



Creando una aplicación con create-react-app

Práctica Integradora

Objetivos

- Vamos a crear una aplicación con **create-react-app**. Luego vamos a identificar componentes, agregarles estilos y hacer consultas a una API externa.
- Al finalizar la aplicación, tendremos un panorama bastante completo de React.

Requisitos

Tener instalado **npm**: para correr el comando `npx`, que es una extensión de `npm`. Este nos va a permitir usar el comando **create-react-app**.

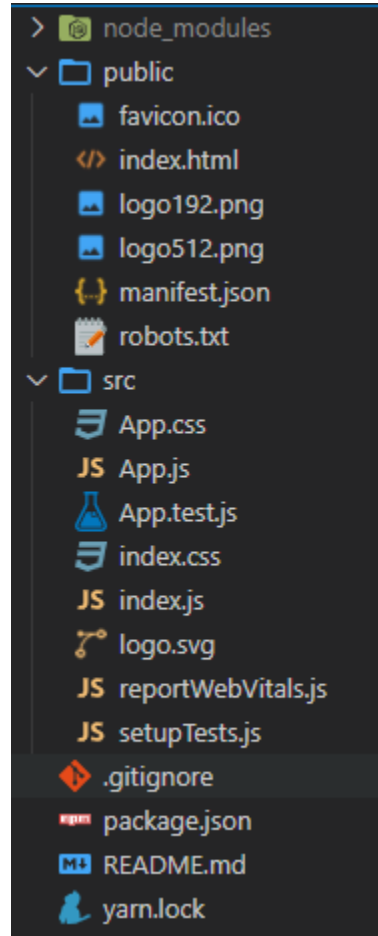
Tener una cuenta en **Giphy**: esta es la API que vamos a consumir desde nuestra app. Te dejamos el link acá: <https://developers.giphy.com>.

Recomendamos bajar los siguientes archivos para que el estilo del sitio sea más amigable: [link](#).



Micro desafío - Paso 1:

- Iniciar un nuevo proyecto con **npx create-react-app giphy-app**. Al finalizar la instalación, debemos revisar que dentro de la carpeta se vea algo parecido a la siguiente imagen.



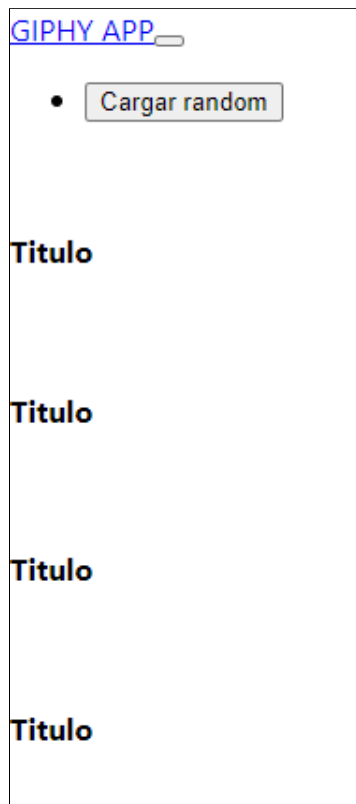
Puede que no sean exactamente los mismos archivos (estos varían dependiendo de la versión de npm y otras herramientas).

- Luego, debemos correr el comando **npm start** para ver que todo funcione correctamente.



Micro desafío - Paso 2:

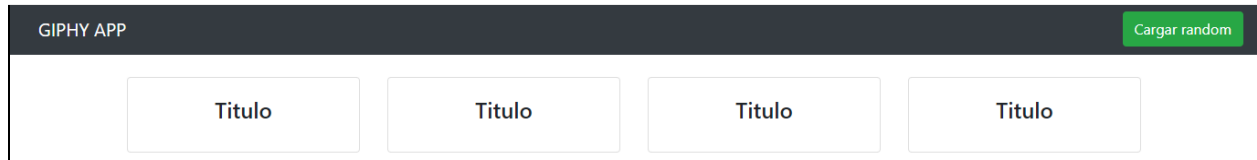
- Vamos a abrir los archivos que dejamos al principio y vamos a identificar los componentes que podemos ver.
- Una vez identificados los componentes, vamos a empezar a mover el HTML a nuestra app de React. Recomendamos crear una carpeta llamada **components** donde vamos a crear y guardar todos los componentes de la aplicación.
- Solucionemos los errores que nos trae hacer el traspaso de **HTML a JSX**.
Recordemos que JSX no es HTML, por lo que tenemos que tener cuidado con el cierre de las etiquetas, con los nombres de las propiedades y con retornar dos elementos HTML diferentes. Si todo sale bien, debería habernos quedado algo parecido a esto:



Pero ¿por qué se ve así? ¿No tenía estilos asociados?

En la **UI** que proporcionamos también hay un CSS, queda en vos pensar cómo podrías agregarlo.

Con los estilos agregados, debería quedar algo como lo siguiente.



En caso que no haya quedado igual, habrá que revisar los errores en la consola y verificar que estemos importando el CSS correctamente.



Micro desafío - Paso 3:

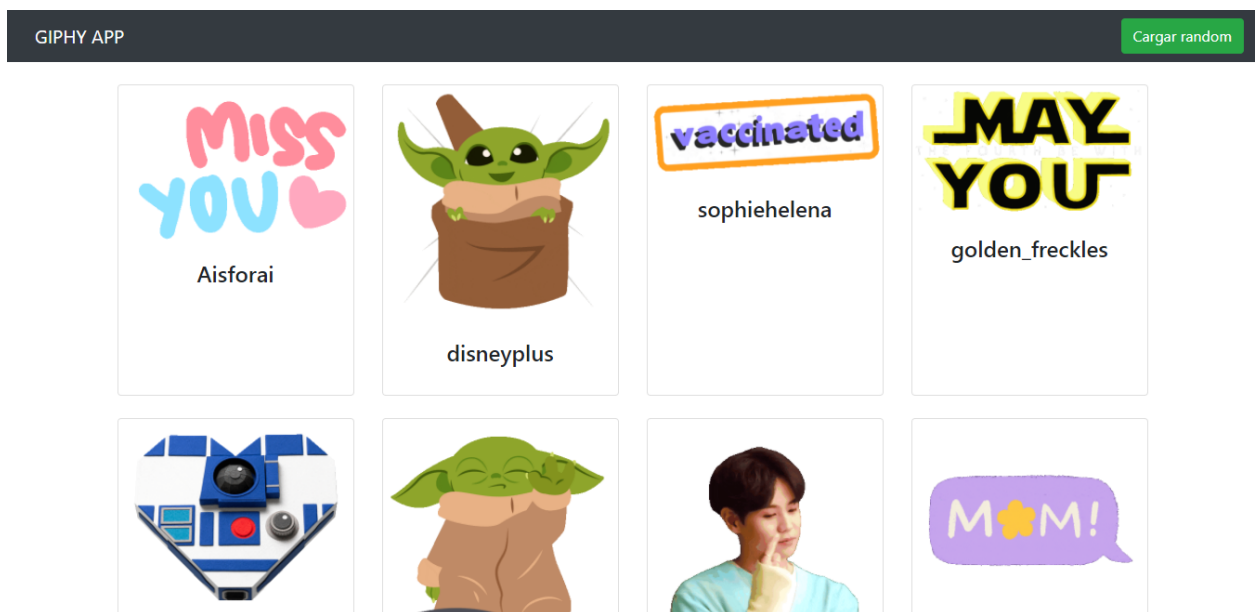
1. Una vez atomizado todo el sitio con sus respectivos componentes, vamos a empezar a darle funcionalidad. La funcionalidad de mostrar gifs, pero... ¿quién debería tener dicha funcionalidad? Esto depende mucho de cómo hayamos estructurado los componentes. Normalmente, cuando son pocos componentes, no atomizamos tanto para no tener que hacer mucho traspaso de información. Supongamos que tenemos tres componentes nada más. **App**, que va a tener toda la lógica; el **Nav**, que se va a encargar de actualizar los Gifs; y el componente **Gif**, que se va a encargar de mostrar la información de un gif.
2. Dentro del componente **App**, vamos a empezar a hacer un componente de clase con todos sus métodos. Recordemos que los más importantes son **render()**, **componentDidMount()** y **componentDidUpdate()**.
3. Hagamos un llamado a la API de giphy. Esta es la URL de la documentación para saber cómo llamarla: <https://developers.giphy.com/docs/api/endpoint#trending>. El endpoint que vamos a usar es el siguiente:
https://api.giphy.com/v1/gifs/trending?api_key=tuapikey.

Algo a tener en cuenta es que necesitas una API Key, y solo se obtiene si estás registrado.

Dentro del objeto del **Gif** tenemos un objeto que se llama **images**, de ahí vamos a seleccionar la propiedad **downsized** u **original**. O sea, vamos a tener que llamar a: **data[0].images.downsized.url** a la hora de mostrar realmente la imagen.

Podríamos hacer un **map** para modificar los valores reales del array y acortar esta llamada.

Si todo salió bien, podremos ver en el sitio algo parecido a esto:



Tené en cuenta que los gifs pueden variar dependiendo de cuándo hayas hecho el ejercicio.

Conclusión

React es una herramienta increíblemente poderosa. Esperamos que puedas aprovecharla al máximo. Si bien lo que nosotros conocemos de React hasta el momento es muy reducido, hay mucha más información en la documentación oficial.

Te dejamos para que pienses como podrías hacer la implementación del botón **“cargar random”** (una pequeña pista sería utilizar eventos).

iHasta la próxima!