

Trabajando con componentes

Práctica integradora

Objetivo


Luego de dar nuestros "primeros pasos con React", ha llegado el momento de continuar con el armado del *dashboard* de la clase anterior para seguir emprolijándolo a medida que incorporamos más conceptos sobre cómo se implementa una aplicación con React.

A continuación, encontraremos los requerimientos técnicos para seguir adelante con esta gran aventura.

¡Buena suerte!   



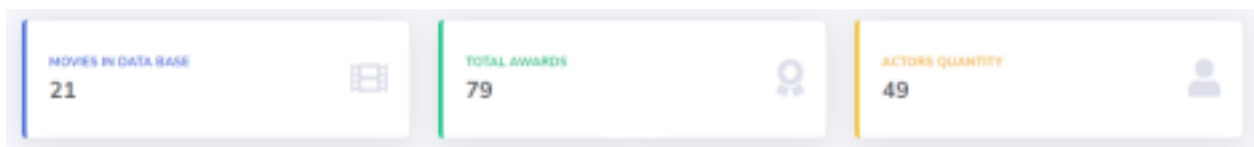
Micro desafío - Paso 1:

Retomemos nuestro proyecto de la práctica anterior. En caso de que no lo tengamos, podemos usar de base el siguiente [proyecto](#) (recordemos instalar todas las dependencias del proyecto, ejecutando el comando **npm install** ).

En el bonus track de la clase anterior teníamos que subdividir el componente creado, **ContentRowTop.js**, en tres **componentes** independientes: **ContentRowMovies.js**, **LastMovieInDb.js** y **GenresInDb.js**, pero si aún no lo logramos, no importa. En esta práctica vamos a completarlo.

Nuestro primer objetivo será generar un componente para las cajas que muestran las métricas:

1. **/components/ContentRowMovies.js**



Dicho componente deberá recibir vía **props** toda la información que estas muestran (cuatro props en total: título, cifra, color de borde, ícono). Es decir, que podamos hacer más dinámica la renderización de la información.

Los tres componentes deberán renderizarse por medio de un **map**, el cual iterará sobre un array de objetos con la información de cada métrica.

También será necesario que dentro del componente implementemos **PropTypes** para que cada **prop** que se recibe, tenga:

- **título**: de tipo string y obligatorio.
- **color de borde**: de tipo string y obligatorio.
- **cifra**: de tipo string o de tipo number y obligatorio.
- **ícono**: una de las siguientes tres opciones: border-left-primary, border-left-success o border-left-warning; y obligatorio.

2

Adicionalmente, para este mismo componente deberemos configurar **defaultProps** para que aquellas **props** de carácter obligatorio tengan un valor definido en cada caso que el mismo no se llegase a enviar.

Posteriormente, seguimos subdividiendo el componente **ContentRowTop.js** en:

2. **/components/LastMovieInDb.js**

3. **/components/GenresInDb.js**



Micro desafío - Paso 2:

Nuestro próximo desafío es sumar a la aplicación un componente adicional, el cual estará antes del footer. Algo importante es que deberemos generar dos componentes. Uno para la tabla como tal y otro para la fila y sus respectivas celdas. Siempre teniendo en cuenta que, por lo menos este último, deberá ser dinámico para recibir la información correspondiente y renderizarla.

Título	Duración	Rating	Género	Premios
Billy Elliot	123	5	<ul style="list-style-type: none">DramaComedia	2
Alicia en el país de las maravillas	142	4.8	<ul style="list-style-type: none">DramaAcciónComedia	3
Título	Duración	Rating	Género	Premios

3

Conclusión

Como vimos, React nos ofrece mecanismos para agregarle prolijidad y validaciones extras a nuestro código que no debemos desaprovechar. Es importante aplicar todo el potencial que ya trae la librería incorporada para mejorar nuestras aplicaciones.

¡Hasta la próxima!