

Watch: 🦉 ¿Por qué el testing en frontend está roto?

17:17

¿Por qué testearmos? Queremos que los tests nos den confianza, no que nos suban métricas como coverage que, por sí solas, no aportan valor. ¿Cómo conseguimos esto?

## Testear como lo haría un usuario

Cuando escribimos tests en frontend tenemos que ponernos en la situación de la persona que va a utilizar nuestra aplicación. Un test que mire el atributo que tiene un tag, no aporta valor. Lo que hace un usuario es buscar cierto elemento que contenga un texto:

```
✗ expect(MessageComponent.attributes().msg).toEqual("toggled message");
```

```
✓ expect(screen.getByText("toggled  
message")).toBeInTheDocument();
```

## Testear la realidad

Si testeamos componentes de manera aislada, perdemos partes de funcionalidad que pueden ser críticas. Por ejemplo, si usamos `shallowMount` para montar el componente, todos los componentes hijos van a ser reemplazados por un tag vacío como `<message-stub>``</message-stub>`. Esto hace que nuestros tests sean incompletos.

```
✗ const wrapper = shallowMount(MessageToggle);
```

```
✓ const wrapper = mount(MessageToggle);
```

## Testing Library

Testing Library nace con el objetivo de evitar que podamos aplicar las prácticas que veíamos anteriormente y forzarnos a testear nuestros componentes como lo haría un usuario. No podremos acceder a la instancia de nuestro componente para llamar directamente a una función o acceder a una propiedad, en lugar de eso buscaremos los elementos con queries como `getByRole`, que nos permite buscar un botón con determinado nombre. Aquí tienes un ejemplo de un test hecho con Testing Library:

```
describe("Error Toggle", () => {  
  it("should display message when button is clicked", () => {  
    render(ErrorToggle);  
  
    const button = screen.getByRole("button", { name: /toggle/i  
  });  
  userEvent.click(button);  
  
  const message = screen.getByRole("alert");  
  expect(message).toHaveTextContent(/contact us/i);  
});
```

```
});  
});
```

## Links relacionados:

- [Testing Library](#).
- [Vue Test Utils Jest Example](#)