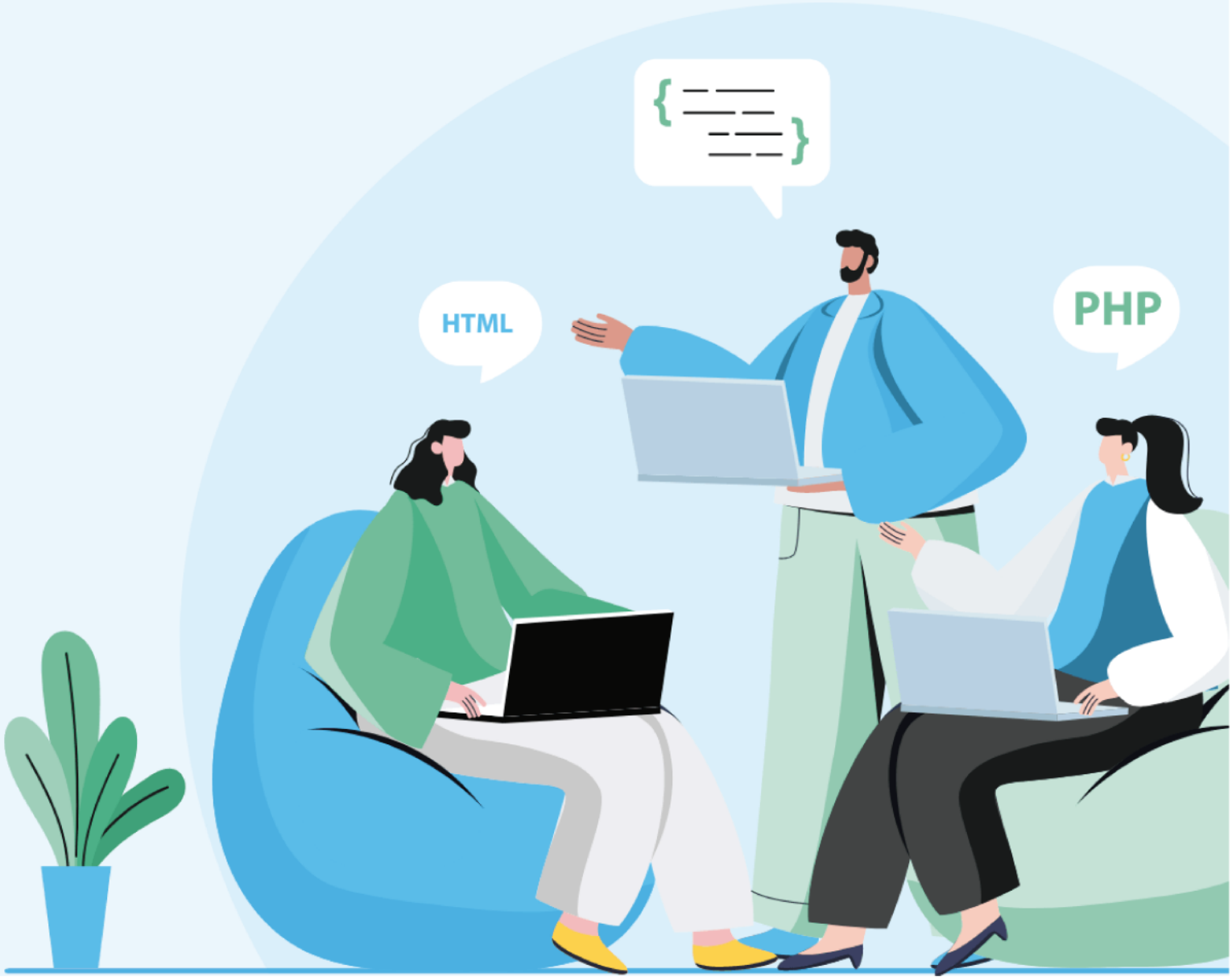
alkemy

Skill Up Lab - React Unidad 6



Unidad 6: Armando el Buscador y la sección Favoritos

Antes de adentrarse en esta unidad

Para que puedas aprovechar mucho más el contenido presente en esta unidad, te recomendamos que antes de avanzar tengas muy en claro lo siguiente:

- cómo capturar la información que viaja en el query string de la URL
- cuál es el sentido de que exista información que viaje en el query string
- cómo generar mensajes de error en el escenario que la información del query string no permite hacer llamados válidos a la API
- como es el proceso de renderización de datos dinámicos sin la necesidad de usar un "*mapeo*" de información

Objetivos de la unidad

Con el estudio de esta unidad buscamos que:

- Armes un sistema de búsqueda que permita encontrar películas por palabra clave
- Valides la información suministrada de parte del usuario en este buscador para evitar hacer llamados innecesarios a la API
- Muestres los resultados obtenidos de la búsqueda en un nuevo componente preparado para esta finalidad
- Implementes la lógica de "Agregar a Favoritos" para que el usuario de la aplicación pueda guardar sus películas favoritas

Clase 1: Armado la barra de búsqueda

¿En qué lugar de la aplicación deberá estar presente la barra de búsqueda?

Generalmente la barra de búsqueda deberá ser implementada en un lugar de fácil acceso para la persona que visita nuestra aplicación, por eso el lugar más recomendado será en la barra de navegación.

¿Cómo debo armar una barra de búsqueda?

En nuestro caso, la haremos realmente sencilla para no complicarnos tanto, pues solamente tendrá un campo de inserción de texto y un botón de acción. Pero ten en cuenta que en proyectos algo más grandes, esta barra de búsqueda puede ser tan compleja como sea necesario.

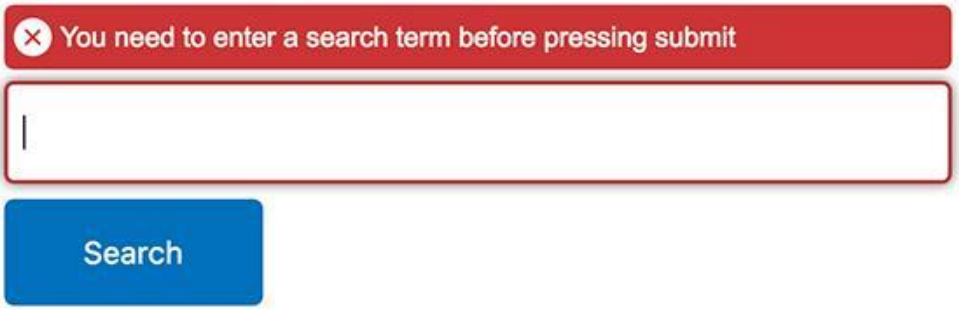
¿Esta barra de navegación se trabaja como si fuera un formulario de HTML?

Exactamente. Así que nuestra misión será implementar una estructura HTML / JSX que tenga lo tradicional de un formulario:

- la etiqueta <form>
- la etiqueta <input /> y
- la etiqueta <button>

Obviamente, con sus atributos respectivos.

Clase 2: Validación del formulario de búsqueda



¿Por qué se hace necesario validar este formulario?

Puntualmente nos interesará validar la información suministrada por el visitante, pues de esta manera nos podremos asegurar que antes de enviar la petición a la API, sepamos que tenemos la información necesaria. De este modo, evitaremos cargar nuestra aplicación de llamados innecesarios logrando de esta manera un mejor performance de la misma.

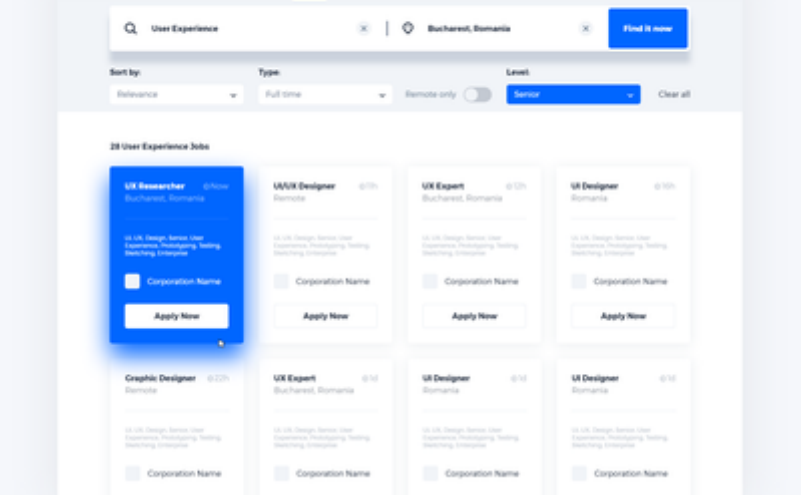
Y ¿en qué consiste este proceso de validación?

Puntualmente nos encargaremos de validar que la persona haya ingresado una cantidad mínima de caracteres en el campo, para poder así hacer el llamado a la API. Y en caso contrario de no contar con dicha cantidad de caracteres, le informaremos a la persona de dicho error.

Pero ¿cómo le podemos mostrar a la persona el error cometido?

De una manera bastante sencilla y ya algo conocida de tu parte. Para este cometido implementaremos las cajas modales de Sweet Alert.

Clase 3: Redirección a la página de resultados



Ahora que ya validamos el formulario de búsqueda ¿Qué sigue?

Una vez que tengamos la información necesaria, podremos redireccionar a la persona a una página de resultados, para que sea dentro de esta, donde podamos mostrar los elementos obtenidos de dicha búsqueda

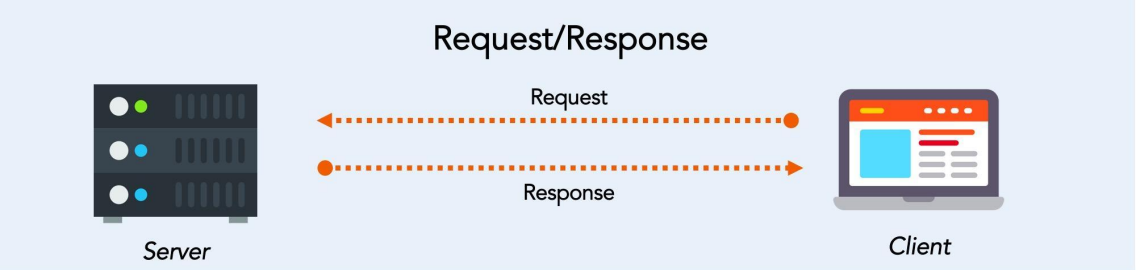
Y esta sección de resultados ¿Es también un listado de elementos a mostrar?

Exactamente, pues al realizar una búsqueda, no sabremos si la información obtenida nos traiga uno o muchos ítems, por lo tanto la misma deberá ser tratada como un listado. Algo con lo que quizás ya tengas algo de familiaridad.

Entendiendo lo anterior ¿se podría reutilizar el componente Listado hecho en clases anteriores?

Totalmente, en la medida de lo posible ese es el objetivo ideal. Pero si te resulta algo complejo o enredado no hay problema con hacer otro componente nuevo, por más que sea muy similar al que ya tenías anteriormente.

Clase 4: Llamado a la API con la palabra clave de búsqueda



Puntualmente, ¿de dónde sacamos la información para la sección de resultados?

Aquí, será la API la que nos provea de un endpoint para poder hacer dicho llamado. Y dependiendo de la palabra clave que enviemos en ese llamado obtendremos los resultados esperados.

Pero ¿cómo haremos para que el formulario de búsqueda pueda enviar al componente de resultados la "palabra clave"?

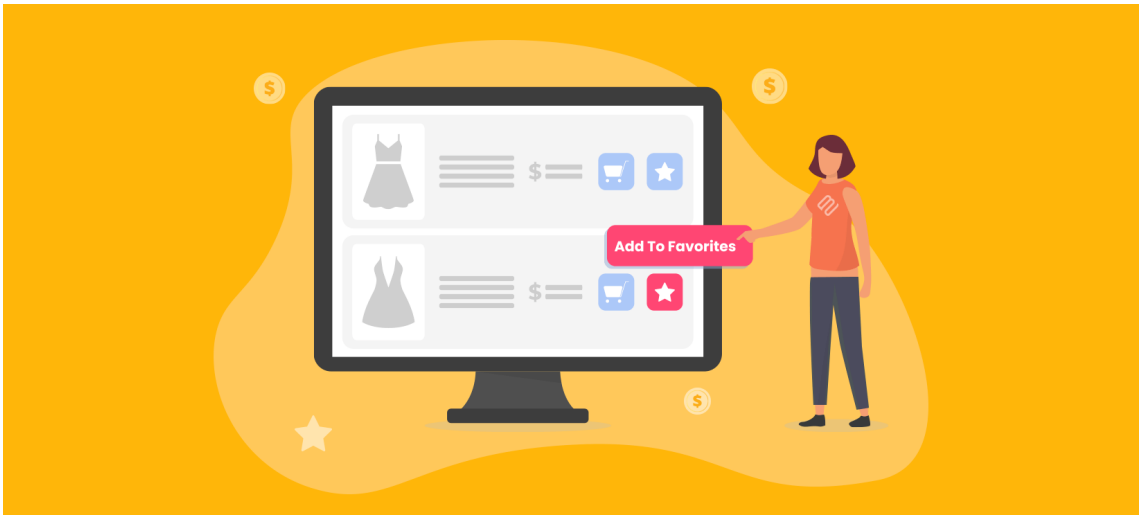
Para este objetivo utilizaremos el query string de la URL. Allí, enviaremos la palabra clave escrita por la persona y la dejaremos disponible para que el componente de resultados pueda tomar la misma y usarla para la petición de información a la API.

¿Y esto sí es algo que suele usarse en la vida real?

Totalmente. De por sí, es uno de los procesos más habituales al momento de desarrollar aplicaciones web, pues pasar información a través de la URL nos permite que, dependiendo de los mismos, podamos mostrar o no una determinada información.

Micro-tip: te invitamos a que revises más a menudo la barra de dirección de tu navegador, para que puedas ver toda la información que los sitios suelen enviar a través de ella.

Clase 5: Generando la lógica para "Agregar a Favoritos"



¿Por qué vamos a implementar esto?

En las aplicaciones web es muy común darle la posibilidad al usuario de que agregue a un listado sus opciones preferidas. Esta opción genera una buena experiencia de uso y le da la sensación a la persona de que nuestra aplicación escucha sus necesidades.

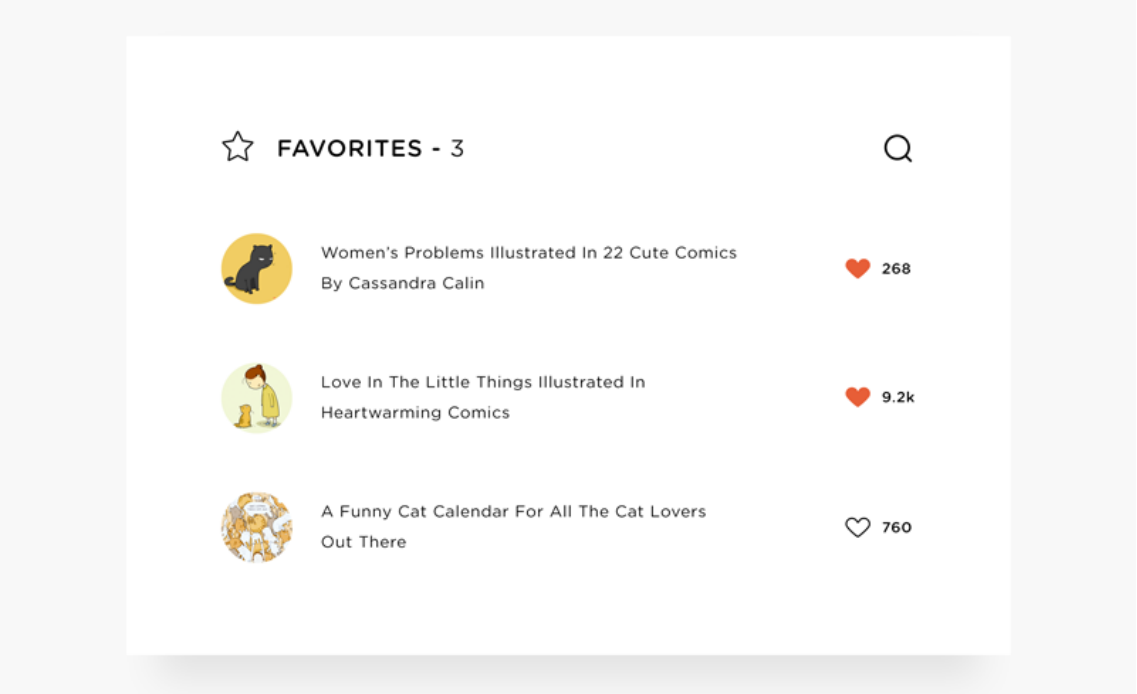
Y... ¿cómo funciona esto de "Agregar a Favoritos"?

En teoría es sencillo, pues lo que tendremos que hacer es implementar antes que nada algún elemento que le permita a la persona llevar a cabo esta acción. Por otro lado, de nuestra parte lo que tendremos que hacer será agregar ese elemento clicado por el usuario a algún almacenamiento en nuestra aplicación (puede ser una base de datos o al almacenamiento local del navegador).

Pero puntualmente, ¿Qué es lo que nos interesa almacenar?

Puntualmente en esta aplicación estaremos almacenando toda la información del ítem elegido de parte del visitante. De esta manera, nos va a ser mucho más sencillo rescatar dicha información más adelante, cuando queramos mostrarla en su totalidad.

Clase 6: Armando el componente "Favoritos"



¿Cómo debemos "pensar" a este componente?

Este componente se parecerá un poco al listado de ítems que hemos venido desarrollando, pero con la particularidad de que aquí, la información no vendrá de la API sino de la información local que previamente hemos guardado.

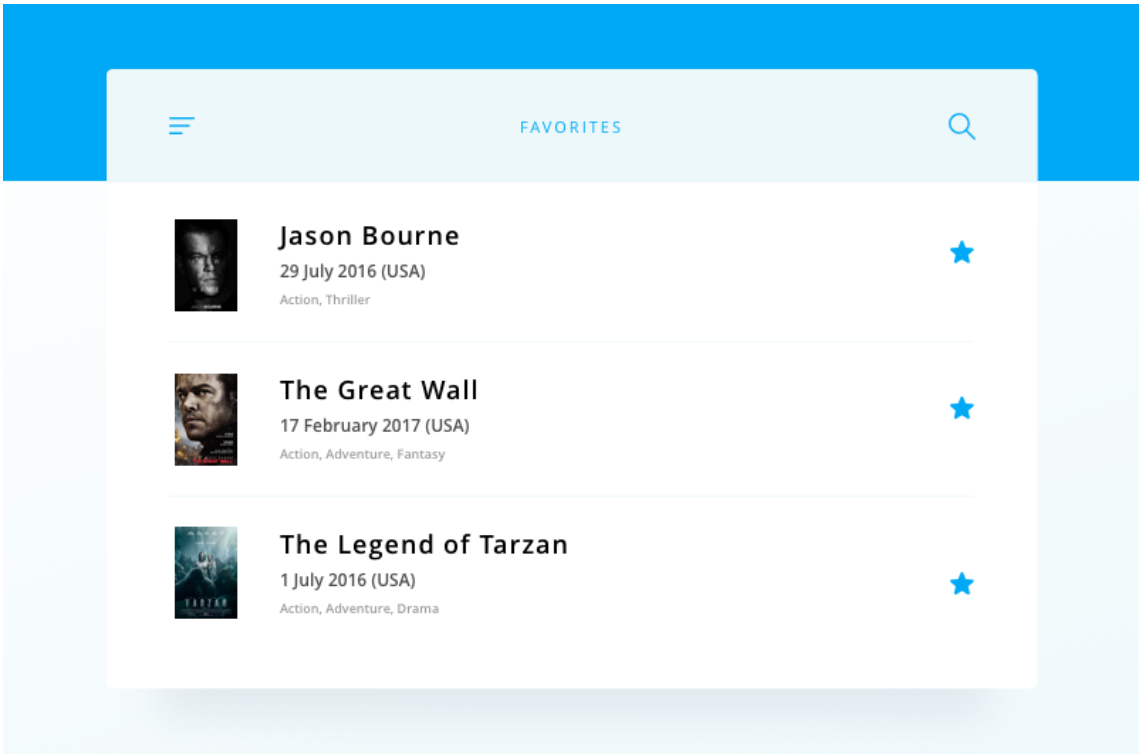
¿Y los elementos agregados a "Favoritos" se pueden eliminar de este listado?

Completamente. Justamente ese será unos de nuestros objetivos, permitir que así como se pueden agregar elementos también se pueda eliminar de dicha lista.

¿Y cómo podremos visualizar aquellos elementos que han sido agregados a "Favoritos" en el resto de nuestra aplicación?

Para ello tendremos que implementar una lógica que nos permita preguntar por cada uno de los elementos del listado general presentes en nuestra aplicación, si se encuentra en el listado de elementos favoritos almacenados. Y en la situación de que exista esa coincidencia, implementar un elemento visual (💖) que ilustre dicha coincidencia.

Clase 7: Finalizando la sección "Favoritos"



Ya terminamos con todo, entonces: ¿Qué haremos en esta clase?

Terminaremos de pulir aquellos detalles que nos hayan hecho falta para dejar nuestra aplicación 100% lista y prolija para que pueda ser presentada.


alkemy