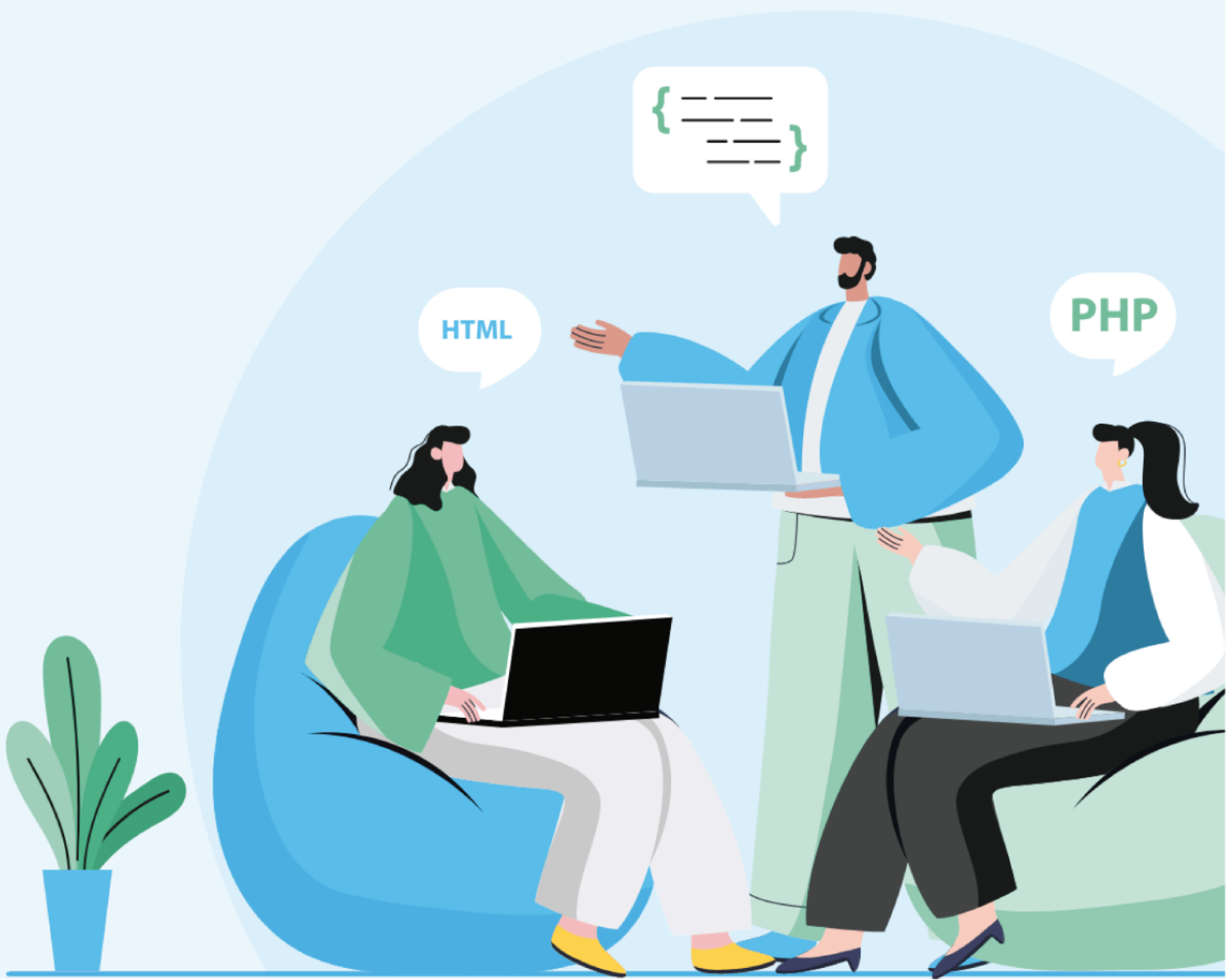




Skill Up Lab - React

Unidad 5





Unidad 5: Armado de la vista "Detalle"

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 se puede proteger el acceso a una determinada ruta de la aplicación
- en donde se almacena la información obtenida de una API dentro del componente
- cómo es el proceso de *mapeo* de información dentro de la estructura JSX de un componente
- cómo generar enlaces dinámicos que permitan navegar hacia otros componentes

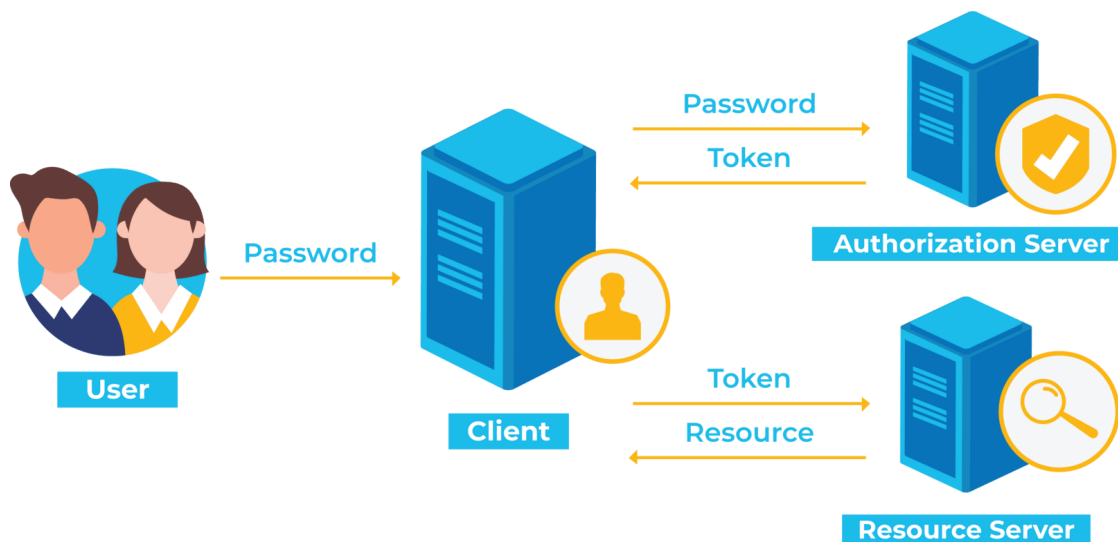
Objetivos de la unidad

Con el estudio de esta unidad buscamos que:

- Armes el componente "Detalle", para que puedas mostrar allí los detalles específicos de un ítem presente en el Listado
- Entiendas cómo generar que un mismo componente te sirva para mostrar la información de cualquier elemento elegido del listado
- Captures la información que identifica a ese ítem de la lista para que puedas hacer el respectivo llamado a la API
- Logres renderizar la información obtenida de la API en la estructura JSX del componente



Clase 1: Verificación de token de usuario



¿Por qué es necesario seguir teniendo en cuenta el Token?

Como hemos hablado anteriormente, no deseamos que algunas rutas sean accesibles si la persona no está correctamente logueada. Por ello, en el desarrollo de cualquier componente que estimemos que debe estar protegido, deberemos hacer siempre esta validación.

Si tengo muchas rutas "protegidas" ¿Deberé hacer siempre el mismo proceso para protegerlas?

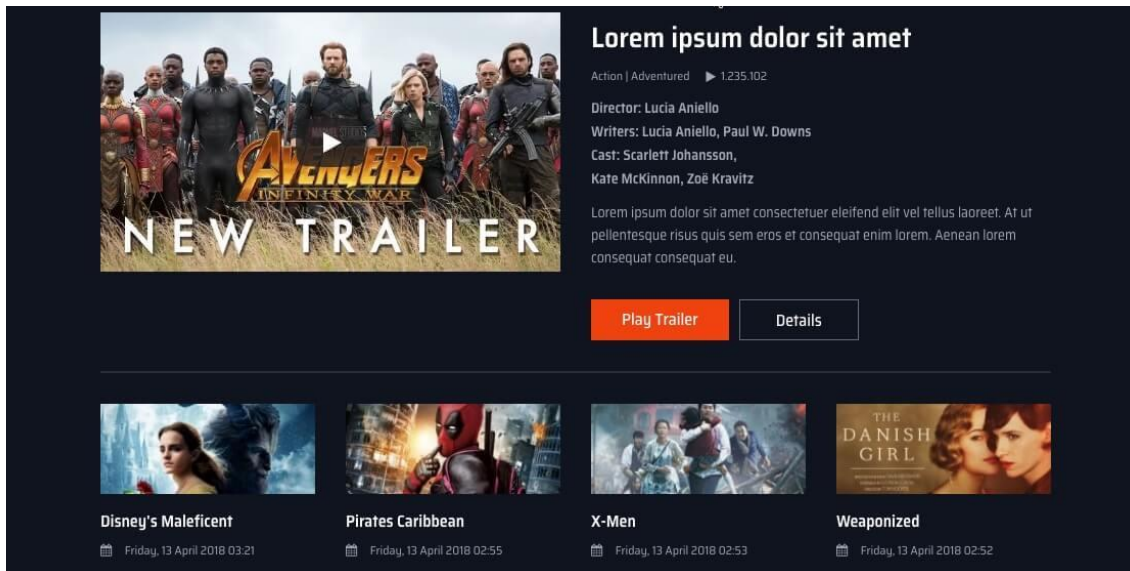
Generalmente sí, pero si pones un poco más en desafío tus conocimientos, puedes incluso crear un componente reutilizable que constate si el token existe dentro de la aplicación. De esa manera no vas a tener que hacer muchas veces el mismo proceso.

¿Existe otra manera de proteger rutas que no sea a través de un token?

Realmente no, pues la más sencilla y habitual es manejar al token como un elemento prioritario y único identificador que te permita saber si la persona que usa la aplicación se ha identificado correctamente con sus credenciales.



Clase 2: Armado de la vista de "Detalle"



¿Pará qué usamos este componente?

Dentro de este componente podremos mostrar la información específica de un ítem en particular. El desarrollo de este componente nos va a permitir crearlo y que el mismo sirva para mostrar cualquier ítem, sea que tengamos uno o sea que tengamos cien.

¿Dentro de este componente se hace necesario usar algún estado?

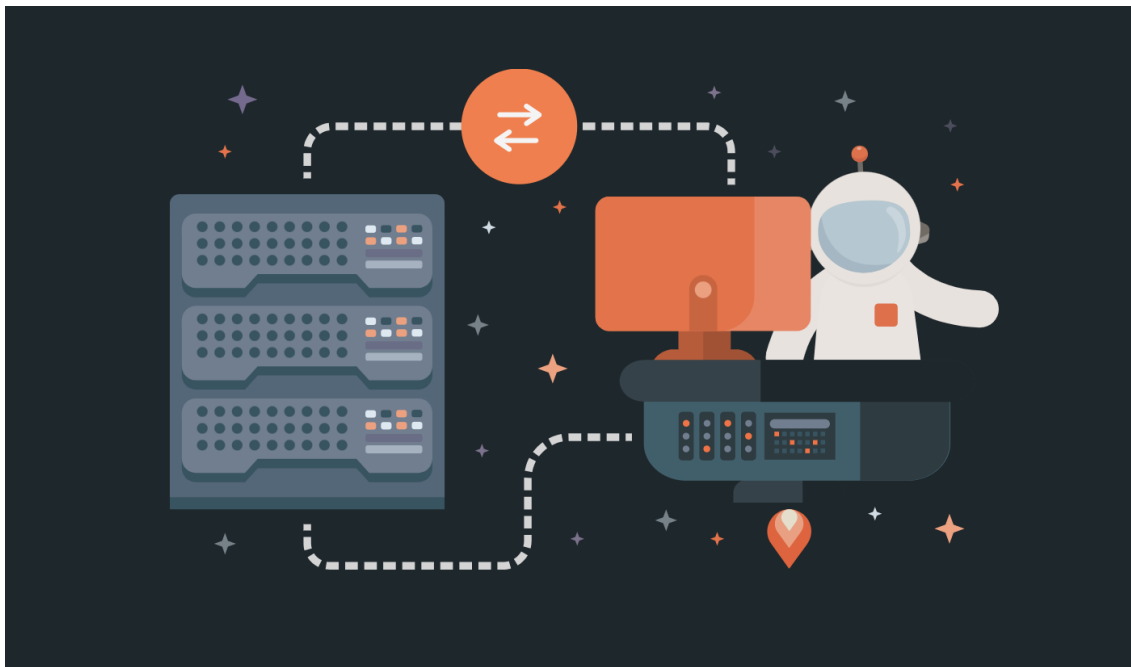
Generalmente sí, pues es en el estado en donde vamos a poder almacenar la información para posteriormente renderizarla en el JSX.

¿El componente Detalle depende del componente Listado para funcionar?

Si bien no están estrictamente vinculados para que uno funcione dependiendo del otro, sí será el componente Listado quien envíe un identificador que nos permite saber a qué elemento en específico deberemos renderizar.



Clase 3: Obteniendo los datos de la API



¿De dónde sacaremos el identificador que nos permitirá hacer el llamado a la API?

Este identificador llegará a este componente en la URL, más exactamente dentro del query string y desde allí lo deberemos tomar para poder hacer el correspondiente llamado a la API.

¿Se suele usar este tipo de implementación en proyectos de la vida real?

Totalmente. Te puedes fijar por ejemplo en cualquier sitio de venta de productos y verás que siempre en el URL hay un elemento identificador que nos permite diferenciar este elemento de otro presente en nuestro listado de elementos.

Pero, ¿cómo se obtiene esa información desde la URL?

Particularmente JS tiene un objeto que nos permite acceder a esta información para que, de esa manera, podamos tomar la porción de la URL que realmente necesitamos para hacer el llamado a la API.



Clase 4: Renderizando la información



Y ahora que tenemos la información necesaria, ¿qué haremos con ella?

A esta altura quizás ya te hayas percatado de que el trabajo dentro de los componentes de React suele ser algo similar, pues generalmente pedimos la información que necesitamos y posteriormente disponemos de la misma en pantalla para que sea consumida por las personas que usan nuestra aplicación. Así que ese será nuestro objetivo puntual en esta clase: mostrar el detalle de la información.

Entiendo, pero ¿aquí también necesitamos usar el método `.map()`?

Posiblemente no, pero todo dependerá de la información recibida. Lo más importante que debes tener en cuenta para el uso del método `.map()`, es que cuando recibes un listado de información (array de objetos por ejemplo), allí es cuando el map cobra real significado y te brindará potencial ayuda.



¿Este es el único componente que necesitaremos para mostrar el detalle de cualquier ítem de la lista?

Sí, y de eso se trata: de que un solo componente funcione para mostrar la información de cualquier elemento del listado, por ello aquí deberemos prestar especial atención a qué información recibiremos y sobre todo saber qué hacer si por ejemplo, algunos elementos del listado no cuentan con esa información. Pero no te preocupes, pues hacer que todo funcione será realmente sencillo.

