

Watch:  **Automatizar tests de accesibilidad con Jest Axe**

15:28 |



Puedes encontrar el código de este vídeo en [GitHub](#).

La accesibilidad en el desarrollo web es muy importante por tres motivos:

- Permite que todo el mundo pueda acceder al contenido independientemente de sus capacidades
- Ayuda al posicionamiento de nuestras páginas
- Finalmente, tal vez el menos evidente: hace que nuestros tests sean más fáciles de escribir.

Si tenemos un formulario sin labels, donde el botón no tiene texto, para conseguir interactuar con todos los elementos tenemos que recurrir o bien a añadir atributos `data-test-id` en nuestro código, o bien recurrir a confiar en el orden de los elementos html y hacer cosas así:

```
const name = screen.getByText(/name/i);
userEvent.type(name.nextSibling, "Jane");
```

Vemos que esto es más complicado y frágil que si el input tuviera un label asociado:

```
const nameInput = screen.getByLabelText(/name/i);
userEvent.type(nameInput, "Jane");
```

Jest Axe es una librería que extiende los asserts de Jest y nos permite pasar una serie de checks en nuestros componentes para validar su accesibilidad y ver que problemas podemos solucionar. Después de instalar la librería, podemos usarla de la siguiente manera:

```
import { axe, toHaveNoViolations } from "jest-axe";
import { render } from "@testing-library/angular";
import { MyComponent } from "../components/my-component.component";

expect.extend(toHaveNoViolations);

describe("MyComponent", () => {
  it("should be accessible", async () => {
    const { container } = await render(MyComponent);

    expect(await axe(container)).toHaveNoViolations();
  });
});
```

Es importante tener en cuenta que si nuestro test pasa, esto **no es garantía 100% de que nuestro código es accesible**. Hay muchas cosas que tenemos que tener en cuenta y no todas son automatizables. Aún así, es un buen punto de partida y nos ayudará a aprender más sobre accesibilidad.