Para testar nuestro proyecto vamos a utilizar Jest. Jest es un framework para testing desarrollado por Facebook. Entre sus características destacadas tenemos:

- Funciona con código síncrono y asíncrono
- Tiene soporte para `async/await`
- Podemos testar componentes de React gracias a que integra un Fake DOM
- Tiene un buen rendimiento gracias a que lanza los tests en paralelo
- Se integra con Babel
- Podemos quedarnos a la espera de cambios en nuestros ficheros para volver a lanzar los tests
- Incluye *coverage* del código. Esto son estadísticas sobre los tests de nuestro código. Nos permite conocer que ficheros o métodos no se están testando

Además, este es el framework que tenemos por defecto en `create-react-app`.

Enzyme

Enzyme de Airbnb es una librería de utilidades de testing para React. Esta nos permite montar nuestros componentes para realizar comprobaciones sobre el código HTML que renderizan. También podemos interaccionar con los componentes comprobando como varían las propiedades y el estado.

Además posee distintos métodos de montaje de componentes. Dependiendo de lo que necesitemos testar, montaremos nuestro componente con un método u otro. Los métodos disponibles son:

Shallow Rendering

Shallow Rendering monta nuestro componente hasta el primer nivel de renderización. Es decir, si montamos un componente que a su vez tiene otros componentes como hijos, solo tendremos disponible este primer nivel como código HTML.

Este método de renderización es muy útil para comprobar componentes como unidades independientes. De este forma, nos aseguramos que los componentes hijos no afectan indirectamente al propio componente.

Además, al renderizar solo un nivel, estos tests son muy rápidos.

Mount Rendering

Mount Rendering monta toda la jerarquía de componentes y lanza los métodos del ciclo de vida de cada componente. `mount` requiere de un fake DOM ya que necesita interaccionar con las APIs del DOM. Esto no representa ningún problema porque Jest ya integra uno.

Este método es mucho más completo y nos permite realizar comprobaciones más exhaustivas. Por ejemplo, podemos comprobar que al hacer click en un elemento que renderiza un componente hijo obtenemos el resultado en el padre

Static Rendering API

Static Rendering API nos permite realizar comprobaciones sobre el HTML que generan nuestros componentes. Al igual que `mount` , también monta toda la jerarquía de componentes pero no lanza eventos del ciclo de vida de los componentes ni acceder a los métodos y propiedades de estos.

Ejemplos

Vamos a testar algunos de nuestros componentes como ejemplo:

- Header.test.js. **NOTA**: En el video utilizo `shallow` ya que es el método que más se utiliza para testar nuestros componentes. No obstante, al no necesitar testar las propiedades de nuestro componente es mejor utilizar `Static Rendering API`
- RepositoryList.test.js

Para lanzar los tests solo necesitamos ejecutar `npm test` en nuestro proyecto.