

Watch:  **Testeando animaciones**

11:24 |



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

Una particularidad de testear en el frontend es que en ocasiones tenemos que lidiar con animaciones. Por ejemplo, cuando queremos testear que un modal desaparece cuando clicamos el botón de cerrar, pero hay una transición de X segundos hasta que éste desaparece completamente. Si testeamos este comportamiento sin falsear la transición de alguna manera, nos encontraremos con tests que tardan mucho en pasar.

Si la librería o implementación propia que usemos para ejecutar esa transición controla el tiempo con JavaScript, mediante una variable de entorno podemos convertir ese tiempo en o:

```
const duration = 3000 * (process.env.TIME_SCALE_FACTOR || 1);
```

Así, podemos pasar esa variable al componente que usemos, en este caso:

```
<Transition
  appear={true}
  mountOnEnter={true}
  unmountOnExit={true}
  in={showContent}
  timeout={duration}
>
```

Pero existen otro tipo de librerías a las que no pasamos este tiempo mediante una prop, sino que cambiamos el tiempo por CSS. Por ejemplo, con Vue y su componente `transition`, al que podemos sólo pasarle un nombre y definir las clases de transición en CSS:

```
<transition name="fade">
  <div v-if="showContent">
    <h2>A subtitle</h2>
    <p>Lorem ipsum dolor sit amet</p>
  </div>
</transition>
```

```
.fade-enter-active,
.fade-leave-active {
  transition: opacity 5s;
}
.fade-enter,
.fade-leave-to {
  opacity: 0;
}
```

En casos así, podríamos usar una variable CSS y modificarla desde JS, pero es más tedioso. En estos casos podríamos considerar hacer un stub del componente `transition`, que es lo que vemos que ya hace por defecto cuando ejecutamos los tests (podemos verlo usando `screen.debug`). Al ser un componente propio del framework que estamos usando, y que en realidad el control de si mostramos o no el contenido está fuera del

componente `wrapper transition`, podemos considerar que hacer un stub no es arriesgado y que el componente no se va a romper sin que los tests nos den error. Si usamos una librería externa, debemos valorar si nos es posible usar la opción de alterar el tiempo mediante una variable de entorno, o bien recurrir a doblar la librería.