

# Cómo pasar variables como atributos en directivas de AngularJS

Las directivas de Angular sirven para encapsular código HTML de manera que nuestras aplicaciones puedan tener elementos reutilizables, también conocidos como componentes.

Cuando Angular apareció fue algo muy novedoso, aunque hoy en día podemos utilizar los Web Components y la librería Polymer para tener algo parecido y de forma nativa de HTML5.

Una de las cosas que suele ser algo complicada de implementar es cuando en nuestra directiva colocamos un atributo y queremos que ese atributo se utilice en el HTML interior de la directiva, y además que ese atributo que pasamos sea una variable dinámica que cambiará según la vista o estado en el que nos encontremos.

En este artículo vamos a ver como hacerlo con un pequeño ejemplo.

Imaginemos que queremos implementar una directiva que sea una caja de búsqueda. Y lo queremos programar como directiva porque lo vamos a usar en más partes de nuestra aplicación y ese es su objetivo, que las directivas sean reutilizables. Queremos que sea algo así:

```
<search-bar busqueda=""></search-bar>
```

Esta directiva tendrá el siguiente código HTML por debajo:

```
<div class="form-group">
  <input type="text"
    class="form-control"
    placeholder="Introduce tu texto a buscar"
    value="" />
</div>
```

¿Cómo hacemos para pasar el contenido del atributo busqueda al atributo value del input dentro de nuestra directiva? Veamos el código JavaScript de la directiva, suponiendo que pertenezca a un módulo myApp.

```
(function () {
  'use strict'

  angular
    .module('myApp')
    .directive('searchBar', searchBar);

  function searchBar () {
    return {
      restrict: 'E',
      scope: {
        busqueda: '@'
      },
      template: '<div class="form-group">'
        + '<input type="text" '
        + '      class="form-control"'
        + '      placeholder="Introduce tu texto a buscar"'
        + '      value={{busqueda}}'
        + '</div>'
    }
  }
})
```

```
})();
```

Veamos que hemos hecho aquí. Hemos pasado el contenido de la directiva como un string al atributo template. Si el HTML es muy grande, podemos tenerlo separado en un archivo HTML y pasarle la ruta de ese archivo en el atributo templateUrl

En la plantilla hemos indicado que el value del input sea value={{busqueda}}. Es decir, le pasamos como variable el contenido del atributo busqueda de la directiva.

Para poder hacer esto, y también, si el contenido que pasamos no es fijo, si no que viene de una variable, tipo:

```
<search-bar busqueda={{textoBusqueda}}></search-bar>
```

necesitamos indicarle a la directiva que tiene un scope propio y cuales son los atributos de ese scope:

```
scope: {  
  busqueda: '@'  
}
```

Indicándole @ le estamos diciendo de manera unidireccional que el valor del atributo busqueda se copie a la propiedad {{busqueda}} del scope de la directiva.

Usar @ tiene un problema, al ser unidireccional, si queremos cambiar el valor de la propiedad busqueda del scope de la directiva, no se cambia la propiedad en el controlador que estemos utilizando, y nos obliga a usar siempre la doble llave {{ }} si queremos enlazar la propiedad al controlador.

Para solucionar esto, podemos usar el caracter = que indica que la propiedad busqueda del scope de la directiva sea exactamente la propiedad del scope del controlador que estemos utilizando y tenga un atributo búsqueda.

Con estos cambios quedaría así la definición:

```
angular  
  .module('myApp')  
  .directive('searchBar', searchBar);  
  
function searchBar () {  
  return {  
    restrict: 'E',  
    scope: {  
      busqueda: '='  
    },  
    template: '<div class="form-group">  
      + <input type="text" '  
      +       'class="form-control" '  
      +       'placeholder="Introduce tu texto a buscar" '  
      +       'value={{busqueda}}' '  
      + </div>'   
    }  
  }  
}  
  
})();
```

y podríamos usar la directiva así:

```
<search-bar busqueda=textoBusqueda></search-bar>
```