



4.275 pts



Curso de Angular: Unit Testing para Formularios

¡No te rindas!

Necesitas una **calificación mínima de 9.0** para aprobar.

Vuelve a intentarlo en 05 horas, 46 minutos, 13 segundos

8.46

Calificación

11 / 13

Aciertos

Resumen

1. ¿Cuál es el objetivo de la siguiente línea de código `defer() => Promise.resolve(data))`?`

Retorna una promesa se resuelve inmediatamente para poder probarlo [REPASAR CLASE](#) escenario de pruebas con fakeAsync y tick

2. ¿Cuál es el beneficio de usar un data-testid en los elementos en vez de usar clases y selectores comunes?

Al utilizar un data-testid evitamos caer errores de quitar o renombrar clases a elementos que queremos probar y son más exactos. ✓

3. ¿Cuál de los siguientes códigos es el correcto para obtener el texto de un elemento?

```
`const debugElement = fixture.debugElement.query(By.css(...)); const text = debugElement?.nativeElement?.textContent;`
```



4. En el curso hicimos dos helpers para clic clickEvent y clickElement ¿Cuál de la diferencia entre cada una?

clickEvent lanza el evento (click) de Angular asociado al elemento mientras que clickElement lanza el evento click nativo de HTML sobre el elemento.



5. ¿Es necesario agregar el módulo de ReactiveFormsModule al módulo de pruebas en el método configureTestingModule, cuando queremos hacer pruebas a componentes con formularios reactivos?

Verdadero



6. Teniendo en cuenta que el campo emailField en un FormControl ¿Cuál sería la forma correcta de verificar si es un campo inválido?

```
expect(component.emailField?.invalid).toBeTruthy();
```

[REPASAR CLASE](#)

7. ¿Cuál es la forma correcta de enviar el evento tipo blur de un elemento?

```
inputEl.dispatchEvent(new Event('blur'));
```



8. ¿Cuál es la razón de en algunos escenarios de pruebas al llenar un campo desde UI nos toca enviar el evento input y blur a la vez?

Esto depende de la lógica de nuestro formulario, pero hay errores en campos que solo se activan una vez nos enfocamos en otro elemento de la UI.



9. ¿Cuál es el método correcto para llenar la información de un formulario reactivo?

patchValue



10. ¿Cuál es el método correcto para verificar si un método fue llamado?

toHaveBeenCalled



11. ¿Cuál es de los siguientes eventos es el correcto para notificar el cambio de un checkbox?

dispatchEvent(new Event('change'));



12. ¿Cuál es de los siguientes métodos usamos para probar que una función lanzó una excepción?

toThrow



13. ¿Cuál es la diferencia entre una validación síncrona vs. una asíncrona?

Las validaciones síncronas responden de forma inmediata y no necesitan de ninguna fuente de datos externa para validar la información, las asíncronas van a tardar x tiempo en responder y normalmente consumen una fuente de datos externa como una API.



Ver menos

Siguiente curso que podría interesarte



Curso de Angular:

Unit Testing Routing



Curso de Angular: Unit Testing para Rutas

Por: Nicolas Molina

[Curso de Angular: Unit Testing para Rutas →](#)

[Volver al Home](#)