Módulo 2: Ejercicio de evaluación final

- Antes de empezar, hay que crear un nuevo repositorio desde GitHub Classroom usando este enlace. Una vez creado, hay que clonar en nuestro ordenador y en la carpeta creada empezaremos a trabajar en el ejercicio.
- A continuación, **si quieres**, hay que descargar e incluir en el proyecto el starter kit de Adalab.

Enunciado

El ejercicio consiste en desarrollar una aplicación web de búsqueda de series de TV, que nos permite des/marcar las series como favoritas y guardarlas en local storage.

El ejercicio también tiene una parte de maquetación con HTML y Sass, os recomendamos dedicar esfuerzo a la maquetación una vez terminada la parte de JavaScript, ya que los criterios de evaluación están relacionados con esta última.

Vamos de definir los distintos hitos del ejercicio:

1. Estructura básica

En primer lugar hay que realizar una estructura básica sobre este modelo. No hay que preocuparse por las medidas, colores ni tipografía hasta un hito posterior.

La aplicación de búsqueda de series consta de dos partes:

1. Un campo de texto y un botón para buscar series por su título.
2. Un listado de resultados de búsqueda donde aparece el cartel de la serie y el título.

2. Búsqueda

- Al hacer clic sobre el botón de **Buscar**, la aplicación debe conectarse al API abierto de TVMaze para búsqueda de series. Os recomendamos echar un vistazo al JSON que devuelve una petición de

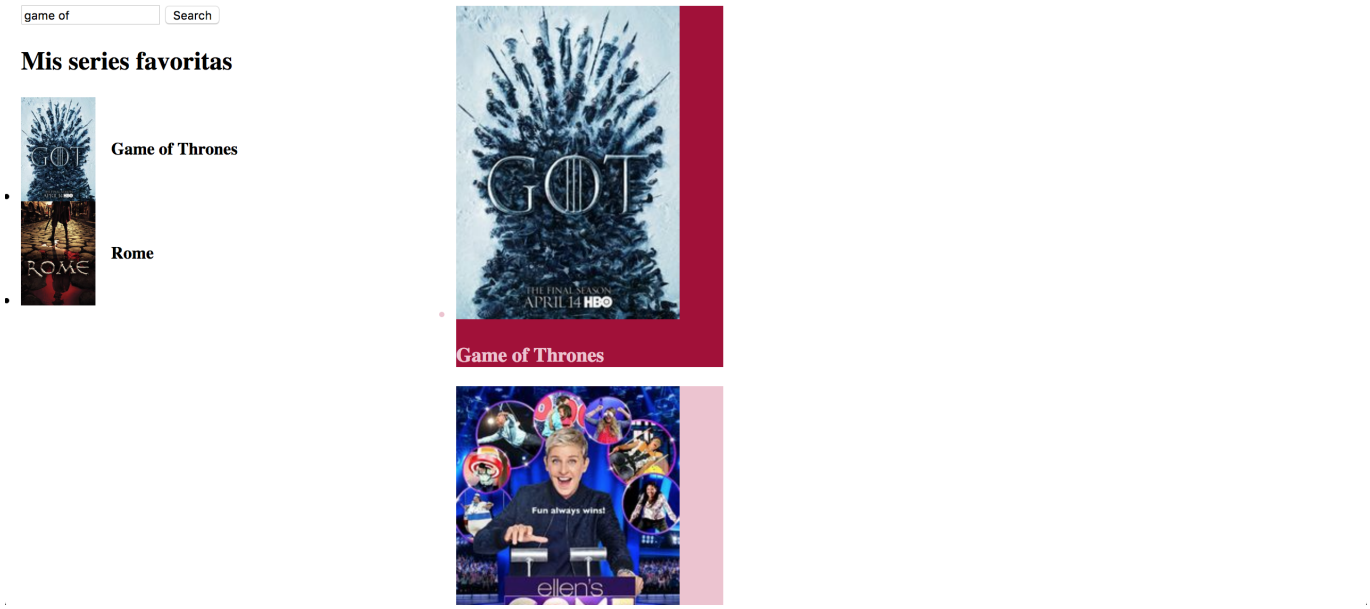
búsqueda para ver qué datos son los que necesitamos.

- Para construir la URL de búsqueda hay que recoger el texto que ha introducido la usuaria en el campo de búsqueda.
- Por cada show contenido en el resultado de la búsqueda hay que pintar una tarjeta donde mostramos una imagen de la serie y el título.
- Algunas de las series que devuelve el API no tienen imagen. En ese caso hay que mostrar una imagen de relleno. Podemos crear una imagen de relleno con el servicio de placeholder.com donde en la propia URL indicamos el tamaño, colores, texto: https://via.placeholder.com/210x295/ffffff/666666/?text=TV.
    - Si buscamos la serie **dexter** veremos que algunas de las series devueltas por el API no tienen imagen.
    - Si buscamos la serie **tronos** veremos que todas las series devueltas por el API sí tienen imagen.
- Para pintar la información en la página se puede elegir entre hacerlo de forma básica con `innerHTML` o manipulando de forma avanzada el DOM.

3. Favoritos

Una vez aparecen los resultados de búsqueda, la usuaria puede indicar cuáles son nuestras series favoritas. Para ello, al hacer clic sobre una serie debe pasar lo siguiente:

- El color de fondo y el de fuente se intercambian, indicando que es una serie favorita.
- Hay que mostrar un listado en la parte izquierda de la pantalla, debajo del formulario de búsqueda, con las series favoritas. Os recomendamos crear un variable o constante de tipo array en JS para almacenar las series favoritas.
- Las series favoritas deben seguir apareciendo a la izquierda aunque la usuaria realice otra búsqueda.



4. Almacenamiento local

Hay que almacenar el listado de favoritos en el localStorage. De esta forma, al recargar la página el listado de favoritos se debe mostrarse.

5. BONUS: Borrar favoritos

Como bonus, os proponemos la opción de borrar favoritos. Al hacer clic sobre el icono de una 'x' al lado de cada favorito, hay que borrar el favorito clicado de la lista y del localStorage.

Para terminar de rematar nuestra app de series, nos gustaría poder añadir/quitar como favorito al hacer clic sobre una serie del lado de la derecha. Y que, si realizamos una nueva búsqueda y sale una serie que ya es favorita, aparezca ya resaltada en los resultados de búsqueda (con colores de fondo y texto intercambiados).

Y ya sería fantástico si al final de la lista de favoritos hay un botón para borrarlos todos los favoritos a la vez.

6. BONUS: Afinar la maquetación

Una vez terminada la parte de interacción, podemos centrarnos en la parte de maquetación donde tenéis libertad para decidir los estilo. En cualquier caso os dejamos una propuesta gráfica.

Entrega

Hemos pautado 12 horas de dedicación al ejercicio, por lo que el límite de entrega es miércoles 10 de febrero a las 14:00.

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 combinarla (merge) hasta que los profesores os lo indiquen.

La evaluación solo se considerará terminada cuando:

- Esté publicada en GitHub Pages y esté funcionando.
- El enlace a GitHub Pages esté en la página 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 asemenjará 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.

General

- Usar una estructura adecuada de ficheros y carpetas para un proyecto web, y enlazar bien los distintos ficheros*.
- Tener el código perfectamente indentado*.
- Uso de control de versiones **con ramas** para manejar un proyecto de código.

JavaScript básico

- Crear código JavaScript con sintaxis correcta y bien estructurado*.
- Usar constantes / variables para almacenar información y re-asignar valores*.
- Usar condicionales para ejecutar acciones distintas en función de una condición*.

- Saber trabajar con listados de datos (arrays)*.
- Usar funciones para estructurar el código.
- Saber modificar la información del DOM para añadir contenido dinámico*.
- Saber escuchar eventos del DOM y actuar en consecuencia*.

Peticiones AJAX y APIs

- Crear peticiones con fetch y promesas*.
- Saber trabajar correctamente con la respuesta del servidor*.
- Gestionar información en formato JSON.
- Usar el localStorage para guardar información en el navegador.

Issues

- Haber resuelto las issues de la evaluación intermedia.

Otros criterios a tener en cuenta

- Usar inglés para nombres de variables, funciones, clases, mensajes de commit, nombres de ficheros.
- El repositorio de GitHub debe tener README explicando muy brevemente cómo arrancar el proyecto.

¡Al turrón! (o a la torrija)