

# Descobreix Teatre

## Práctica de Tecnología Multimedia

**Resumen**— Desarrollo de una página web dinámica que mostrará los teatros disponibles en las islas de Mallorca, Eivissa y Formentera. Además, dispondrá de un catálogo de obras de teatro que se representarán dentro de poco. Por último, incorporar una sección “sobre nosotros” y un mapa que mostrará la ubicación de los teatros u otros eventos haciendo uso de JSON de compañeros.

**URL web**— “<https://www.descobreixteatre.com>”

**URL github público**— “<https://github.com/RamienM/Descobreix-Teatre>”

**Icono Propio para el aplicativo** (intento de la bandera de Baleares con unas máscaras)



### —Quienes Somos—

Nombre	DNI	Correo
Arnaud Vidal	43467666N	arnau.vidal1@estudiant.uib.cat
Sílvia Moià Canyelles	43465685D	silvia.moià1@estudiant.uib.cat
Rubén Ramis Martínez	46390261C	ruben.ramis1@estudiant.uib.eu

*Indice—*

<b>1. Funcionalidades de la Web.....</b>	<b>3</b>
<b>2. Explicación del Diseño.....</b>	<b>6</b>
<b>3. Maquetación.....</b>	<b>11</b>
3.1. JavaScript.....	11
<b>4. Estructura del servidor.....</b>	<b>15</b>
<b>5. Formato de los Datos.....</b>	<b>17</b>
<b>6. API's.....</b>	<b>22</b>
<b>7. Librerías.....</b>	<b>23</b>
<b>8. Web-App.....</b>	<b>23</b>
<b>9. Web-Responsive.....</b>	<b>24</b>
<b>10. Web-Semántica.....</b>	<b>27</b>
<b>11. Rendimiento.....</b>	<b>27</b>
<b>12. Media.....</b>	<b>28</b>
<b>13. Conclusiones de la práctica.....</b>	<b>33</b>
<b>14. Posibles mejoras.....</b>	<b>33</b>
<b>15. Valoración.....</b>	<b>34</b>

# 1. Funcionalidades de la Web

En la web y web-app, como bien dice el nombre “Descobreix Teatre”, podemos encontrar información sobre los diferentes teatros de las Baleares.

Estos teatros aparecerán según la isla que se seleccione en el filtro (todas las islas, Mallorca, Eivissa o Menorca):

The screenshot shows a dropdown menu with the following options: Menorca, Totes les illes, Mallorca, Menorca (highlighted in blue), and Eivissa. Below the menu, there are three images of theater interiors: Teatre Des Born, Teatre Principal de Maó, and Sala Polivalent Albert Camus.

Una vez que el usuario selecciona un teatro, leyendo del JSON se actualizará la página web donde se mostrará información sobre el teatro. El diseño de la misma se explicará en el apartado “Explicación del Diseño”

También se ofrece un catálogo de las obras que se van a representar dentro de poco:

The screenshot shows a catalog of upcoming plays with the following details:

- Bulle À Bulle** (20€): A creation by Pep Bou and Martín Cattani, featuring a large, colorful bubble.
- Los Perros** (9€): A proposal about resistance, featuring a man in a green coat.
- Camí a l' Escola** (15€): A story about three sisters facing obstacles, featuring three women in a dark setting.

Las obras de teatro seleccionadas actualizarán el documento html leyendo el JSON para obtener información sobre la obra. En esta página se mostrará información de la obra, los actores de la misma y sus reviews.

La web también tiene las ubicaciones de los diferentes teatros y tendrá otros eventos de interés (jsons externos) en un mapa, con el objetivo de poder situar las diferentes ubicaciones de manera muy visual y cómoda. Además, permitirá localizar al usuario.

**Teatre.**

Inici Teatres Obres **Mapa events** Sobre nosaltres

MAPA

DESCOBREIX EVENTS

Este mapa interactivo muestra la costa mediterránea de España, desde Valencia hasta Murcia. Se visualizan varias islas y puntos de interés, cada uno marcado con un icono que indica su tipo: teatro (teatro clásico), obra (teatro contemporáneo) o evento (otro tipo de actividad). Los iconos están rodeados por círculos transparentes que representan una zona de influencia o alcance. Una barra desplegable permite seleccionar qué tipo de evento se desea visualizar: Teatres, Obras o Eventos. Una opción para localizar la posición del usuario se activa con el botón 'Ubicam'.

Seleccioni quin event vol visualitzar:

Teatres

Vols visualitzar la tèva posició?

Ubicam

Además, la página también cuenta con una sección de “Sobre nosotros” donde se podrá comprobar quienes han diseñado la web y una video propio.

**Teatre.**

Inici Teatres Obres Mapa events **Sobre nosaltres**

SOBRE NOSALTRES

QUI SOM ?

Som un grup que està cursant l'assignatura de Multimèdia del grau d'Enginyeria Informàtica Universitat de les Illes Balears.

✓ Rubén Ramis  
✓ Arnau Vidal  
✓ Silvia Moià

Una captura de pantalla de la sección 'Qui som?' de la página. Muestra la descripción del grupo, los nombres de los miembros y una imagen del interior de un teatro con un videojugador central.

Una captura de pantalla de la sección 'Sobre nosotros'. Abajo de la descripción y la lista de miembros, hay un videojugador central que parece reproducir un video.

Finalmente, hay un contador dinámico que muestra la cantidad de islas, localidades con teatros, teatros en total y obras en total:

**Contacte**

Telèfon: +34 622 22 22 22  
Correu: info@descobreixteatre.com

**Disponibilitat horària**

Dill-Div: 11AM - 14PM

**Advertència**

Aquesta pàgina web no ofereix informació encertada, si us plau no fer servir la informació com a referència

**Propòsit**

Pàgina web desenvolupada com a pràctica per a la assignatura 21755-Tecnologia Multimedia d'Enginyeria Informàtica de la Universitat de les Illes Balears



## **2. Explicación del Diseño**

**Plantilla bootstrap**— “<https://bootstrapmade.com/yummy-bootstrap-restaurant-website-template/>” . Hemos adaptado la plantilla a nuestras necesidades: hemos eliminado elementos que no necesitábamos y hemos añadido otros que nos hacían falta, por ejemplo, el selector y el mapa. Además, el css se ha modificado un poco para incorporar nuevos estilos para las imágenes, el mapa y el reproductor de audio. También se ha eliminado contenido que no se usaba.

- **Página principal**

La página principal ya se ha mostrado por encima para explicar las funcionalidades de la página web. Por tanto se hará un resumen rápido de lo que se puede encontrar.

La pantalla principal consta de cuatro partes:

1. Lo primero que vemos al entrar es una breve descripción de lo que es “Descobreix Teatre”.
2. Más abajo podemos ver una lista de todos los teatros disponibles, estos se pueden filtrar según la isla. Este filtro es dinámico, para cada teatro que coincide su atributo address.addressRegion del json con el nombre de la isla seleccionado, y se creará el código html correspondiente.
3. Si continuamos bajando llegamos al listado de obras representadas este mes. La obtención de estas obras sigue el mismo método que con los teatros pero en este caso se mira el mes del atributo startDate de cada evento.
4. Más adelante se encuentra un mapa con la ubicación de los diferentes teatros u eventos que el usuario desee consultar. Además incluye un botón que permitirá al usuario averiguar su ubicación. La ubicación, como se explicará más adelante, se guarda usando Local Storage.
5. En el siguiente apartado se puede apreciar una explicación de los integrantes que han realizado la página web.
6. Finalmente, se encuentra un contador dinámico que muestra la cantidad de teatros, obras, islas y localidades(con teatros) que hay en las Baleares.

- **Página de teatros**

En la siguiente página web lo primero que se puede apreciar es una imagen del teatro seleccionado con una descripción, ubicación y teléfono del mismo, si posee página web contendrá un enlace hacia ella y finalmente hay un reproductor de audio, los audios son propios, donde se lee la descripción del teatro:

**Teatre.**

# TEATRE PRINCIPAL DE PALMA

Antic teatre clàssic edificat el 1667 i reformat que ofereix espectacles d'òpera, sarsuela i dansa.

Palma - Mallorca

971219700

<https://www.teatreprincipal.com/ca/>



0:00 / 0:15

OBRES

En un apartado inferior se mostrarán todas las obras que tiene programadas el teatro seleccionado:

**Teatre.**

OBRES

## PRÒXIMES OBRES A LES BALEARIS



**Los Perros**  
**\$15**

Los Perros proposa assolar un estat de resistència, com a gossos que passegen, gossos que es troben, que es reconeixen vulnerables, que comparteixen a...



**La mala dicció**  
**\$8**

La mala dicció completa la Trilogia del Lament, amb la qual Jordi Oriol reinterpreta tres tragèdies shakespearianes posant la propia paraula al centre...



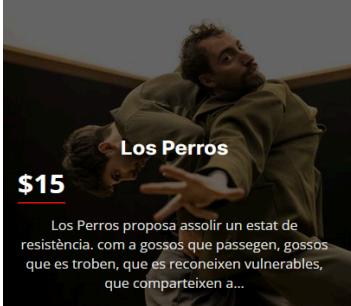
**Els dies bons**  
**\$10**

Els dies bons és la història d'una família travessada per les addiccions. I tot comença amb la intensa vida de Xavier, marcada per les drogues i les men...

En caso de que solo haya una obra se mostrará de la siguiente manera:

OBRES

## PRÒXIMES OBRES A LES BALEARIS



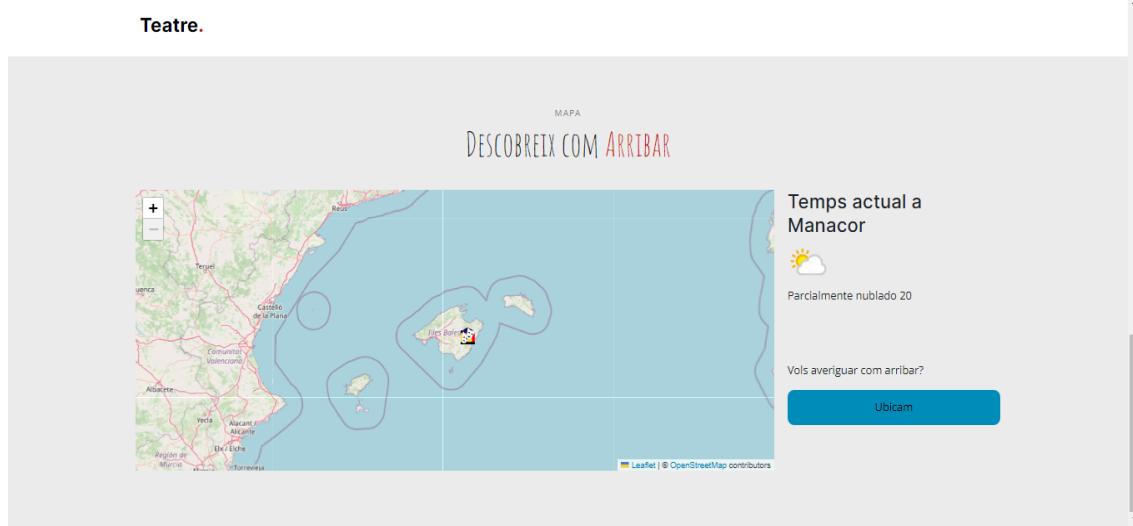
**Los Perros**  
**\$15**

Los Perros proposa assolar un estat de resistència, com a gossos que passegen, gossos que es troben, que es reconeixen vulnerables, que comparteixen a...



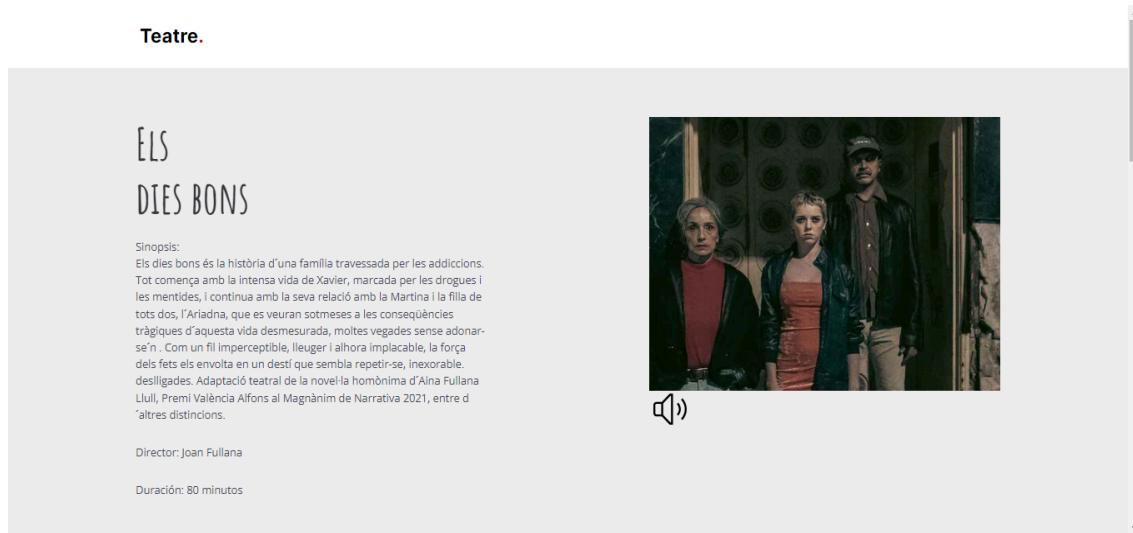


Finalmente, se puede encontrar una apartado donde se incluye un mapa con la ubicación del teatro. Este apartado además ofrece la posibilidad de conocer el tiempo en la zona del teatro y un botón que permite generar una ruta desde la posición del usuario.



#### ● *Página de obras*

En la siguiente página web lo primero que se puede apreciar es la imagen de la obra de teatro seleccionada con la sinopsis de la obra, quién ha sido su director y cuánto durará la obra. Además de incluir lectura automatizada de la descripción haciendo uso de SpeechSynthesis.



Posteriormente, se puede apreciar los actores que han participado en la obra tanto nombre como imagen.

**Teatre.**

## ÉLENCO DE ACTORES



Caterina Alorda

• • • •

Si continuamos bajando se puede apreciar el trailer de la obra de teatro, en caso de que tenga.

**Teatre.**

## TRAILER



Finalmente, en la última sección se muestran las opiniones de la gente. Donde las puntuación se mostrará con estrellas, el usuario que ha escrito la opinión y la opinión.

Se ha implementado la opción de añadir una review pero ésta sólo se podrá ver en la sesión en la que se ha creado. No se va a almacenar. Se debe a la utilización del web storage para la implementación de esta funcionalidad.

El web storage es una API que te permite guardar información solamente para la duración de la sesión. Es decir, mientras el navegador está abierto. Recargar la página no va a cambiar la sesión y se van a seguir viendo las reviews.

## REVIEW

“ Que recuerdos!!! ”  
**Haiter**  
★★★★★

DEIXA LA TEVA OPINIÓ:

El teu nom

★★★★★

La teva opinió.

Desa

## **3. Maquetación**

Para nuestro proyecto como se nos ha pedido desarrollar un página web dinámica solo hacemos uso de 1 html llamado index.html. Este html se va ir actualizando dependiendo del teatro u obra de teatro que seleccionamos.

De primeras el documento html contiene una cabecera donde se encuentran todos los enlaces hacia los estilos. Esta cabecera es común tanto para página principal, teatros y obras. También podemos encontrar el script necesario para poder usar Google Analytics, este script tal y como se pide por parte de google, se encuentra justo debajo de la etiqueta <body>. También se puede encontrar el script de [Web Semántica](#).

A partir de aquí ya se encuentran variaciones dependiendo de la acción que estemos realizando en la página web:

- **Página web principal**

Por defecto el documento html ya contiene todos los elementos necesarios para la generación de la página web. Pero para la hora de actualizar la página al volver de teatros o obras, se utiliza el JavaScript de “index.js”. Este JavaScript está comentado en el apartado *JavaScript index*.

Para que la lista de teatros y obras que aparezcan en la página principal el index contiene dos divs vacíos: “teatres” y “obras”. El objetivo es luego crear un archivo javascript que sobreesciba estos elementos con los jsons.

- **Página web teatros**

Para visualizar la página de los teatros se sobreescibe el html manteniendo únicamente el head. Este JavaScript está comentado en el apartado *JavaScript teatro*.

- **Página web obras**

El funcionamiento para visualizar las obras es el mismo que el de los teatros. Se explicará en el apartado *JavaScript obra*.

### **3.1. JavaScript**

En el siguiente apartado se pretende explicar el uso y que aportan los JavaScripts usados a nuestra página web.

- **JavaScript index**

Este JavaScript vuelve a escribir el index como estaba inicialmente el index.html. Se van cogiendo los diferentes elementos a sobreescibir: el header (no confundir con el head del html, eso no se toca), la info (que corresponde a la imagen principal con el texto) y el main (las diferentes secciones de la página: teatros, obras, mapa, sobre nosotros, el contador y el footer).

Para cada uno de estos elementos se escribe un string con el html original y se le asigna al html utilizando la propiedad innerHTML. Cabe destacar que, el div de *teatres* y *obras* que se ha comentado anteriormente que se dejaban vacíos, van a seguir dejándose vacíos al regenerar este código. De manera que cuando se

actualicé la página y se regenere el código html con este JS, la página va seguir siendo dinámica porque se van a seguir rellenando estos dos divs con los dos JS correspondientes.

El código de los dos divs acabados de mencionar es generado también en este archivo. Para los teatros hay una función “mostraTeatres()” que obtiene los teatros de la opción seleccionada en el filtro. Primero se va ha hacer un fetch de todos los jsons. En el caso que el filtrado esté en todos, la lista de jsons filtrados será igual a todos los obtenidos. En el caso que se haya elegido cualquiera otra opción se va a filtrar para que el campo address.addressRegion del json sea igual al valor del filtro. A continuación se va a generar el código de cada teatro para todos los teatros del json. Así consiguiendo una página dinámica.

Las obras funcionan más o menos igual. Para cada teatro recorremos sus obras y miramos si el mes actual corresponde al mes del campo startDate de la obra. Si es así lo mostramos. Cabe destacar que, como cada teatro tiene sus obras, puede ser que estas estén repetidas entonces, aparte de mirar el mes también se comprueba que sea una obra que no se haya pintado ya. Para eso hay un array con el nombre de las obras y se comprueba que la obra está en el array o no. Todo este proceso se encarga “mostraObres()”

- ***JavaScript teatro***

En este archivo se obtiene la información del json del teatro seleccionado. Se crea el html necesario para la visualización de la pantalla con la información específica del json. Así se consigue que dependiendo del teatro seleccionado se obtenga una información u otra, pero siempre llamando a este mismo JS que generará la misma plantilla de html.

La información que hace uso el siguiente JavaScript del Json es:

- Sección Header
  - Nombre del teatro
  - Descripción
  - La imagen del teatro
  - El teléfono
  - Si tiene página web añadir la url
  - Si tiene audio grabado por nosotros, se añade el reproductor de audio
- Sección Obras
  - Nombre de la obra
  - Imagen de la obra
  - Precio de la obra
  - Descripción de la obra (limitada a 100 caracteres, si los supera se añade “...” al final)
- Sección Mapa
  - Ubicación del teatro (sirve tanto para la API del tiempo como la representación del teatro en el mapa)

A destacar de este Javascript sería cómo se obtiene el tiempo. Aunque en el apartado de API's ya se explicará Weather API vamos a comentar como se ha implementado. Haciendo uso de nuestra KEY hemos hecho un fetch a la página de Weather API con las coordenadas que queremos consultar el tiempo.

Este nos ha devuelto un JSON con la información requerida. Haciendo uso de “generarHTMLWeatherTeatro(weather)” donde weather es el JSON que nos ha devuelto la API hemos mostrado:

- El icono del tiempo (solo, nubes, nubes con lluvia...)
- Un texto que indica si es soleado, está lloviendo o nevando por ejemplo.
- La temperatura

- ***JavaScript obra***

Este archivo funciona de manera análoga al descrito anteriormente, la diferencia es que se crea un html diferente, el que corresponde a la página de obras. Se coge la obra seleccionada del json y así se modifica la información dinámicamente. Las obras no son un json a parte, están dentro de cada teatro. Consecuentemente, no es tan inmediata la obtención de una obra ya que se tienen que ir recorriendo los jsons y para cada uno mirar sus obras hasta encontrar la que estamos buscando (hay un identificador en cada obra que tendrá que corresponder con el id de la obra a la cual se ha entrado). La información del JSON que se usa es la siguiente:

- Sección Header
  - Nombre de la obra
  - Sinopsis
  - Director de la obra
  - Duración de la obra
  - Imagen de la obra
- Sección Actores (Hay obras que no se han podido encontrar actores, el usuario verá un mensaje que pone “Actores no disponibles” en vez de las fotos)
  - Nombre del actor
  - Imagen del actor
- Sección Trailer (Hay obras que no se ha podido encontrar el trailer, el usuario verá un mensaje que pone “Trailer no disponible” en vez del reproductor de video)
  - Video del trailer (el video está almacenado en .mp4 y .webm, en el JSON solo está reflejado el .mp4, pero a la hora de usar el script del reproductor de vídeo, convertimos el url .mp4 a webm tambien, por lo que tenemos dos sources en caso de que uno no funcione)
- Sección de Reviews (En el apartado de reviews el usuario es capaz de añadir nuevas reviews gracias al uso de WebStorage)
  - Puntuación de la review
  - La opinión
  - Autor

- ***JavaScript Mapa***

El siguiente JavaScript es el encargado de generar los mapas tanto del index.html, index.js y teatros.js. En él tenemos dos funciones importantes, una que se encarga de generar el mapa para index y otra para

teatros. Ambas se comportan de manera similar, lo que se les diferencia es que en el mapa de index debe mostrar todos los iconos del filtro seleccionado, mientras que el del teatro solo muestra el seleccionado. Usando las coordenadas latitud y longitud ubicadas en cada teatro del JSON se puede representar.

Otra función importante es la de geoPosicion, esta permite localizar al usuario cuando este da al botón de “Ubicam” y acepta los permisos de geolocalización. Una vez localizado se mostrará en el mapa con el ícono correspondiente.

Finalmente, otra función clave es la de generación de rutas. Aportando las ubicaciones la librería “Leaflet Routing Machine” se encargará de generar una ruta en OpenStreetMap.

Aquí también se ha utilizadola API de web storage para poder almacenar la última ubicación de la sesión. En caso de que el usuario perdiera la conexión con la sesión abierta, seguiría guardada la ubicación de la última vez registrada.

- ***JavaScript Main***

El siguiente JavaScript ha sido proporcionado por la plantilla, simplemente se ha editado eliminando las funciones que no se usan y añadiendo funciones que recarguen ciertas funciones que necesitan ser actualizadas cada vez que actualizamos la página como el Swiper.

- ***JavaScript Service-Worker***

El siguiente JavaScript permite la conversión de nuestra página web a un aplicativo que puede ser descargado por los usuarios.

## 4. Estructura del servidor

Explicación de cómo están estructuradas nuestras carpetas en el servidor proporcionado por Dondominio, además de la explicación de los archivos necesarios para habilitar el CORS.

Nombre			Tamaño	Fecha	Permisos
..					
assets				Apr 25, 2024	drwxrwxr-x
lib				Apr 26, 2024	drwxrwxr-x
.htaccess			42B	Apr 23, 2024	-rw-rw-r--
index.html			147KB	May 4, 2024	-rw-rw-r--

- **Assets**

Assets es una carpeta que contiene todos los elementos necesarios para el correcto funcionamiento del aplicativo web. Como imágenes, javaScript, css y otros. Se mostrarán todos a continuación.

Nombre			Tamaño	Fecha	Permisos
..					
aud				Apr 25, 2024	drwxrwxr-x
css				Apr 25, 2024	drwxrwxr-x
img				May 4, 2024	drwxrwxr-x
js				Apr 29, 2024	drwxrwxr-x
json				Apr 29, 2024	drwxrwxr-x
vendor				Apr 25, 2024	drwxrwxr-x
vid				Apr 26, 2024	drwxrwxr-x

- **aud**

La siguiente carpeta contiene todos los audios implementados para leer las descripciones de los teatros.

- **vid**

La siguiente carpeta contiene todos los trailers de las obras de teatro y el video explicativo propio.

- **img**

La siguiente carpeta contiene todas las fotos necesarias para la página web, como son los svg, la foto de los actores, obras y teatros.

- ***js***

La siguiente carpeta contiene todos los códigos JavaScript necesarios para el correcto funcionamiento de la página web, la explicación de los mismos forma parte del apartado de “JavaScript”.

- ***json***

La siguiente carpeta contiene los json propios usados para el desarrollo de la página web, los json están explicados en el apartado de “Formato de Datos”

- ***vendor***

En la siguiente carpeta se incluyen todas las librerías que vienen con la plantilla de Yummy. Entre ellas se encuentra bootstrap y swiper.

- ***css***

La siguiente carpeta contiene la hoja de estilo principal de nuestra página web.

- ***Lib***

La siguiente carpeta contiene todas las librerías usadas para la página web, estas son leaflet y leaflet routing machine. La explicación de las librerías están explicadas en el apartado de “Librerías”

The screenshot shows a file manager interface with a dark header bar containing icons for back, forward, and search, along with the text "WebFTP" and a menu icon. Below the header is a breadcrumb navigation bar showing the path: public / lib. The main area is a table listing files and directories:

Nombre	Tamaño	Fecha	Permisos
..			
leaflet		Apr 25, 2024	drwxrwxr-x
leaflet-routing-machine		Apr 25, 2024	drwxrwxr-x

- ***index.html***

HTML principal de la página web, está explicado en el apartado de “Maquetación”.

- ***.htaccess***

Archivo que permite el acceso a otros usuarios a nuestro JSON.

## **5. Formato de los Datos**

En este apartado se explicará el formato de datos usado. En nuestro caso hemos usado JSON para nuestra página web. En todo el contenido del JSON se ha usado la estructura y estándar que presenta Schema su página web “<https://schema.org>”, a excepción de manifest.json.

En cada apartado se presenta el porque hemos considerado la correspondiente estructura, que atributos hemos usado y añadido, y finalmente, que contiene cada atributo del JSON de Teatros.

**Url JSON—”<https://www.descobreixteatre.com/assets/json/Teatre.json>”**

### **● *MovieTheater***

La estructura de Schema MovieTheater nos ha permitido representar el conjunto de teatros que hay en las Islas Baleares.

**name:** Nombre que recibe el teatro.

**identifier:** Identificador del Teatro

**address:** Ubicación del teatro usando la estructura PostalAddress de Schema, en esta se contempla la localidad, región y código postal del teatro.

**geo:** Coordenadas del teatro usando la estructura GeoCoordinates de Schema, se contempla la latitud y longitud.

**telephone:** Número de teléfono del teatro.

**image:** Imagen del teatro usando la estructura ImageObject de Schema, contiene la ubicación url de la imagen.

**description:** Texto que describe el teatro, ya sea historia o que actividades se realizan.

**url:** Url del teatro si este dispone de su propia página web

**event:** Conjunto de obras de teatros, la estructura usada es la de Event de Schema. En ella se encuentra toda la información de las obras.

### **● *Event***

La estructura de Schema Event nos ha permitido representar el conjunto de obras que tiene cada teatro.

**name:** Nombre de la obra.

**identifier:** Identificador de la obra

**startDate:** Fecha de inicio de la obra

**director:** Representación del director usando la estructura Person de Schema, en el apartado “5. Person” se explicará la información requerida del director.

**duration:** Duración de la obra de teatro. Se ha usado el ISO 8601 para la su correspondiente representación, como dicta Schema.

**actor:** Representación del conjunto de actores de las obras de teatro usando la estructura Persona de Schema. En el apartado “5. Person” se explicará la información requerida de los actores.

**image:** Imagen de la obra de teatro usando la estructura ImageObject de Schema, contiene la ubicación url de la imagen.

**additionalProperty**: Atributo que nos ha permitido añadir atributos que no se encuentran predefinidos en Schema. Se ha usado la estructura PropertyValue de Schema.

**description**: Texto que se encarga de describir la obra o explicar de que trata.

**review**: Conjunto de opiniones de ciertas personas sobre las otras. Se ha usado la estructura Review de Schema.

### ● ***Review***

La estructura de Schema Review nos ha permitido representar el conjunto de opiniones que tiene la gente respecto a una obra.

**reviewBody**: Opinión sobre la obra de teatro.

**reviewRating**: Usando la estructura Rating de Schema hemos representado la persona que ha publicado la review y que puntuación le ha otorgado.

### ● ***Rating***

La estructura de Schema Rating nos permite conocer quien ha escrito una opinión y que valoración numérica le ha aportado.

**author**: Nombre de la persona que añade su opinión de una obra. Se ha usado la estructura Persona de Schema. En el apartado “5. Person” se explicará la información requerida de estas personas.

**ratingValue**: Puntuación numérica que una persona ha aportado a una obra.

### ● ***Person***

La estructura de Schema Person nos ha permitido representar un conjunto de personas, ya sea de directores como de actores.

#### - **Directores y Personas que añaden sus opiniones**

De los directores/Personas que añaden sus opiniones solamente es necesario conocer su nombre.

**givenName**: Nombre del director/Persona que añade su opinión.

#### - **Actores**

De los actores nos interesa saber tanto su nombre como tener alguna imagen de ellos.

**givenName**: Nombre del actor.

**image**: Imagen del actor usando la estructura ImageObject de Schema, contiene la ubicación url de la imagen.

### ● ***ImageObject***

La estructura de Schema ImageObject nos ha permitido añadir todas las imágenes necesarias, ya sean de teatros, obras de teatros o personas.

**contentUrl**: Ubicación de donde está la imagen.

- ***PropertyValue***

La estructura de Schema PropertyValue nos ha permitido añadir todos los videos y precios de las obras.

**name**: Nombre de la propiedad

**value**: Valor de la propiedad

**url**: Ubicación de donde está el video. (Solo en el caso de los videos, los precios no contienen este atributo.)

- ***GeoCoordinates***

La estructura de Schema GeoCoordinate nos ha permitido añadir las coordenadas de donde está ubicado un teatro. Información bastante útil para mapas.

**latitude**: Latitud de donde se encuentra el teatro.

**longitude**: Longitud de donde se encuentra el teatro.

- ***PostalAddress***

La estructura de Schema PostalAddress nos ha permitido añadir la ubicación del teatro.

**addressLocality**: Localidad donde se encuentra el teatro

**addressRegion**: Región donde se encuentra el teatro

**postalCode**: Código postal del teatro.

Explicación del JSON manifest.json. Este JSON nos permite brindar información a una web app cuando se instala en los dispositivos de los usuarios.

**name**: Nombre completo de la web app.

**short\_name**: Nombre abreviado para la web app.

**description**: Descripción de la web app.

**start\_url**: Dirección url que le mostrará al usuario al abrir la web app.

**background\_color**: Se ha optado por un color blanco como fondo.

**theme\_color**: Se ha optado por un color negro como tema.

**icons**: Iconos necesarios para el correcto funcionamiento de la aplicación.

**display**: Hemos optado por un display “standalone”.

**scope**: Ámbito URL de nuestra aplicación a considerar para PWA.

**service\_worker**: dirección de donde se encuentra el service worker.

## - Validación del JSON

Usando [JSONLint](#) nos proporciona el siguiente resultado:

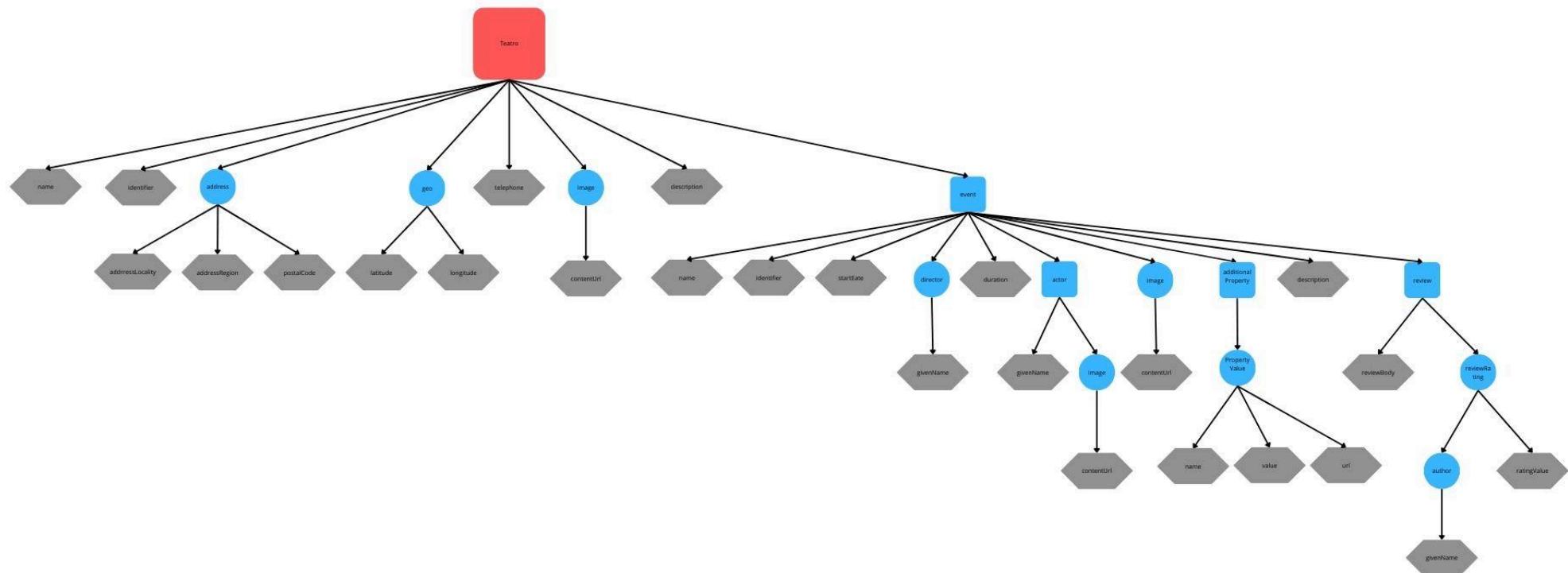
The screenshot shows the JSONLint interface. At the top, there is a code editor window displaying JSON code. Below the editor are three buttons: 'Validate JSON' (green), 'Clear' (white), and 'Compress' (white). At the bottom, a green bar displays the message 'JSON is valid!'. The background of the interface is dark.

Usando la validación de [Schema](#) para web semántica:

The screenshot shows the Schema.org validation interface. On the left, there is a code editor window displaying JSON-LD code. On the right, there is a detailed validation report titled 'ItemList' with sections for 'ItemList', 'itemListElement', and individual properties like '@type', 'name', 'identifier', etc. The report indicates '0 ERRORES 0 ADVERTENCIAS'. The background of the interface is red at the top and white below.

- **Estructura del JSON**

**Nota**— Los cuadrados representan un conjunto, los círculos hacen referencia a otra estructura y, finalmente, los polígonos son atributos.  
*Recomendado hacer Zoom en el pdf.*



## **6. API's**

En este apartado se explicarán las API's usadas en la web app:

- ***OpenStreetMap***

OpenStreetMap es una API que hemos utilizado para poder visualizar las ubicaciones de los teatros. Además, haciendo uso de los JSON de los compañeros nos permitirá mostrar la ubicación de las ferias y fiestas. Su implementación ha sido haciendo uso de la librería Leaflet.

- ***Geolocation (con getCurrentPosition())***

HTML 5 nos ofrece una API que nos permite obtener la ubicación del usuario, conociendo la ubicación podemos mostrar su ubicación en el mapa. La función “geoPosicion()” de mapa.js se encarga de este proceso.

- ***Weather Api***

Weather Api es una API que nos permite conocer el tiempo de una zona. Con esto podemos ofrecer al usuario el tiempo que hace en la zona del teatro que ha seleccionado. Su implementación está explicada en el apartado de teatre.js de “[3.1 JavaScript](#)”.

- ***Web Storage***

Web Storage es una api que permite guardar cierta información en la sesión (siempre que el navegador esté abierto, incluidas las recargas y restauraciones de la página). La hemos utilizado para guardar las reviews y la última ubicación del usuario de una sesión.

- ***Form API***

Form es una API que incluye HTML, esta API nos permite crear formularios de una manera simple. La utilizamos principalmente para la creación de reviews.

- ***SpeechSyntesis***

SpeechSyntesis es una API que nos permite leer de manera automática el título, descripción, director y duración de las diferentes obras de teatro. Su implementación se encuentra en obres.js y la función es “reproductorAudio(audio)” donde audio es un String que va a reproducir el SpeechSyntesis.

## **7. Librerías**

- ***Leaflet***

La librería leaflet nos permite acceder a la API de OpenStreetMap además de ofrecer el css necesario para la representación del mapa y marcadores.

- ***Bootstrap***

Bootstrap es una librería que nos ofrece una cantidad enorme de css para nuestro proyecto, permitiendo sobre todo que la página web sea responsive.

- ***Swiper***

La librería Swiper incluye el css y JavaScript necesario para permitir el cambio y movimiento de imágenes de las obras de teatros, autores de la obras y reviews.

- ***Leaflet Routing Machine***

Leaflet Routing Machine es una librería que permite generar una ruta sobre el mapa de OpenStreetMap, es necesario conocer los 2 puntos (punto inicial y punto destino). Haciendo uso de la función “generaRutaMapaTeatros(teatre)” de mapa.js, únicamente conociendo theatre que es el id del teatro, podemos trazar una ruta.

## **8. Web-App**

Como se ha ido mencionando en apartados anteriores, nuestra página web puede ser instalada en los dispositivos móviles\* gracias al uso de la web-app. Para poder crear el aplicativo ha sido necesario añadir un service-worker explicado en el apartado de [JavaScript](#) y de un manifiesto que está comentado en el apartado de [formato de datos](#).

\*: ¡¡Importante!! Ya no es posible descargar Web-Apps en los dispositivos de Apple. Adjunto noticia:

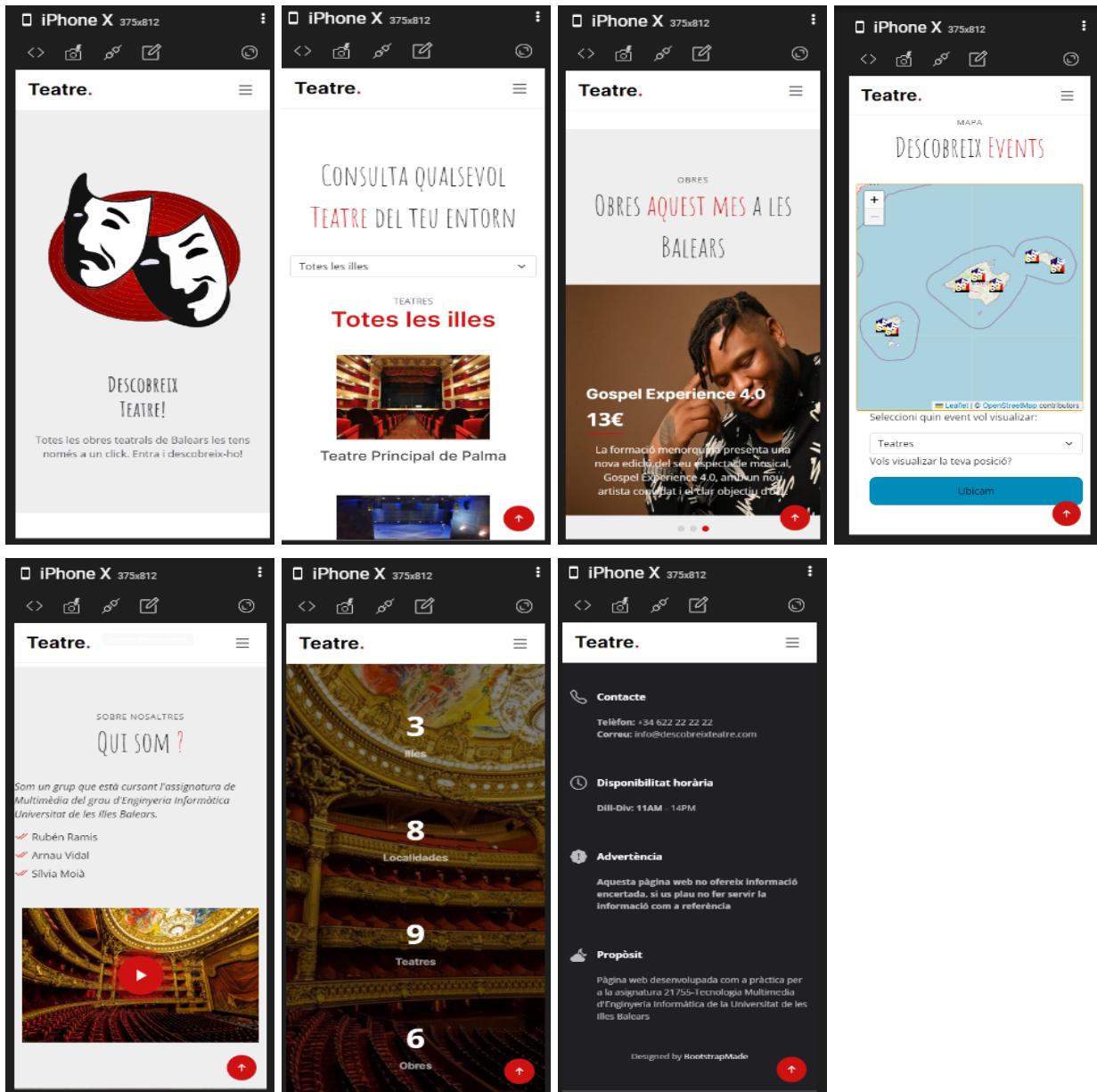
- <https://www.xataka.com/moviles/apple-mata-progressive-web-apps-iphones-ue-para-cumplir-dma-desarrolladores-no-estan-contentos>

## 9. Web-Responsive

Para mostrar que nuestra web es responsive nos hemos descargado la aplicación “Responsively App” que permite visualizar tu página web en varios dispositivos.

### • Iphone X

Mostraremos como se visualiza la pantalla principal en un iPhone:



### • Ipad

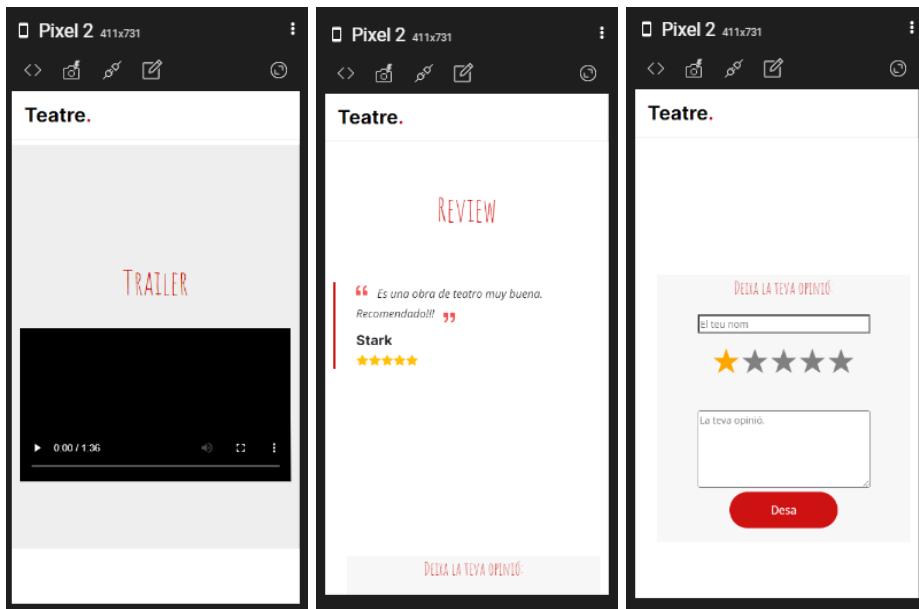
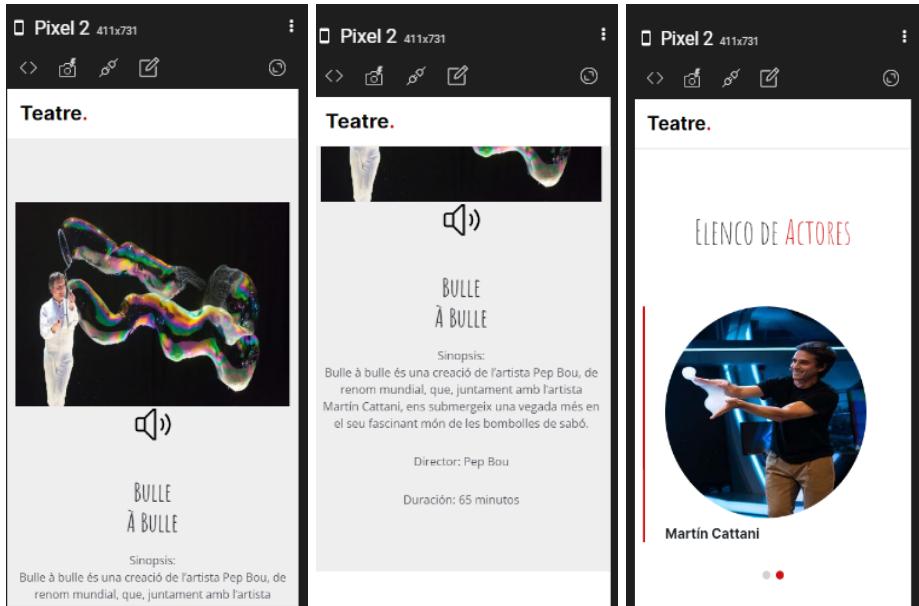
Mostraremos como se visualiza la pantalla de teatros en un iPad:

The screenshots show the following content:

- Screenshot 1:** Shows a video thumbnail of the theater interior with red seats and a chandelier. Below it is the title "TEATRE PRINCIPAL DE INCA".
- Screenshot 2:** Shows a larger view of the theater interior. Below the image is the title "TEATRE PRINCIPAL DE INCA" and a short description: "Aquell antic i històricament estimat Teatre de Inca ha quedat rejuvenit i posat al dia segons les noves maneres, formes i tecnologies actuals. Serà interessant fer una ullada cap enrere i conèixer el seu naixement. El Teatre de Inca s'inaugurà el dia 14 de novembre de l'any 1914, diada de la revetlla i completes de la festa de Santa Maria la Major i, també, pocs dies abans del Dijous Bo."
- Screenshot 3:** Shows a video thumbnail for the play "Els dies bons" by \$12. Below it is the title "PRÒXIMES OBRES A LES BALEARSG" and a snippet of the play.
- Screenshot 4:** Shows a map of Mallorca with Inca highlighted. To the right, there is weather information: "Temps actual a Inca" (Partially cloudy, 20°C), a "Ubicam" button, and a question "Vols averiguar com arribar?".

### ● Pixel 2

Mostraremos como se visualiza la pantalla de obras en un móvil Pixel:



## 10. Web-Semántica

En el apartado “[5. Formato de datos](#)” hemos comprobado que el JSON es correcto para ser añadido como semántica de nuestra web. Por tanto, explicaremos cómo se ha añadido la semántica.

Primero de todo haciendo uso del siguiente script:

```
<script type="application/ld+json">"Aquí va los apartados del JSON que queremos añadir"</script>
```

En nuestro caso hemos optado por añadir todo el JSON completo desde el principio y hacerlo estático para todas las páginas.

## 11. Rendimiento

A continuación, se mostrará el rendimiento de nuestra página web haciendo uso de [PageSpeed Insights](#).

- **Rendimiento en móvil**

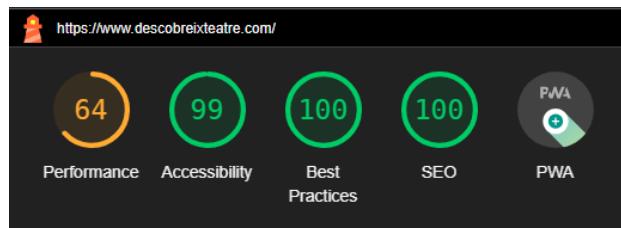


- **Rendimiento en escritorio**

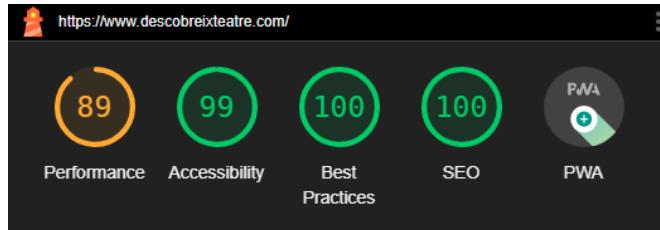


Haciendo uso de Lighthouse que se encuentra en el apartado de Inspeccionar de Google Chrome tenemos:

- **Rendimiento en móvil**



- **Rendimiento escritorio**



## 12. Media

En este apartado se mostrarán los elementos multimedia usados, ya sean propios como disponibles en internet:

- ***Propios***

- ***SVG*** (ubicado en `/assets/img/svg`)
  - ***Marcador para los teatros y ícono de la aplicación web***



- ***Video*** (ubicado en `/assets/vid`)

Video explicativo de la web, el siguiente video será usado en el apartado de quienes somos. Se encuentra subido en [Youtube](#).

- ***Audio*** (ubicado en `/assets/aud`)

En estos audios los alumnos se han encargado de leer las descripción de los diferentes teatros incorporados.

- ***Obtenidos de internet***

- ***SVG*** (ubicado en `/assets/img/svg`)
  - ***Marcador Mapa***

Usado para representar la posición del usado actualmente por medio de geoposición.



- ***Logo Principal de Descobreix Teatre***



- ***Marcador para las ferias de baleares (JSON compañero)***



■ *Marcador para las fiestas (JSON compañero)*



○ *Imágenes*

■ *Teatros* (ubicado en /assets/img/teatros)

● *Mallorca*

○ *Teatro Principal de Palma*



○ *Teatro Principal de Inca*



○ *Teatro Mar i Terra*



○ *Teatro de Manacor*



● *Eivissa*

○ *Teatro Caló de S'Oli*



○ *Teatro Pereira*



- *Menorca*
  - *Teatro Principal de Mahón*



- *Sala Polivalente de Menorca*



## ■ *Obras* (ubicado en /assets/img/obras)

- *Los perros*



- *La mala diccio*



- *Joseph Bernard*



- *Els dies bons*



- *Cami Escola*



- *Bulle A Bulle*



- *Próximamente (en caso de que un teatro no tenga 3 obras para poner en la cartelera)*

**PRÓXIMAMENTE**

- *Actores* (ubicado en /assets/img/actores)

- *Alvar Triay*



- *Carles Pedragosa*



- *Caterina Alorda*



- *Jon Lopez*



• *Jordi Oriol*



• *Joseph Bernard*



• *Lluis Marques*



• *Lulu Cormican*



• *Martín Cattani*



• *Marxel Rodriguez*



• *Neus Cortés*



• *Paula Malia*



• *Pep Bou*



- **Video** (ubicado en `/assets/vid`)

- [Los perros](#)
- [La mala diccio](#)
- [Els dies bons](#)
- [Cami a la escola](#)
- [Bulle A Bulle](#)

## 13. Conclusiones de la práctica

En un principio la práctica nos resultó un reto considerable ya que no habíamos diseñado ninguna página dinámica hasta el momento y menos usado JavaScript, pero al final la práctica ha resultado ser una tarea bastante divertida.

Gracias al material que se ha aportado en la asignatura y disponible en internet se ha podido agilizar mucho el trabajo.

## 14. Posibles mejoras

Voy a listar una lista de posibles mejoras que se le podrían aplicar a la siguiente práctica.

- **Mejorar el rendimiento en móvil:** Aunque nos hemos dedicado a mejorar el rendimiento todo lo posible en nuestra práctica aún se podría optimizar más.
- **Eliminar CSS y JavaScript no usado de librerías usadas:** Usando Node se podría conseguir limpiar funciones de JavaScript y estilos de CSS no usados.
- **Optimizar o cambiar Leaflet Routing Machine:** Es una librería muy útil pero supone un gran coste en el rendimiento. Hemos tenido que limpiar mucho esta librería para conseguir el rendimiento actual pero aún se podría pulir mucho más.
- **Web Semántica Dinámica:** Una mejora segura a realizar es convertir la Web Semántica incorporada en dinámica para la página principal/teatros/obras.
- **Mejor uso de los JSON de compañeros:** Hemos optado por un uso simple de los JSON de compañeros pero se podría dar mucho más juego.
- **Comprimir los JavaScript, CSS y JSON:** Esto permitiría un aumento de rendimiento considerable.

Esta es una lista de mejoras que considero como mejoras futuras importantes que a nosotros nos hubiera gustado realizar si no tuvieramos tantos trabajos al mismo tiempo. Aún existen mejoras que no se han listado pero considero que no son tan críticas o interesantes como las mencionadas.

## **15. Valoración**

A pesar de que se han mencionado muchas mejoras en el apartado anterior, consideramos que nuestra práctica es una muy buena práctica a la cuál se le ha mimado bastante. Por eso, teniendo en cuenta que aún se puede mejorar y debido a la limitaciones de tiempo debido a otras asignaturas y prácticas de empresas de la mayoría de miembros, considero que la práctica es de un **9 o 9,5**. Los puntos a destacar:

- **Web App:** Se ha creado la web app de la página web
- **JSON completo:** Nos hemos tomado nuestro tiempo para desarrollar un JSON lo suficientemente completo como para poder añadir en cualquier momento obras o teatros sin problemas. Además de garantizar la validación.
- **APIS y Librerías:** Nos hemos tomado nuestro tiempo pensando cómo incorporar la mayor cantidad de APIS/Librerías posibles y que proporcionen valor a la página web. En especial Leaflet Routing Machine que hemos tenido que editar.
- **Comentar el código:** Aunque era un criterio obligatorio, nos hemos tomado un tiempo largo para poder redactar comentarios completos y simples de comprender.
- **No explotar APIS/Librerías:** Para poder probar diferentes funciones para nuestra página web hemos evitar usar siempre los mismo elementos para una tarea concreta. Por ejemplo:
  - **SpeechSynthesis:** Podríamos haber usado SpeechSynthesis para que leyera la descripción de los teatros al mismo tiempo que el de las obras. Pero hemos optado en teatro por usar un reproductor de audio donde hemos tenido que grabar todos los audios.
  - **Reproductor de video:** Para evitar usar <video></video> en dos sitios diferente (página principal y obras) hemos optado que el video propio se reproduzca en un “iframe” de Youtube.
- **GitHub público:** Para que futuros puedan apreciar nuestra práctica y puedan usarla como guía se ha hecho público un repositorio de GitHub que contiene la práctica definitiva.