

## Cómo interacciona el usuario con nuestra webapp

(cómo recoger los datos que introduce un usuario en las vistas)

Hay varias formas de poder recoger la información que un usuario introduce en un `<input>` de HTML o saber qué eligió en un `<select>` de HTML, o en un `<input>` de tipo checkbox. También veremos cómo reaccionar ante los diferentes eventos que ocurren cuando el usuario interacciona con nuestras vistas:

- Hace click en los botones
- Escribe algo en un input y pulsa INTRO o hace click luego en otra parte de la vista
- Cambia el valor de un checkbox o un input de tipo radio.
- Pasa por encima de un elemento
- Pulsa una tecla en concreto en el teclado
- Etc.

Página oficial: <https://angular.io/guide/user-input>

Otra página interesante que lo explica: <https://desarrolloweb.com/articulos/declaraciones-eventos-statements-angular.html>

## EVENTOS

En este enlace de la página oficial del Javascript aparecen todos los eventos del DOM disponibles ordenados por tipos y elementos a los que se pueden aplicar. Por ejemplo los que se pueden aplicar a elementos relacionados con los `<input>` del HTML, o los de teclado (pulsar una tecla, soltarla, etc), o relacionados con el ratón (los que lanza el ratón). **Si pulsáis en la 3ª columna de la tabla que aparece en el siguiente enlace podréis ver qué eventos hay en cada tipo de elemento.**

El enlace es: <https://developer.mozilla.org/en-US/docs/Web/Events>

Si queremos enviar desde nuestra vista los datos del evento ocurrido hasta nuestro controlador indicando el tipo adecuado de datos y usando `$event` como forma de recoger los datos, necesitamos saber de qué tipo será ese “event” en el controlador. Y podéis guiáros de la siguiente tabla para averiguarlo:

MouseEvent	FocusEvent	TouchEvent	DragEvent	KeyboardEvent
click	focus	touchstart	drag	keypress
mouseup	blur	touchmove	drop	keyup
mouseleave	focusin	touchcancel	dragend	keydown
mouseover	focusout	touchend	dragover	

Y luego del tipo genérico “Event” estarían los siguientes eventos (*los marcados en rojo son los más típicos*):

Event
close
<b>change</b>
invalid
play
reset
<b>scroll</b>
select
<b>submit</b>
<b>toggle</b>
waiting

Formas más usada de recoger datos de la vista de un componente y pasarlos al controlador:

- 1) Usar “**template variables**” de Angular (*enlace oficial: <https://angular.io/guide/template-reference-variables>*):

*Ejemplo de uso:*

*ejemplo.componente.html* → *es el fichero de la plantilla del componente*

```
<input
  type="number"
  id="edad"
  class="form-control"
  aria-labelledby="edadUsuario"
  #edadUsuario
  (input)="onEdadCambiada(edadUsuario.value)"
/>
```

*ejemplo.componente.ts* → *es el fichero controlador del componente*

```
onEdadCambiada(edad: string) {
  this.edad = edad;
  let parrafo: HTMLElement | null =
    document.getElementById('edadUsuario');
  if (parrafo != null) {
    parrafo.textContent = edad;
  }
}
```

## 2) Capturando el evento con **\$event**:

*Ejemplo de uso:*

*ejemplo.componente.html*

→ es el fichero de la plantilla del componente

```
<input
  type="text"
  id="ciudad"
  class="form-control"
  aria-labelledby="ciudadUsuario"
  (change)="onCiudadCambiada($event)"
/>
```

*ejemplo.componente.ts*

→ es el fichero controlador del componente

```
onCiudadCambiada(evento: Event) {

  /* El target es el elemento sobre el que sucedió el evento. En
  nuestro caso en un <input>.
  El tipo de los <input> en Javascript es HTMLInputElement.
  Por tanto, hay que castear ese target a HTMLInputElement para
  recoger la propiedad "value" que es la que contiene el valor
  escrito en dicho <input>
  */
  let target: EventTarget | null = evento.target;
  this.ciudad = (target as HTMLInputElement).value;
}
```

## 3) Usando el “**two-way binding**” de Angular o **[(ngModel)]** o también llamado “**banana in box**”:

*recordad que para usar [(ngModel)] hay que importar el módulo FormsModule en el app.module.ts porque no viene por defecto en el @angular/core.*

*Ejemplo de uso:*

*ejemplo.componente.html*

→ es el fichero de la plantilla del componente

```
<input
  type="text"
  id="empleo"
  class="form-control"
  aria-labelledby="empleoUsuario"
  [(ngModel)]="empleo"
/>
```

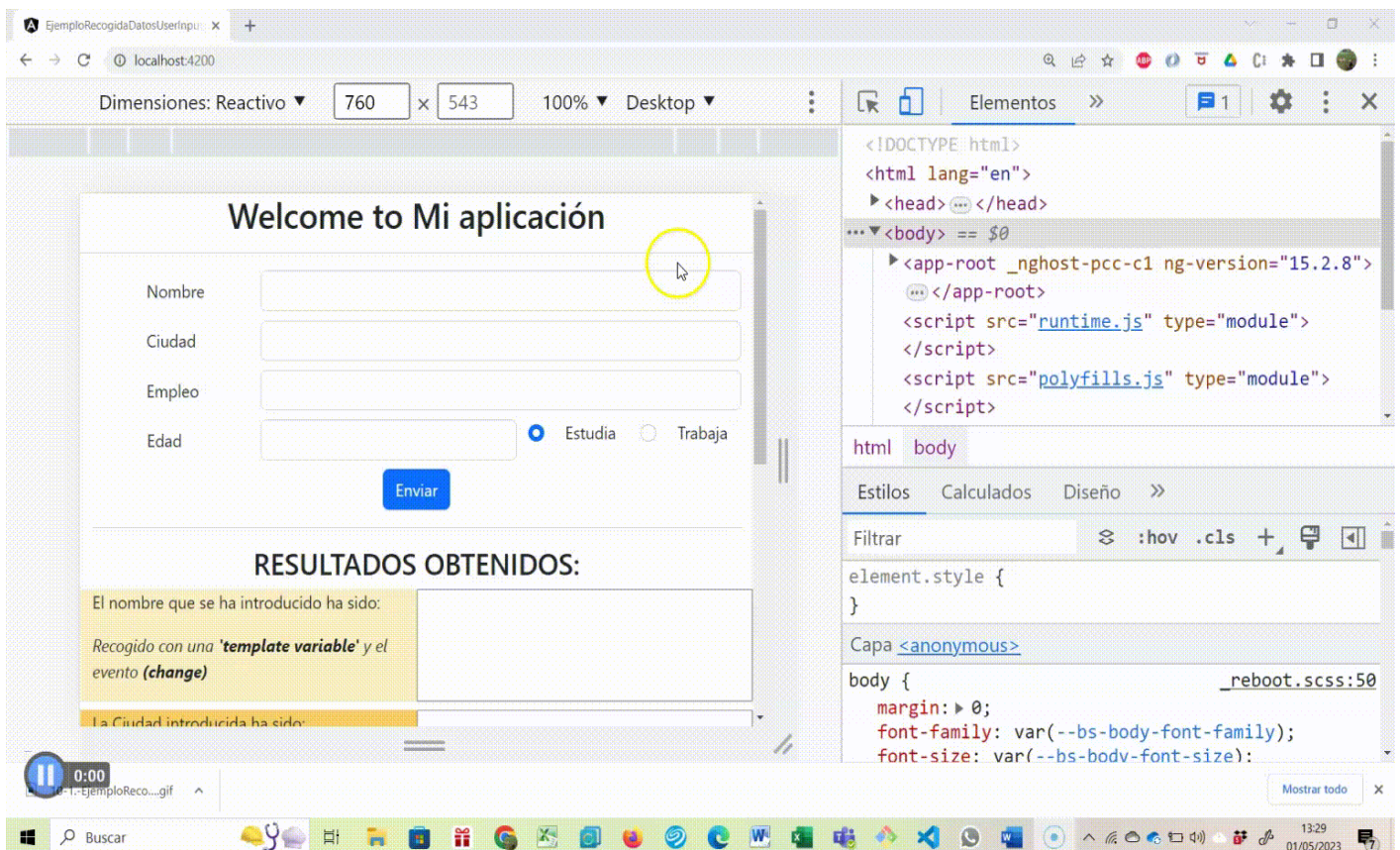
*ejemplo.componente.ts*

→ es el fichero controlador del componente

```
import { Component } from '@angular/core';

@Component({
  selector: 'app-formulario-usuario',
  templateUrl: './formulario-usuario.component.html',
  styleUrls: ['./formulario-usuario.component.css']
})
export class FormularioUsuarioComponent {
  ...
  empleo!: string;
  ... //resto de código
```

Aquí os dejo un gif que muestra un ejemplo de una webapp donde se recogen los datos de un formulario de varias formas que aparecen explicadas:



Y en el siguiente enlace a GitHub tenéis el código de esa app:

[https://github.com/alejandrtef/DAM2\\_PMDM\\_22\\_23/tree/main/ANGULAR/EjemploComoRecogerDatosDeUnaVista/EjemploRecogidaDatosUserInputConBootstrap](https://github.com/alejandrtef/DAM2_PMDM_22_23/tree/main/ANGULAR/EjemploComoRecogerDatosDeUnaVista/EjemploRecogidaDatosUserInputConBootstrap)