

# Actividad Lección 9: Formularios Reactivos

1. Por favor, indique las librerías y dependencias que se deben utilizar para poder utilizar formularios reactivos en Angular.

Deberemos utilizar la librería nativa llamada `ReactiveFormsModule`, del paquete `@angular/forms`, en nuestro archivo `app.module.ts`:

```
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';

import { ReactiveFormsModule } from '@angular/forms';
import { AppRoutingModule } from './app-routing.module';
import { AppComponent } from './app.component';

@NgModule({
  declarations: [
    AppComponent
  ],
  imports: [
    BrowserModule,
    AppRoutingModule,
    ReactiveFormsModule
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule {}
```

2. A continuación, indique las propiedades que se deben utilizar al implementar formularios reactivos en componentes de Angular.

En el archivo `{component}.component.ts`:

- Un objeto `FormBuilder`, que nos servirá para crear el propio formulario reactivo y validarlo
- Un objeto `FormGroup`, el propio formulario reactivo

```

public formGroup: FormGroup
constructor(private FormBuilder: FormBuilder) {
  this.formGroup = FormBuilder.group({
    username: ['example'],
    password: [''],
  })
}

```

En el archivo {component}.component.html:

- La propiedad [formGroup] para enlazar el .html al objeto formulario del .ts
- La propiedad formControlName para enlazar los input a los atributos del objeto FormGroup

```

<form [formGroup]="formGroup">
  <label for="username">
    Username:
    <input type="text" name="username" value="" id="username" formControlName="username" />
  </label>
  <br />
  <label for="password">
    Password:
    <input type="password" name="password" value="" id="password" formControlName="password" />
  </label>
  <br />
</form>

```

### 3. ¿Por qué es aconsejable utilizar formularios reactivos aunque sea para un formulario pequeño?

Porque los formularios reactivos son mucho más escalables a comparación de un formulario creado con valores unidos a variables de TS mediante ng-models en lo que respecta a dar nuevos campos y validar los mismos.