



## Enunciado 4

*“Showtime! Señoras y señores. Ladies and gentlemen. Muy buenas noches, damas y caballeros, tengan todos ustedes. Good-evening, ladies & gentlemen. Tropicana, el cabaret más fabuloso del mundo...”* Tres tristes tigres – Guillermo Cabrera Infante, 1967.

Solo porque no hemos querido competirle.

Con este caluroso saludo, queremos darles la bienvenida a casi el último reto y casi la última semana. Esperamos que estén disfrutando de recorrer este camino tanto como nosotros, para esta semana el equipo de iMaster ha decidido experimentar con el **router** de React, ha realizado aproximadamente 20 pruebas las cuales por facilidad se han separado en 2 entregas, una parte descrita en este documento y otra parte para el documento final.

El proyecto, aunque es sencillo, aplica todo lo visto hasta el momento, más dos elementos que aparecen como nuevos en nuestros proyectos, uno es el **router** de React y el otro es el **useContext**, también tenemos un **reducer** con su **useReducer**, **localStorage**, **useEffect**, en resumen, se aplica todo lo visto.

Y para hablar un poco de la aplicación necesitamos darles un contexto y es el siguiente:

- La App tiene rutas públicas y privadas que están protegidas por el **login**, un usuario que no este autenticado no puede visitar una ruta privada y un usuario autenticado no puede visitar la pantalla de **login**.
- El **router** general se llama **AppRouter** y cuenta con un **router** llamado **DashboardRouter** protegido por un **PrivateRoute** y un **Login** controlado por el **router PublicRoute**.
- El **DashboardRouter** del App cuenta con 3 componentes que hacen las



veces de las páginas a mostrar, una de ellas ubicada en **/marvel** permite visualizar los héroes de Marvel Comics que se listan de un archivo **heroes.js**, lo mismo ocurre para **/dc**, solo que permite visualizar los héroes de DC Comics. Finalmente se cuenta con una ruta llamada **/search**, que permite al usuario realizar búsquedas de los héroes tanto de Marvel como de DC.

- El componente **SearchScreen** que se llama en la ruta **/search**, utiliza una **query** en la URL para recuperar una búsqueda realizada, algo similar a **/search?q=batman**.
- Cada héroe tiene una visualización personalizada en la que se pueden ver en mayor tamaño los detalles de este, para acceder a dicha vista, se emplean los parámetros obligatorios del **router** en el llamado a la ruta **/hero**, enviando un id, **/hero/:heroeld**.
- Se cuenta con 3 selectores de filtro, uno aplicado para Marvel y DC separando los héroes mediante una función **filter**. Para los parámetros obligatorios del **router** y mostrar el detalle de un único héroe se utiliza un **find** con el id entregado y, para la búsqueda se utiliza el método **includes** evaluando la cadena de búsqueda ingresada por el usuario.
- La aplicación cuenta con una barra de navegación superior que permite recorrer las diferentes rutas establecidas, mostrar el nombre del usuario autenticado y un botón de **logout** para salir de la misma.
- Se utiliza un **authReducer** para manejar el estado general de la aplicación, dicho estado cuenta con un usuario con los atributos **name** como nombre a mostrar y **logged** como bandera para identificar si el usuario está autenticado o no.
- El estado se almacena en el **localStorage** y es consultado y/o modificado desde el **LoginScreen** y desde la **Navbar**, los cuales obtienen el **user** y el **dispatch** del **reducer** a través del contexto de la aplicación enviado desde la raíz del proyecto o componente principal **HeroesApp**.
- El contexto es llamado **AuthContext** y el **value** que comparte, **user** y **dispatch**, son los que permiten actualizar o consultar el estado de la aplicación desde cualquier componente hijo.



Debido a que las pruebas se van a separar en dos bloques diferentes, el equipo de iMaster reestableció algunas versiones de su código antes de tener todo el grupo de errores juntos, los errores actuales corresponden al **router** y al **reducer**. Para la semana final, se compartirá todo el código con todos los errores presentes.

Ahora la lista de los errores que se han detectado hasta este momento es:

- El estado inicial de un usuario no autenticado debe ser false, sin embargo, el **authReducer** tiene un valor diferente, al parecer cuando se le envía una action no identificada retorna el error.
- Nuestro compañero Archivaldo se ha intentado autenticar, pero su nombre no es establecido correctamente, al parecer el **authReducer** tiene algún inconveniente en la actualización del **login**.
- Y, por el contrario, al hacer **logout**, el app no esta borrando el nombre del usuario ni poniendo la bandera **logged** en false.
- Ahora el **AppRouter** debe mostrar el componente Marvel a la hora de tener un usuario autenticado pero el **router** que controla el comportamiento del **login**, que es el **PublicRoute** al parecer no hace un correcto uso del **Navigate** de **react-router-dom**
- Ahora en las rutas privadas se tiene un espacio para mostrar el nombre del usuario en un espacio con una clase **text-info**, pero se usa una clase diferente y no se lee muy bien el nombre, pero claro, tampoco muestra el nombre, al parecer muestra la bandera **logged**, se puede encontrar en el **Navbar**.
- El título en cada página debe estar en una etiqueta H1, al parecer no todos cumplen.
- Finalmente, el **router** privado debe almacenar en el **localStorage** la última ruta visitada, se recomendó utilizar el nombre **lastPath**, sin embargo, algo falla al momento de recuperar el elemento.

Su tarea, como ya se debe imaginar, es corregir dichos errores y levantar nuevamente el aplicativo.



El equipo de iMaster envía unas capturas de como se vería el aplicativo funcionando correctamente:

# Login

Login

Ilustración 1 Login funcionando

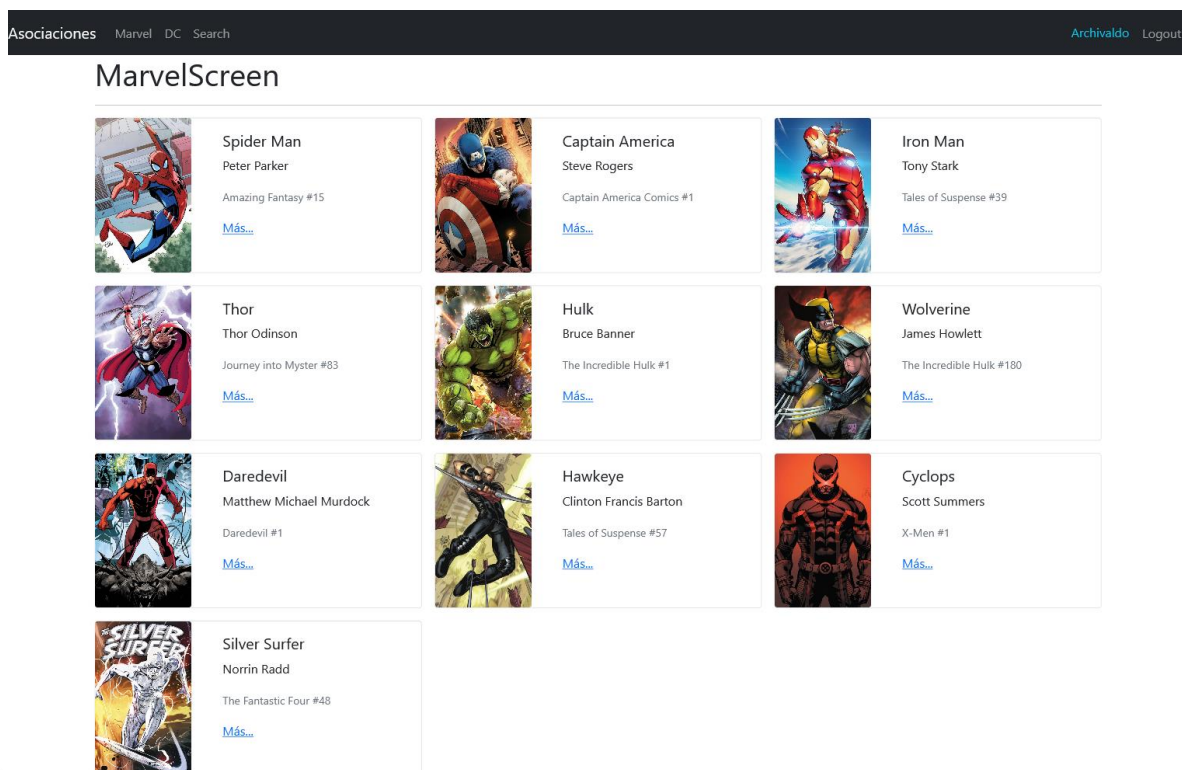


Ilustración 2 Marvel Screen List

[Más...](#)



[Más...](#)



[Más...](#)



[Más...](#)



[Más...](#)



[Más...](#)



[Más...](#)



[Más...](#)



[Más...](#)

[Más...](#)

**‘Mision TIC 2022’**





## Wonder Woman

**Alter ego:** Princess Diana

**Publisher:** DC Comics

**First Appearance:** All Star Comics #8

### Characters

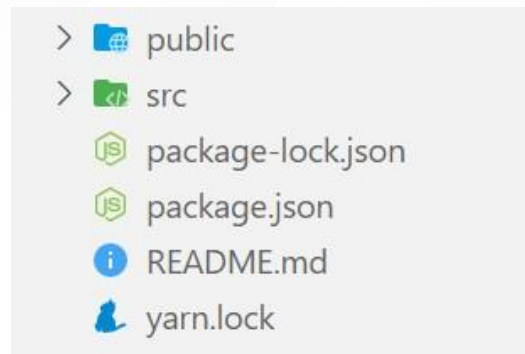
Princess Diana

[Regresar](#)

*Ilustración 5 Hero Screen Wonder Woman*

Al igual que con nuestros proyectos anteriores, tenemos algunas herramientas a nuestro favor y algunos requerimientos que debemos cumplir/installar para poder ejecutar el proyecto.

1. Debe contar con Node.js instalado en su equipo, para ello solo requiere ir al sitio oficial (<https://nodejs.org/es/>) y seleccionar el paquete que se ajuste a su máquina. La interventoría recomendó utilizar la versión LTS.
2. Verifique la instalación de Node.js escribiendo en su terminal **node -v**.
3. Una vez todo correcto, puede descomprimir el paquete de código que se le ha enviado y almacenarlo en una carpeta nombrada con su documento de identidad, todos los documentos son necesarios, no olvide incluir ninguno.
4. Con la carpeta nombrada correctamente, abra el proyecto con el editor de código de su preferencia, debe visualizar una estructura como la siguiente:



*Ilustración 6 Estructura del proyecto*

5. Ahora ya está listo para ejecutar el proyecto, para ello debe abrir una terminal dentro de la carpeta del proyecto y ejecutar el comando ***npm install***.
6. Si todo está correcto, deberá tener una nueva carpeta llamada ***"node\_modules"***.
7. Recuerde que cuando crean que ya cumple con todos los puntos solicitados, puede ejecutar el validador, para hacerlo debe abrir una terminal dentro de la carpeta del proyecto y ejecutar el comando ***npm run test***.
8. Ahora solo resta comprimir el proyecto y enviarlo a la plataforma.
9. Antes de comprimir la carpeta nombrada con su número de documento de identidad, donde se encuentran todos los archivos, debe eliminar la carpeta ***"node\_modules"*** del proyecto.
10. Ahora ya puede crear su archivo comprimido, el único formato permitido es ***.zip***.
11. El archivo comprimido ***.zip*** debe conservar como nombre su número de documento de identidad para proceder a la carga en la plataforma.
12. Finalmente siga las instrucciones de la plataforma para recibir una evaluación.

Recuerde que el reto puede ser subido por un único integrante del equipo, indicando el número de documento de identidad de los demás compañeros que hayan trabajado en el proyecto, separados por comas y sin puntos u otra clase de separadores, ej.



1111111,2222222,33333333,4444444

La carpeta del proyecto debe ser nombrada solo con el número de documento de identidad del usuario que hace la carga a la plataforma así la presente por todo su equipo, ej.

