

# Módulo 3: Ejercicio de evaluación final

Antes de empezar a realizar el ejercicio, hay que crear un nuevo repositorio desde GitHub Classroom usando [este enlace](#). Una vez creado, hay que clonarlo en nuestro ordenador y en la carpeta creada empezaremos a trabajar en el ejercicio.


El ejercicio consiste en desarrollar una página web con el listado de las escenas de las películas donde el actor Owen Wilson ha dicho 'wow', [API Wow de Owen Wilson](#) es la API que usaremos en este ejercicio. Vamos a usar React para realizarlo.

## Owen Wilson's "wow"

Movie


Year 

All




Cars 3 - 2017

Wow.



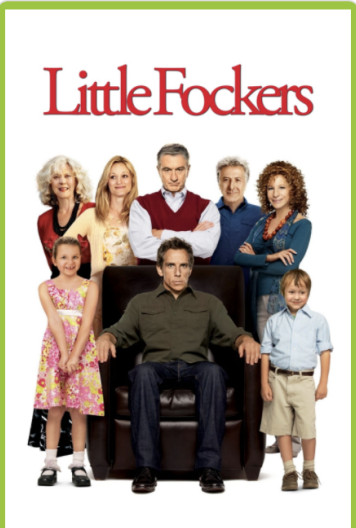
Drillbit Taylor - 2008

Wow.




Are You Here - 2013

Yeah. Wow.




Little Fockers - 2010

Wow.



Cars 3 - 2017

Wow. You're all here.



Starsky & Hutch - 2004

Wow. This is a nice boat.

Vamos de definir las distintas partes del ejercicio:

## 1. Listado de escenas

En primer lugar, vamos a realizar una web con el listado de 50 escenas donde el actor Owen Wilson ha dicho 'wow'.

Para eso, vamos a utilizar el servicio de <https://owen-wilson-wow-api.herokuapp.com/wows/random?results=50> que nos devuelve información de 50 escenas de películas aleatorias. Sobre cada una, vamos a mostrar al menos:

- Poster (**poster**)
- Película (**movie**)
- Frase completa (**full\_line**)
- Año (**year**)

Puedes ingresar a la página oficial del Api en el siguiente enlace <https://owen-wilson-wow-api.herokuapp.com/>

## 2. Filtrado por película

Ahora que ya tenemos el listado de escenas, la segunda parte consiste en realizar un filtro para buscar por película. Para eso, añadimos un **input** a la interfaz, de forma que al ir escribiendo un nombre queden en la interfaz solo las escenas cuya película contiene las letras escritas.

**Nota:** en principio no es necesario tener en cuenta si las letras están en mayúscula / minúscula para la búsqueda, pero si queréis añadir esta mejora pues genial.

## 3. Filtrado por año

Ahora que ya tenemos el listado de escenas en pantalla, y filtrado por nombre de la película donde aparece la escena, la siguiente parte consiste en filtrar el listado por año de la película. Para eso, añadimos un **select** a la interfaz, de forma que al seleccionar un año queden en la interfaz solo las escenas que coincidan con el año seleccionado.

**Nota:**

- Por defecto, cuando carga la página debe aparecer **todos**.
- Obtén los años de las películas del listado.

## 4. Componentes de la aplicación

La aplicación debe tener los siguientes componentes como mínimo:

- Componente para los filtros.
- Componente para el listado (**MovieSceneList**).
- Componente para la tarjeta de cada escena del listado (**MovieSceneItem**).
- Componente para el detalle de cada escena del listado (**MovieSceneDetail**).

## 5. Detalle de cada escena

Vamos a implementar una nueva funcionalidad: al hacer clic sobre la tarjeta de una escena, su información aparecerá a pantalla completa. Para hacer esto usaremos rutas y React Router DOM. En la pantalla de detalle aparecerá:

- nombre de la película,
- frase completa
- director
- el enlace del audio de la escena, al darle clic debe mostrarse en una pestaña aparte en el navegador.

## Owen Wilson's "wow"



**Nota:** no recomendamos mostrar el detalle de cada escena con una ventana modal por encima del listado de escenas de las películas porque es más complejo. Es mejor quitar el componente `MovieSceneList` y mostrar el componente `MovieSceneDetail` usando React Router.

## 6. Detallitos de calidad

- Como nos gusta cuidar la semántica, el campo de texto debe estar recubierto por una etiqueta `<form />`.
- Si estando en el campo de filtrado pulsamos intro debéis impedir que el navegador navegue o cambie la ruta sin querer.
- Si se busca por un texto por ejemplo "ZZZ" y no hay ninguna escena de película que coincida con dicho texto se debe mostrar un mensaje del tipo "No hay ninguna nombre de película que coincida con la palabra ZZZ".
- El filtro debe filtrar independientemente de que la usuaria introduzca el texto en mayúsculas o minúsculas.

- Al entrar en el detalle de un escena y a continuación pulsar atrás, el campo de texto debe mostrar el texto que tenía anteriormente.

## 7. BONUS: Mejoras visuales

Para terminar, podéis realizar algunas mejoras visuales del ejercicio. Por ejemplo:

- Poner bonita tu web y agregar el estilo que desees.
- Agregar iconos a la información que se muestre en el detalle de las escenas.
- Usar algún sistema de grid para pintar el listado.
- Que funcione bien el responsive en dispositivos pequeños.

## 8. BONUS: URL compatible

- Como ejercicio extra os proponemos que la URL del detalle de cada escena sea compatible, es decir, que si visitamos esa URL directamente en el navegador se vea el detalle de la escena de una película. Si refrescamos el navegador en el detalle de una escena de la película debe volver a mostrar el detalle de la película, solo si está almacenado en el local storage.
- Y en el caso de que el usuario navegue a una URL inexistente como por ejemplo `http://localhost:3000/detail/12345` (el id 12345 no existe) debemos mostrar un mensaje del tipo "La escena que buscas no existe".

## 9. BONUS: Ordenación

Un extra interesante sería que ordenáseis el listado de escenas alfabéticamente por el nombre de la película.

## Entrega

El límite de entrega es

08/05/2022 a las 14:00 horas

Solo debéis hacer commits y merges en la rama master de vuestro repositorio hasta la fecha límite. Si después del ejercicio queréis seguir trabajando sobre el ejercicio, lo podéis hacer en otra rama y no debéis mergearla hasta después de la entrevista técnica.

La evaluación solo se considerará terminada cuando:

- Esté publicada en GitHub Pages y esté funcionando, para lo cual tendréis que subir el código, también a la carpeta `docs/` del repositorio.
- El enlace a GitHub Pages esté en la página principal del repositorio, en la parte superior, al lado de la descripción.

## Normas

Este ejercicio está pensado para que lo realices de forma individual en clase, pero podrás consultar tus dudas con la profesora y tus compañeras si lo consideras necesario. Ellas no te darán directamente la solución de tu duda, pero sí pistas para poder solucionarla. Aún facilitando la comunicación entre

compañeras, durante la prueba no debes copiar código de otra persona ni acceder a su portátil. Confiamos en tu responsabilidad.

La evaluación es una buena oportunidad para conocer cómo estás progresando, saber qué temas debes reforzar durante las siguientes semanas y cuáles dominas. Te recomendamos que te sientas cómoda con el ejercicio que entregues y no envíes cosas copiadas que no entiendas.

Si detectamos que has entregado código que no es tuyo, no entiendes y no lo puedes defender, pasarás directamente a la re-evaluación del módulo. Tu objetivo no debería ser pasar la evaluación sino convertirte en programadora, y esto debes tenerlo claro en todo momento.

Una vez entregado el ejercicio realizarás una revisión del mismo con la profesora (25 minutos), que se asemejará a una entrevista técnica: te pedirá que expliques las decisiones tomadas para realizarlo y te propondrá realizar cambios in situ sobre tu solución.

Es una oportunidad para practicar la dinámica de una entrevista técnica donde te van a proponer cambios sobre tu código que no conoces a priori. Si evitas que otras compañeras te den pistas sobre la dinámica de feedback, podrás aprovecharlo como una práctica y pasar los nervios con la profesora en lugar de en tu primera entrevista de trabajo.

Al final tendrás un feedback sobre aspectos a destacar y a mejorar en tu ejercicio, y sabrás qué objetivos de aprendizaje has superado de los listados a continuación.

## Criterios de evaluación

Vamos a listar los criterios de evaluación de este ejercicio. Si no superas al menos el 80% de estos criterios o no has superado algún criterio clave (marcados con \*) te pediremos que realices una re-evaluación con el fin de que termines el curso mejor preparada y enfrentes tu primera experiencia profesional con más seguridad. En caso contrario, estás aprendiendo al ritmo que hemos pautado para poder afrontar los conocimientos del siguiente módulo.

### React básico

- Crea componentes con sintaxis correcta\*.
- Crea una estructura adecuada de componentes\*.
- Usa las props para pasar datos a componentes hijos\*.
- Sabe pintar listados\*.
- Sabe usar métodos funcionales de array (`map`, `filter`, etc.)\*.
- Usa el estado para gestionar información de la interfaz\*.
- El componente principal `App.js` maneja el estado de la aplicación\*.
- Usa eventos en React para atender a interacciones del usuario\*.
- Hooks para las peticiones al servidor.
- Escribe un código sólido, sin errores en la consola\*.
- Usa `defaultProps` en los componentes.
- Usa `propTypes` para evitar errores de tipado.
- Tiene soltura a la hora de realizar cambios en el ejercicio presencial.

### React router

- Crea rutas navegables dentro de una aplicación.

## Issues

- Haber resuelto las issues de la evaluación intermedia\*.

## Otros criterios a tener en cuenta

- Usar inglés para nombres de variables, funciones, clases, comentarios, mensajes de commit, nombres de ficheros.
- El repositorio de GitHub debe tener README.md.