

Desarrollo y programación de Aplicaciones Avanzadas Angular

11 Formularios

CORE
networks



Modelo template driven

En el módulo:

```
import { FormsModule } from '@angular/forms';
```

En la plantilla:

```
<form #nombreForm ="ngForm"  
      (ngSubmit )="metodo(nombreForm)">  
  <input ... name="propiedad" ngModel>  
  ...
```

Modelo template driven

En la clase del componente:

```
metodo(objetoForm) {  
  this.objeto = {  
    propiedad: nombreForm.controls.propiedad.value,  
    ...  
  };  
}
```

Validación

Estados del formulario/campo

- pristine | dirty
- untouched | touched
- invalid | valid

```
#propiedad="ngModel" propiedad.estado //boolean
```

Validación CSS

Clases CSS identificadas según los estados

- ng-pristine | ng-dirty
- ng-untouched | ng-touched
- ng-invalid | ng-valid

```
.ng-estado {  
    /* reglas css */  
}
```

Modelo model driven (reactivo)

En el módulo:

```
import { ReactiveFormsModule } from '@angular/forms';
```

En la plantilla:

```
<form [formGroup]="nombreForm"  
      (ngSubmit)="metodo()">  
  <input ... formControlName="propiedad">  
  ...
```

Modelo model driven (reactivo)

En la clase del componente:

```
import { FormGroup,
        FormControl, Validators } from '@angular/forms';
...

ngOnInit() {
    this.nombreForm = new FormGroup({
        propiedad: new FormControl(valordeinicializacion),
        ...
    });
}
```

Modelo model driven (reactivo)

```
metodo() {  
    this.objeto = {  
        propiedad: this.nombreForm.get(propiedad).value,  
    };  
}
```


Validación de campos

En la clase del componente:

```
propiedad: new FormControl('', [Validators.validador]),
```

En la plantilla:

```
*ngIf="nombreForm.get('propiedad').estado..."  
...  
*ngIf="nombreForm.get('propiedad').hasError('validador')..."
```

Programación de validación

Archivos TypeScript con formato nombre.validator.ts:

```
import { FormControl } from '@angular/forms';

export class NombreValidator {
    static metodoValidator(control: FormControl) {
        // código devuelve true o null
    }
}
```