los datos antes de seguir con la compra  
**CONTINUAR**

### RESERVA DE ENTRADAS: Wicked: Part I

Nombre y apellidos del comprador  
Jesus Gomez

Correo electrónico \*  
jggomeztocino@gmail.com

Número de teléfono  
612345678

Número de espectadores menores de 18 años \*\*  
0

Entrada 1: Menú  
Menú Better Together (10 €)

Fecha  
2024-04-03

Sesión  
15:07

Butacas seleccionadas  
[1] A1

\* Proporcione una dirección de correo válida, ya que deberá confirmar la compra de entradas en dicha dirección. En caso de no confirmar la compra, las entradas serán liberadas para su venta en un plazo de 1 hora.

\*\* A la hora de acudir a la sala, se solicitará la documentación acreditativa de la edad de los menores de 18 años. En caso de no poder acreditar la edad, deberá abonar la diferencia de precio en taquilla.

---

**RESUMEN**  
1x - Entradas adultas (6 €)  
Entrada 1: Menú Better Together (10 €)  
Total: 16 €

**INFORMACIÓN**  
Nombre: Jesus Gomez  
Correo: jggomeztocino@gmail.com  
Teléfono: 612345678

Revisa los datos antes de seguir con la compra  
**CONTINUAR**

Se ha solucionado los errores visuales y de funcionalidad:

- Se ha implementado una correcta validación para cada uno de los datos introducidos.
- Se muestra correctamente la información proveniente de la página de antesala a la reserva.
- Se muestra a tiempo real el resumen e información de la reserva, siempre y cuando los datos sean correctos.
- La página es visualmente responsive en dispositivos de pantalla pequeña.

#### 6.4.1. Validaciones JavaScript

Se han implementado las siguientes validaciones:

- **Nombre:** El nombre debe contener, exactamente, un nombre y un apellido. Estos deberán empezar por mayúscula y no contener números ni tildes.
- Correo: El correo debe ser del formato *nombre@dominio.com*, pudiéndose reemplazar la extensión *.com* por *.es*.

Las validaciones de los inputs se realizan mediante el siguiente código JavaScript:

```
function validarNombre() {
    let nombre = document.querySelector('#nombre').value;
    if (!nombre.match(/^ [A-Z] [a-z]+ \s [A-Z] [a-z]+$/)) {
        mostrarErrorValidacion('#nombre',
            'El nombre debe contener, exactamente, un nombre y un apellido,
            ambos empezando por mayúscula, sin números y sin tildes.');
        return false;
    } else {
        eliminarErrorValidacion('#nombre');
        return true;
    }
}

function validarCorreo() {
    let email = document.querySelector('#correo').value;
    if (!email.match(/^\^ @ \s + @ \^ @ \s + \. (com | es) (\^ / \S *) ? $ /)) {
        mostrarErrorValidacion('#correo', 'El correo debe tener el formato
            correo@dominio.com / es');
        return false;
    } else {
        eliminarErrorValidacion('#correo');
        return true;
    }
}
```

```
function validarTelefono() {  
    let telefono = document.querySelector('#telefono').value;  
    if (!telefono.match(/^6789]\d{8}$/)) {  
        mostrarErrorValidacion('#telefono', 'El número de teléfono debe empezar  
        por 6, 7, 8 o 9 y contener 9 dígitos.');//  
        return false;  
    } else {  
        eliminarErrorValidacion('#telefono');//  
        return true;  
    }  
}
```

Estas funciones tienen un *listener* para cada vez que se realice un cambio en los correspondientes inputs, validen los datos y muestren o eliminen los mensajes de error asociados.

Si se intentase reservar sin cumplir los requisitos de los distintos inputs, el usuario será alertado con un mensaje indicando que los corrija.

## 7. Páginas no documentadas en el anterior hito

A continuación, se procede a la documentación de aquellas páginas que tampoco se documentaron en el anterior hito, pero que no han sufrido modificaciones.

### 7.1. Página: Pre-Reserva

En esta página el usuario podrá comprobar la disponibilidad de butacas para las distintas sesiones programadas.

#### 7.1.1. Determinación de sesiones disponibles

Inicialmente, la página plantea las siguientes situaciones y respuestas:

1. No hay ninguna sesión programada.
  - Se muestra una pantalla informando de esta situación.
2. Hay sesiones programadas anteriores a la fecha de hoy.
  - El sistema las descarta (no aparecen como opción en el desplegable).
3. Hay sesiones programadas con fecha igual o posterior a la del sistema, pero sin butacas disponibles.
  - El sistema las descarta (no aparecen como opción en el desplegable).
4. Hay sesiones programadas con fecha igual o posterior a la del sistema y con butacas disponibles.
  - El sistema las muestra como opción en el desplegable.

Por defecto, siempre se elegirá la sesión más próxima y disponible a la hora de la consulta del usuario.

#### 7.1.2. Mapa de asientos

Cada vez que se elige una fecha y una sesión, incluida la por defecto, se dibuja un mapa de asientos en JavaScript, que permite la selección y des-selección de asientos. Los asientos que se elijan y se eliminen de la selección se mostrarán, en tiempo real, para informar al usuario de su elección.

Además, la selección de asientos está limitado a 5 por compra, informando al usuario cuando intente seleccionar más de 5.

Una vez se seleccione, al menos un asiento, se mostrará un botón que permita continuar el proceso de compra.

El dibujado del mapa se realiza mediante la siguiente función:

```
function mostrarMapaDeAsientos(butacas) {
    butacasSeleccionadas = [];
    // Reinicia las butacas seleccionadas al cambiar de sesión
    actualizarButacasSeleccionadas();
    // Actualiza visualmente las butacas seleccionadas

    const mapaButacas = document.querySelector('.mapaButacas');
    mapaButacas.innerHTML = '<div class="pantalla">PANTALLA</div>';
    mapaButacas.style.margin = '20px 0';
    mapaButacas.style.paddingBottom = '10px';

    for (let i = 0; i < 6; i++) {
        const fila = document.createElement('div');
        fila.style.display = 'flex';
        fila.style.justifyContent = 'center';

        for (let j = 0; j < 5; j++) {
            const index = i * 5 + j;
            if (index < butacas.length) {
                const asiento = document.createElement('div');
                asiento.className = `asiento ${butacas[index].estado}`;
                asiento.textContent = butacas[index].numero;
                asiento.onclick = () =>
                    seleccionarAsiento(asiento, butacas[index]);
                fila.appendChild(asiento);
            }
        }
        mapaButacas.appendChild(fila);
    }

    // Leyenda de colores para los asientos
    const leyenda = document.createElement('div');
    leyenda.className = 'leyenda';
    leyenda.style.color = 'black';
    leyenda.innerHTML =
        `<div><div class="color" style="background-color:
        var(--color-ayiliar);"></div>Libre</div>
        <div><div class="color" style="background-color:
        var(--color-terciario);"></div>Ocupado</div>
        <div><div class="color" style="background-color:
        var(--color-seleccionado);"></div>Seleccionado</div>
        `;
    mapaButacas.appendChild(leyenda);
}
```

### 7.1.3. Resultado

Wicked: Part I

PANTALLA

A1	A2	A3	A4	A5
B1	B2	B3	B4	B5
C1	C2	C3	C4	C5
D1	D2	D3	D4	D5
E1	E2	E3	E4	E5
F1	F2	F3	F4	F5

■ Libre   ■ Ocupado   ■ Seleccionado

Fecha de la sesión:

2024-04-11

Butacas seleccionadas:

[5] D3, C3, C2, C1, B1

Hora de la sesión:

15:15

Revisa los datos antes de seguir con la compra

**CONTINUAR**

\* El número de entradas está limitada a 5 por compra.

## 8. API RESTful

Para implementar una API RESTful funcional y acorde al proyecto, se ha hecho uso de la tecnología Node.js, cuyo archivo de configuración *package.json* refleja el siguiente contenido:

```
{  
  "name": "elmocines",  
  "version": "1.0.0",  
  "description": "Dependencias para la API NodeJS de ElmoCines",  
  "main": "api.js",  
  "scripts": {  
    "start": "node api.js"  
  },  
  "dependencies": {  
    "dotenv": "^16.4.5",  
    "cors": "^2.8.5",  
    "express": "^4.19.2",  
    "mongodb": "^6.5.0",  
    "morgan": "^1.10.0",  
    "nodemon": "^3.1.0",  
    "nodemailer": "^6.0.0"  
  },  
  "author": "Jesús Gómez",  
  "license": "ISC"  
}
```

A continuación, se detalla el propósito de cada dependencia:

- **dotenv**: Esta dependencia se utiliza para cargar variables de entorno desde un archivo *.env* a *process.env*. Facilita la gestión de la configuración del proyecto sin necesidad de codificar valores sensibles directamente en el código fuente.
- **cors**: *Cross-Origin Resource Sharing* (CORS) es un mecanismo que permite a los recursos de una página web ser solicitados desde otro dominio distinto al del recurso original.
- **express**: Es un marco de aplicación web para Node.js, diseñado para construir aplicaciones web y API de forma rápida y fácil.
- **mongodb**: Esta dependencia proporciona un cliente MongoDB para Node.js, facilitando la conexión entre la aplicación Node.js y la base de datos MongoDB. Permite realizar operaciones de base de datos como insertar, buscar, actualizar y eliminar documentos en MongoDB.
- **morgan**: Es un middleware para la biblioteca Express que facilita el registro (logging) de solicitudes HTTP.
- **nodemon**: Permite reiniciar el servidor HTTP automáticamente cuando se detectan cambios en los archivos.
- **nodemailer**: Se utiliza para enviar correos electrónicos desde aplicaciones Node.js.

## 9. Operaciones RESTful

Para cada documento alojado en MongoDB Atlas, se han desarrollado las siguientes operaciones:

1. Recuperar todas las reservas.
2. Recuperar una única reserva existente por ID.
3. Añadir una nueva reserva.
4. Actualizar una reserva existente por ID.
5. Eliminar todas las reservas.
6. Eliminar una única reserva existente por ID.

### 9.1. Documento *users*

Este documento de MongoDB contiene los usuarios de ElmoCines.

Para cada usuario se almacena:

- **ID:** Correo electrónico
- **Nombre**
- **Número de teléfono**
- **Número de reservas**
- **Array de reservas**, para cada reserva:
  - ID de reserva: *pelicula-correo-fechahora*
  - Número de entradas adultas
  - Número de entradas infantiles
  - Array de asientos
  - Array de menús
  - Total de la compra

### 9.1.1. Formato del documento

El formato, para cada usuario, es el siguiente:

```
{  
    "_id": "jesus.gomeztocino@alum.uca.com",  
    "name": "Jesus Gomez",  
    "phone": "612345678",  
    "n_bookings": {"$numberInt": "1"},  
    "bookings": [  
        {  
            "booking_id": "dune2-jesus.gomeztocino@alum.uca.com-202404061634",  
            "adults": {"$numberInt": "1"},  
            "children": {"$numberInt": "1"},  
            "seats": [  
                "F1",  
                "F2"  
            ],  
            "menus": [  
                "entrada1-betterTogether"  
            ],  
            "total": {"$numberInt": "20"}  
        }  
    ]  
}
```

### 9.1.2. Operaciones

Se detallan las operaciones para este documento, cuyo formato y sintaxis ha sido replicado para los demás documentos (por eso, solo se detalla para este documento). La razón de mostrar estas operaciones en cuestión es porque se hace uso de *nodemailer*, que permite enviar correos al usuario que se registra, y, por tanto, hace una reserva, y al que realiza una nueva reserva.

```
// Ruta GET para obtener todos los usuarios  
router.get('/', async (req, res) => {  
    try {  
        await client.connect();  
        const db = client.db(dbName);  
        const users = await db.collection('users').find({}).toArray();  
        res.json(users);  
    } catch (error) {  
        res.status(500).send('Error al obtener los usuarios: ' + error.message);  
    } finally {  
        await client.close();  
    }  
});
```

```

// Ruta GET para obtener un usuario por ID
router.get('/id', async (req, res) => {
    try {
        const userId = req.params.id;
        await client.connect();
        const db = client.db(dbName);
        const user = await db.collection('users').findOne({ _id: userId });
        if (user) {
            res.json(user);
        } else {
            res.status(404).send('Usuario no encontrado');
        }
    } catch (error) {
        res.status(500).send('Error al obtener el usuario: ' + error.message);
    } finally {
        await client.close();
    }
});

// Ruta POST para añadir un nuevo usuario
router.post('/', async (req, res) => {
    try {
        const newUser = req.body;
        await client.connect();
        const db = client.db(dbName);
        const result = await db.collection('users').insertOne(newUser);
        if (result.acknowledged) {
            const welcomeEmailHtml =
`<p>¡Hola ${newUser.name}!</p>
<p>¡Bienvenido a ElmoCines!
Estamos ElmoCionados ;) de tenerte con nosotros.</p>`;
            await sendEmail(
                newUser._id,
                '¡Bienvenido a ElmoCines!',
                welcomeEmailHtml
            );
        }
        res.status(201).send('Usuario añadido correctamente');
    } catch (error) {
        res.status(500).send('Error al añadir el usuario: ' + error.message);
    } finally {
        await client.close();
    }
});

```

```

// Ruta PUT para actualizar un usuario
router.put('/:id', async (req, res) => {
  try {
    const userId = req.params.id;
    const updateData = req.body;
    await client.connect();
    const db = client.db(dbName);
    const result = await db.collection('users').
    updateOne({ _id: userId }, { $set: updateData });

    if (result.matchedCount === 0) {
      res.status(404).send('Usuario no encontrado');
    } else {
      const booking = updateData.bookings[updateData.bookings.length - 1];

      // ID sesión: movie-userid-datetime
      // --> Date: 8 dígitos (YYYYMMDD) + Time: 4 dígitos (HHMM)

      // Seleccionar la película de la reserva a partir del
      ID de la reserva y su correspondiente en el diccionario
      let movie = booking.booking_id.split('-')[0];
      movie = peliculas[movie];

      // Seleccionar la fecha a partir del
      ID de la reserva y formatearla a DD/MM/YYYY
      let date = booking.booking_id.split('-')[2];
      date = `${date.slice(6, 8)}-${date.slice(4, 6)}-${date.slice(0, 4)}`;

      // Seleccionar la hora a partir del
      ID de la reserva y formatearla a HH:MM
      let time = booking.booking_id.split('-')[2];
      time = `${time.slice(8, 10)}:${time.slice(10, 12)}`;

      // Formatear los menús a su correspondiente en el diccionario
      let menus = booking.menus.map(menuCode => {
        const menuKey = menuCode.split('-')[1];
        return menusDiccionario[menuKey];
      });
      const bookingEmailHtml = `
        <p>¡Hola ${updateData.name}!</p>
        <p>Has realizado una reserva para la película: ${movie}</p>
        <p>Fecha: ${date}, Hora: ${time}</p>
        <p>Menús seleccionados: ${menus.join(', ')})</p>
        <p>Total: ${booking.total} €</p>
        <p>
          <a href="https://localhost:3000/html/confirmed.html">
            Confirma tu compra aquí</a>
        </p>
      `;
      await sendEmail(userId, 'Confirmación de reserva - ElmoCines', bookingE
    }
  }
}

```

```

        res.send('Usuario actualizado correctamente');
    }
} catch (error) {
    res.status(500).send('Error al actualizar el usuario: ' + error.message);
} finally {
    await client.close();
}
});

// Ruta DELETE para eliminar un usuario por ID
router.delete('/:id', async (req, res) => {
    try {
        const userId = req.params.id;
        await client.connect();
        const db = client.db(dbName);
        const result = await db.collection('users').deleteOne({ _id: userId });
        if (result.deletedCount === 0) {
            res.status(404).send('Usuario no encontrado');
        } else {
            res.send('Usuario eliminado correctamente');
        }
    } catch (error) {
        res.status(500).send('Error al eliminar el usuario: ' + error.message);
    } finally {
        await client.close();
    }
});

// Ruta DELETE para eliminar todos los usuarios
router.delete('/', async (req, res) => {
    try {
        await client.connect();
        const db = client.db(dbName);
        const result = await db.collection('users').deleteMany({});
        if (result.deletedCount === 0) {
            res.status(404).send('No se encontraron usuarios para eliminar.');
        } else {
            res.send(`Usuarios eliminados correctamente. Total: ${result.deletedCount}`);
        }
    } catch (error) {
        res.status(500).send('Error al eliminar los usuarios: ' + error.message);
    } finally {
        await client.close();
    }
});

```

Donde se ha de hacer especial mención a los endpoints PUT y POST, ya que gestionan el envío de correos. Como se comentaba anteriormente, este es el modelo seguido para los demás documentos de la base de datos.

### 9.1.3. Gestión usuarios

Para la gestión de usuarios, se ha implementado una interfaz CRUD que permite realizar las 6 operaciones definidas.

The screenshot shows a user management interface with the following details:

User Information		Actions	
Rafael Leal	rafaelrleal@gmail.com	ELIMINAR	EDITAR
Jesus Gomez	jesus.gomeztoledo@alum.uca.es	ELIMINAR	EDITAR
Marina Ramirez	marinaramirez@gmail.com	ELIMINAR	EDITAR
Roberto Rivero	roberto@gmail.com	ELIMINAR	EDITAR
Concha Entro	holasoyconcha@entro.com	ELIMINAR	EDITAR

Below the table are two buttons: "ELIMINAR TODOS LOS USUARIOS" and "MOSTRAR TODOS LOS USUARIOS".

## 9.2. Documento *movies*

Este documento de MongoDB contiene la cartelera de ElmoCines, desde la que se carga en las distintas páginas web la información correspondiente.

### 9.2.1. Formato del documento

El formato, para cada película, es el siguiente:

```
[  
  {  
    "_id": "wicked",  
    "title": "Wicked: Part I",  
    "cover_url": "https://i.imgur.com/rGkbqrw.jpg",  
    "directors": "Jon M. Chu",  
    "actors": [  
      "Cynthia Erivo",  
      "Ariana Grande",  
      "Michelle Yeoh",  
      "Jeff Goldblum",  
      "Jonathan Bailey",  
      "Ethan Slater",  
      "Marissa Bode",  
      "Bowen Yang",  
      "Bronwyn James",  
      "Keala Settle"  
    ],  
    "duration": 165,  
    "synopsis": "Una amistad improbable surge entre Elphaba y Glinda, dos brujas enfrentadas en la mágica tierra de Oz. Elphaba es injustamente catalogada como una bruja malvada mientras intenta exponer al farsante y corrupto Mago que gobierna la tierra.",  
    "trailer":  
      "https://www.youtube-nocookie.com/embed/F1dvX9Vs0ns?si=T201ceF16vLVLArM"  
  }  
]
```

### 9.3. Documento *pelicula-sesiones*

Existe un documento para cada película que conste en la cartelera, cambiando 'pelicula' por el ID de la película en cuestión.

Cada documento de MongoDB contiene las sesiones programadas para una película, desde la que se carga en las distintas páginas web la información correspondiente.

#### 9.3.1. Formato del documento

El formato, para cada película, es el siguiente:

```
{"_id": "20240401-1511", "fecha": "2024-04-01", "hora": "15:11",  
  "sala": {"$numberInt": "6"}, "butacas_libres":  
  {"$numberInt": "1"}, "butacas_detalles":  
  [{"numero": "A1", "estado": "ocupado", "email": "teddy_craven@guests.elmocines.com"},  
   {"numero": "A2", "estado": "ocupado", "email": "dale_girouard@guests.elmocines.com"},  
   {"numero": "A3", "estado": "ocupado", "email": "minnie_hernandez@guests.elmocines.com"},  
   {"numero": "A4", "estado": "ocupado", "email": "michael_linares@guests.elmocines.com"},  
   {"numero": "A5", "estado": "ocupado", "email": "william_oestreich@guests.elmocines.com"},  
   {"numero": "B1", "estado": "ocupado", "email": "raymond_brooks@guests.elmocines.com"},  
   {"numero": "B2", "estado": "ocupado", "email": "mary_sonnenberg@guests.elmocines.com"},  
   {"numero": "B3", "estado": "ocupado", "email": "jeanne_callan@guests.elmocines.com"},  
   {"numero": "B4", "estado": "ocupado", "email": "james_richmond@guests.elmocines.com"},  
   {"numero": "B5", "estado": "ocupado", "email": "peter_smith@guests.elmocines.com"},  
   {"numero": "C1", "estado": "ocupado", "email": "meghan_mcmanemy@guests.elmocines.com"},  
   {"numero": "C2", "estado": "ocupado", "email": "lara_vaughn@guests.elmocines.com"},  
   {"numero": "C3", "estado": "ocupado", "email": "maryann_fox@guests.elmocines.com"},  
   {"numero": "C4", "estado": "ocupado", "email": "chad_peña@guests.elmocines.com"},  
   {"numero": "C5", "estado": "ocupado", "email": "betty_ryan@guests.elmocines.com"},  
   {"numero": "D1", "estado": "ocupado", "email": "frank_gilson@guests.elmocines.com"},  
   {"numero": "D2", "estado": "ocupado", "email": "leo_smith@guests.elmocines.com"},  
   {"numero": "D3", "estado": "ocupado", "email": "joan_villanueva@guests.elmocines.com"},  
   {"numero": "D4", "estado": "ocupado", "email": "william_grayson@guests.elmocines.com"},  
   {"numero": "D5", "estado": "ocupado", "email": "laura_anderson@guests.elmocines.com"},  
   {"numero": "E1", "estado": "ocupado", "email": "mary_irizarry@guests.elmocines.com"},  
   {"numero": "E2", "estado": "ocupado", "email": "margaret_walls@guests.elmocines.com"},  
   {"numero": "E3", "estado": "ocupado", "email": "william_delarosa@guests.elmocines.com"},  
   {"numero": "E4", "estado": "ocupado", "email": "pedro_kunkel@guests.elmocines.com"},  
   {"numero": "E5", "estado": "ocupado", "email": "steve_williams@guests.elmocines.com"},  
   {"numero": "F1", "estado": "ocupado", "email": "mitchell_howell@guests.elmocines.com"},  
   {"numero": "F2", "estado": "ocupado", "email": "damian_wolfe@guests.elmocines.com"},  
   {"numero": "F3", "estado": "libre"},  
   {"numero": "F4", "estado": "ocupado", "email": "alan_crippen@guests.elmocines.com"},  
   {"numero": "F5", "estado": "ocupado", "email": "cleo_wilson@guests.elmocines.com"}]}
```

Para generar las distintas sesiones programadas para cada película, y mostrar un mapa de asientos poblado y que refleje distintas situaciones, se hace uso de un script desarrollado en Python. Este script se mostró anteriormente en el documento, pero para adaptarse al formato de MongoDB ha sufrido ligeras modificaciones:

```

import random
from datetime import datetime, timedelta
import json
import names # https://pypi.org/project/names/

def generar_butacas(max_butacas):
    """
    Genera una lista de butacas con estados aleatorios entre 'libre' y 'ocupado'.
    Las butacas ocupadas incluirán un email generado aleatoriamente.
    """
    filas = ['A', 'B', 'C', 'D', 'E', 'F']
    columnas = [1, 2, 3, 4, 5]
    butacas = []
    for fila in filas:
        for columna in columnas:
            butaca = {"numero": f"{fila}{columna}", "estado": "libre"}
            butacas.append(butaca)

    butacas_ocupadas = random.sample(butacas, max_butacas)
    for butaca in butacas_ocupadas:
        butaca["estado"] = "ocupado"
        nombre_completo = names.get_full_name().lower().replace(' ', '_')
        # Genera un nombre completo, lo convierte a minúsculas y reemplaza
        # espacios por guiones bajos
        butaca["email"] = f"{nombre_completo}@guests.elmocines.com"

    return sorted(butacas, key=lambda x: x["numero"])

def generar_sesiones(fecha_inicio, fecha_fin, max_sala, total_butacas,
sesiones_por_dia, hora_inicio=15, probabilidad_dia_vacio=0.1):
    """
    Genera sesiones aleatorias entre las fechas de inicio y fin,
    con detalles como la hora de inicio, sala,
    y butacas disponibles/ocupadas.
    """
    sesiones = []
    fecha_actual = fecha_inicio

    while fecha_actual <= fecha_fin:
        if random.random() > probabilidad_dia_vacio:
            # Probabilidad de generar sesiones en un día determinado
            horario_sesion = hora_inicio
            for _ in range(sesiones_por_dia):
                if horario_sesion >= 24:

```

```

        break
minuto = random.randint(0, 59)
hora = f"horario_sesion:{02d}:{minuto:02d}"
sala = random.randint(1, max_sala)
butacas_libres = random.randint(0, total_butacas)
id_sesion =
f"fecha_actual.strftime('%Y%m%d')-{hora.replace(':', '')}"
# ID único de la sesión
sesion = {
    "_id": id_sesion,
    "fecha": fecha_actual.strftime("%Y-%m-%d"),
    "hora": hora,
    "sala": sala,
    "butacas_libres": butacas_libres,
    "butacas_detalles": generar_butacas(total_butacas - butacas_libres)
}
sesiones.append(sesion)
horario_sesion +=
random.randint(1, (24 - hora_inicio) // sesiones_por_dia)

fecha_actual += timedelta(days=1)

return sesiones

```

```

# Datos para la generación de sesiones
id_peliculas =
['wicked', 'argylle', 'wonka', 'dune', 'dune2', 'kung-fu-panda-4',
'one-love', 'madame-web']
fecha_inicio = datetime(2024, 4, 1)
fecha_fin = datetime(2024, 4, 30)
max_sala = 15
total_butacas = 30 # Total de butacas por sesión
sesiones_por_dia = 5
hora_inicio_sesiones = 15 # Hora de inicio de las sesiones (15:00)
probabilidad_dia_vacio = 0.1 # Probabilidad de que un día no tenga sesiones

```

Los cambios implementados están motivados por:

- Se hace uso de la librería *names* para poblar las reservas con nombres distintos.
- Adaptación al formato MongoDB para una exportación directa.