

# Segundo entregable de la UF1 - Lenguaje de marcas

## Descripción general de la prueba

### INTRODUCCIÓN

Para la siguiente práctica vamos a crear una página web que nos dará información sobre películas.

La página tendrá las siguientes secciones:

- Página de login
- Página que nos permitirá buscar películas

Para ello, vamos a ayudarnos en la creación de las estructuras mediante los ['wireframes'](#) que os compartiremos a continuación.

**NOTA:** Para esta práctica tendremos una única página HTML. Iremos mostrando, ocultando y creando los elementos desde Javascript, como hemos visto en clase.

## CRITERIOS DE EVALUACIÓN

Se tendrá en cuenta los siguientes puntos a la hora de evaluar vuestro ejercicio de mayor a menor importancia:

- Cumplir lo requerido por el enunciado de la página
- Respetar el wireframe
- Respetar los estándares de buen código y calidad del mismo.
- La creatividad y utilización de herramientas que hemos aprendido durante el curso aunque no se pidan explícitamente.
- Realizar un diseño del site que, con la ayuda del wireframe, sea bonito a la vista para las personas que visitan el site.

Restará:

- Uso indebido de JS (código duplicado, código innecesario, código deprecated como var, o mal uso de let y const)
- Uso indebido de tags HTML, selectores CSS y reglas CSS. Aunque la práctica no se va a centrar en eso, se penalizará el uso indebido
- Duplicar código de manera innecesaria.

Está totalmente prohibido:

- Copiar código entre compañeros.
- Uso de código copiado de sites de internet (podéis apoyaros en la red para documentarnos e informarnos).
- Uso de CSS inline o de etiquetas `<style>` dentro del código html de la página.
- Uso de JS inline código dentro de las etiquetas `<script />`

## INFORMACIÓN ADICIONAL

Para la realización de la práctica vamos a usar la API de [The Movie Database](#). Y para realizar la búsqueda de películas utilizaremos en [endpoint de search](#):

`https://api.themoviedb.org/3/search/movie?api_key=API_KEY&query=MOVIE`

- Podéis obtener la **API\_KEY** necesaria para realizar la consulta si os [registráis](#) en la página y posteriormente solicitáis el [API\\_KEY](#). Aún así, os dejamos un API\_KEY para que empecéis a trabajar:

40d741b75b657fc99fa896bb456729c1

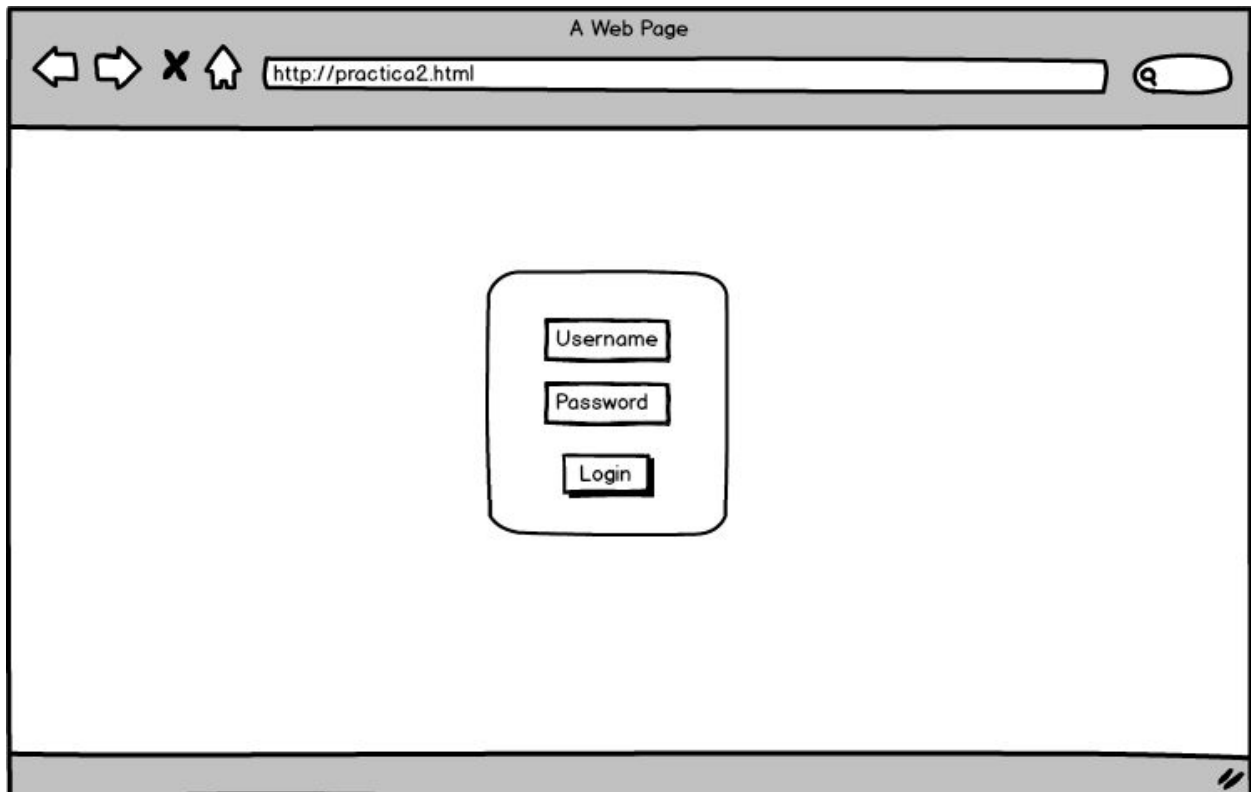
Las API\_KEY tienen ciertas restricciones de uso, por lo que os recomendamos que os registreis y obtengáis la vuestra propia para evitar problemas.

- El campo **MOVIE** lo obtendréis de la información que introduzca el usuario en el campo de texto.

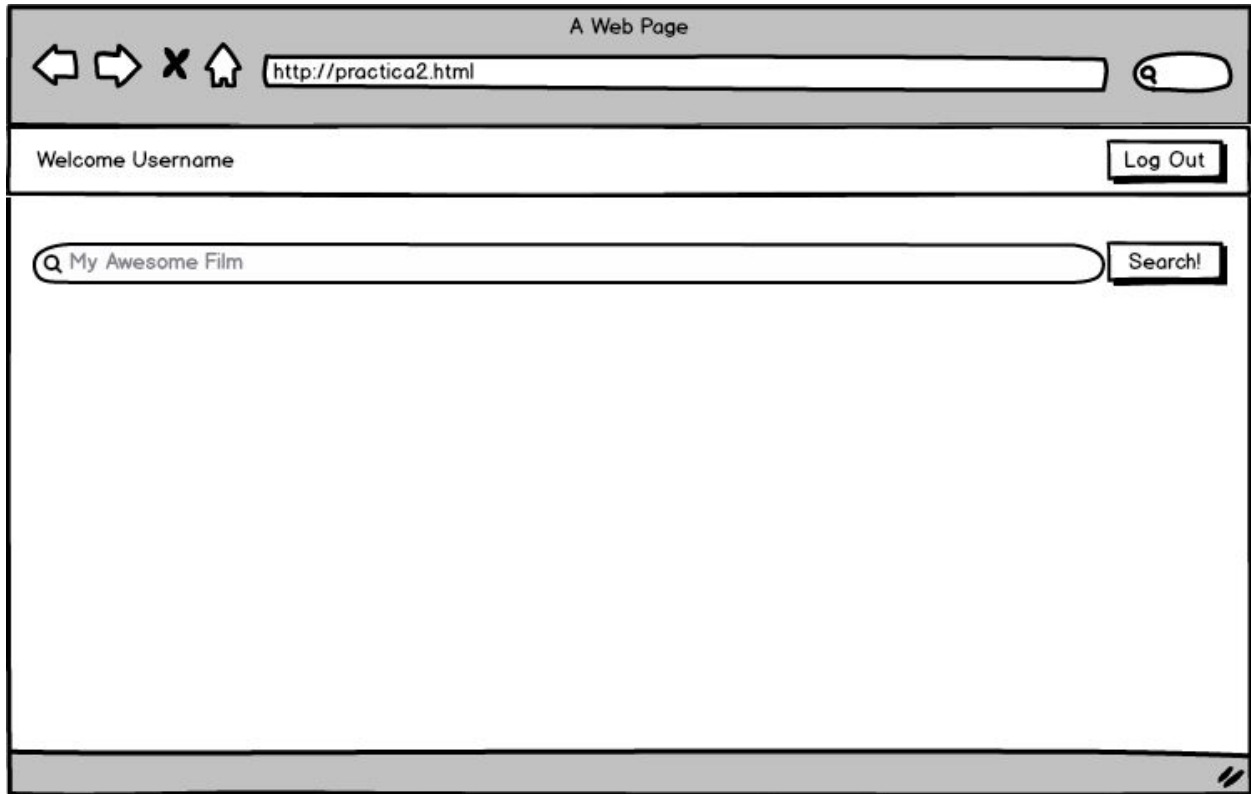
## Ejercicios

### 1 LOGIN Y PÁGINAS PRINCIPALES (3.5 Puntos)

Cuando el usuario entra por primera vez a la página web, deberá ver una página de login como muestra el siguiente wireframe:



Una vez logeado, se le mostrará al usuario el contenido del website (donde el usuario podrá realizar la búsqueda de películas). El usuario podrá desloguearse haciendo click en el botón del header como muestra el siguiente wireframe:



La página de Login dispondrá de:

- 1 Formulario que contendrá:
  - Campo de texto username
  - Campo de texto password
  - Botón

La página principal del site dispondrá de:

- 1 Header que contendrá:
  - Un mensaje de bienvenida con el nombre del usuario
  - Un botón para hacer logout
- Una barra de búsqueda que contendrá:
  - Campo de texto, donde el usuario introducirá la película a buscar
  - Botón para efectuar la búsqueda

Utilizaremos [Local Storage](#) para guardar el nombre del usuario (no haremos nada con el campo password). De manera que, si existe el nombre de usuario, mostraremos la página principal del website, en caso contrario mostraremos la página de login.

Al hacer click en login, guardaremos el usuario en local storage y al hacer click en logout, lo borrarémos de local storage. La página debe comportarse correctamente cuando la actualizamos (F5 / Refresh).

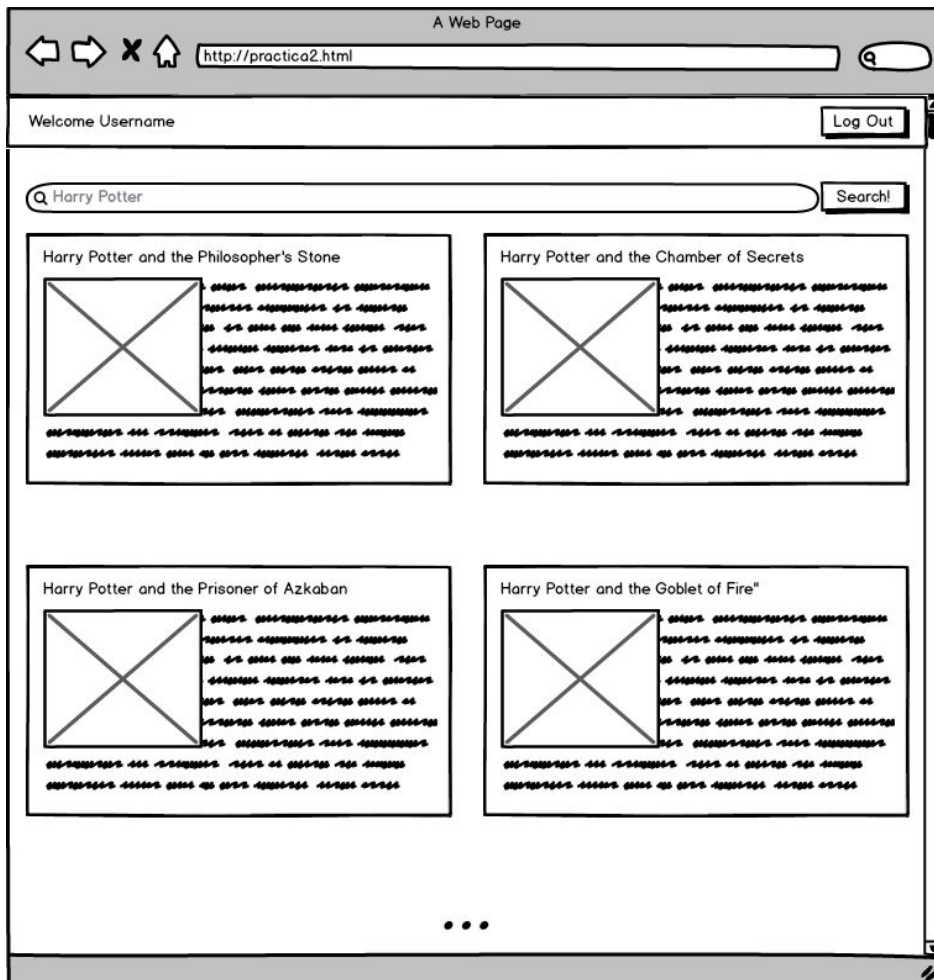
## 2 BÚSQUEDA DE PELÍCULAS (4 Puntos)

Cuando el usuario introduzca el nombre de una película en el campo de texto y haga click en el botón, se le deberá mostrar la lista de películas que coinciden con el texto introducido.

Para ello, utilizaremos este endpoint como hemos explicado anteriormente:

`https://api.themoviedb.org/3/search/movie?api_key=API_KEY&query=MOVIE`

Este endpoint devolverá un JSON con un array de películas que cumplen los criterios de búsqueda, podéis ver un ejemplo [aquí](#). Con estos datos, deberemos mostrar al usuario la lista de películas siguiendo el siguiente wireframe:



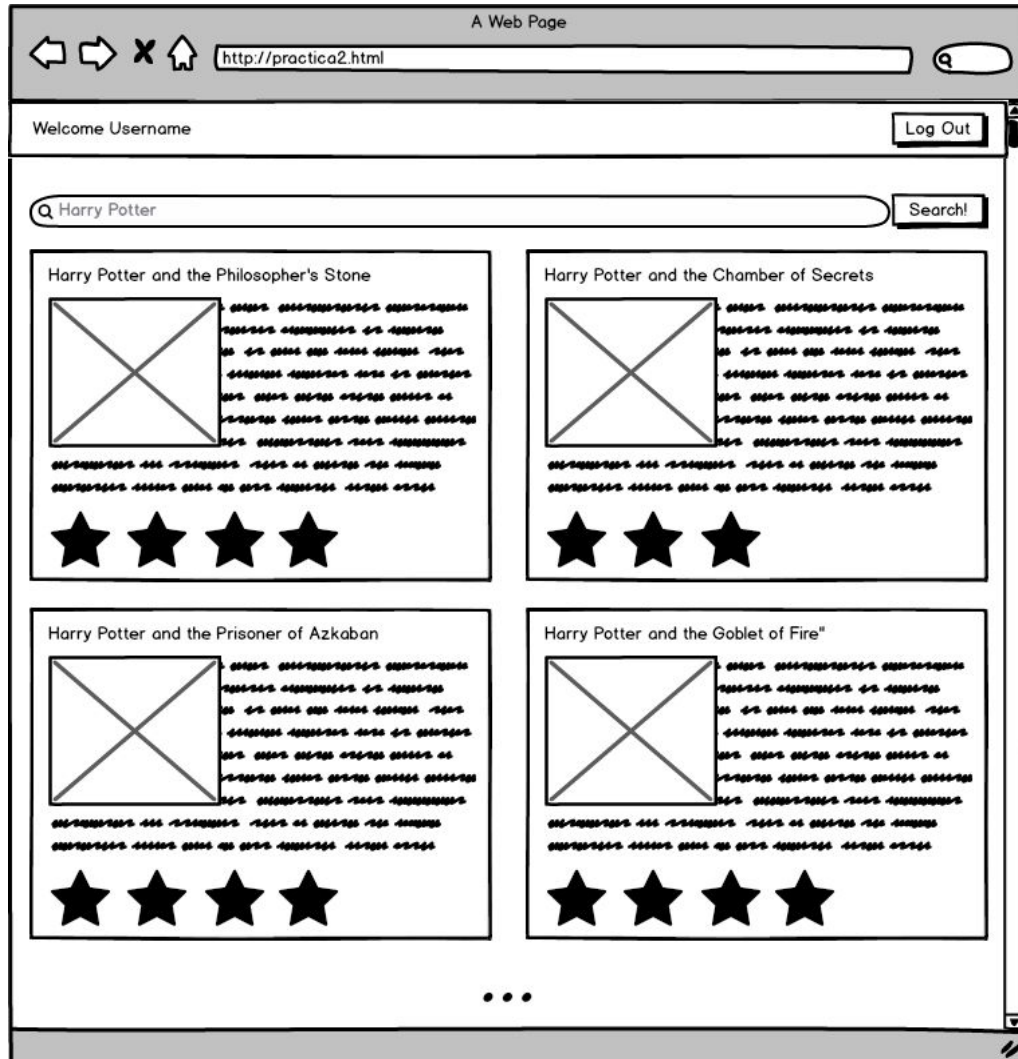
El bloque película deberá constar de:

- Título
- Imagen de la película. Para obtener la imagen de la película deberéis usar el campo `poster_path`. La URL de la imagen será:  
``https://image.tmdb.org/t/p/original/${poster_path}``
- Descripción de la película



### 3 PUNTUACIÓN DE LA PELÍCULA (2.5 Puntos)

Junto con los datos de la película, mostraremos la puntuación de la película en forma de estrellas.



La puntuación de la película la obtendréis a partir del campo `vote_average`. La relación `vote_average` - estrella será:

- 1 Estrella (`vote_average` entre 0 y 2)
- 2 Estrellas (`vote_average` entre 2 y 4)
- 3 Estrellas (`vote_average` entre 4 y 6)
- 4 Estrellas (`vote_average` entre 6 y 8)
- 5 Estrellas (`vote_average` entre 8 y 10)

