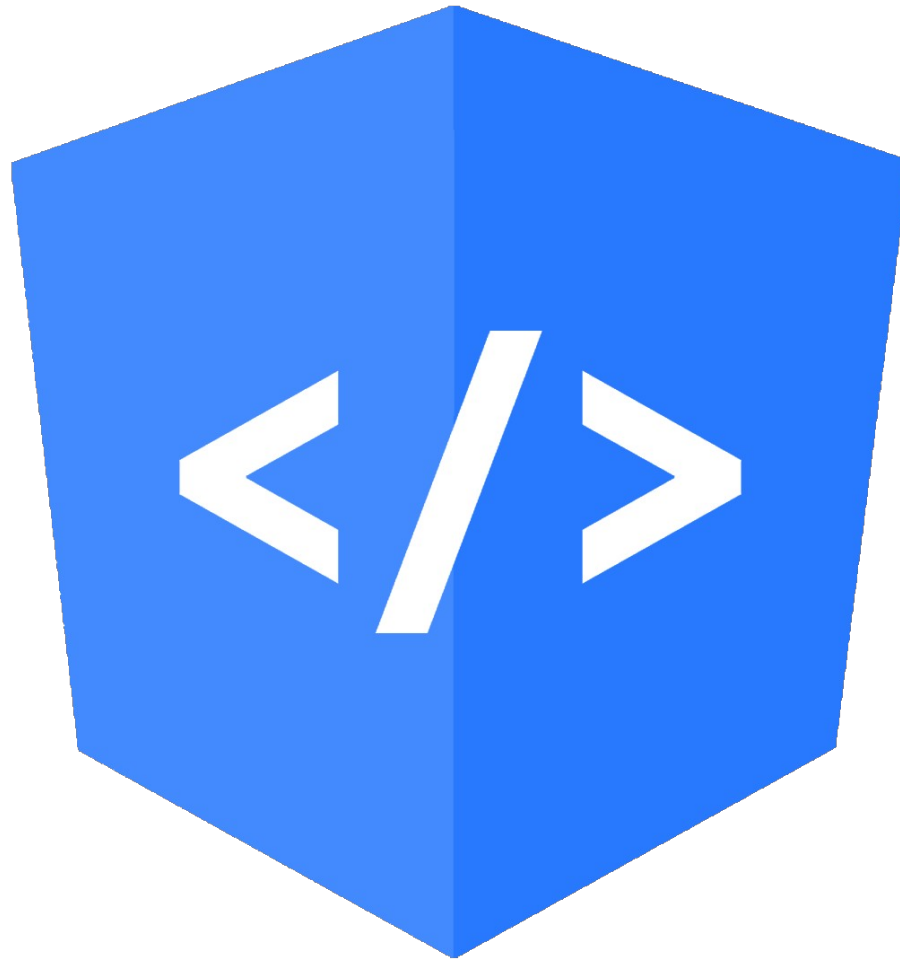
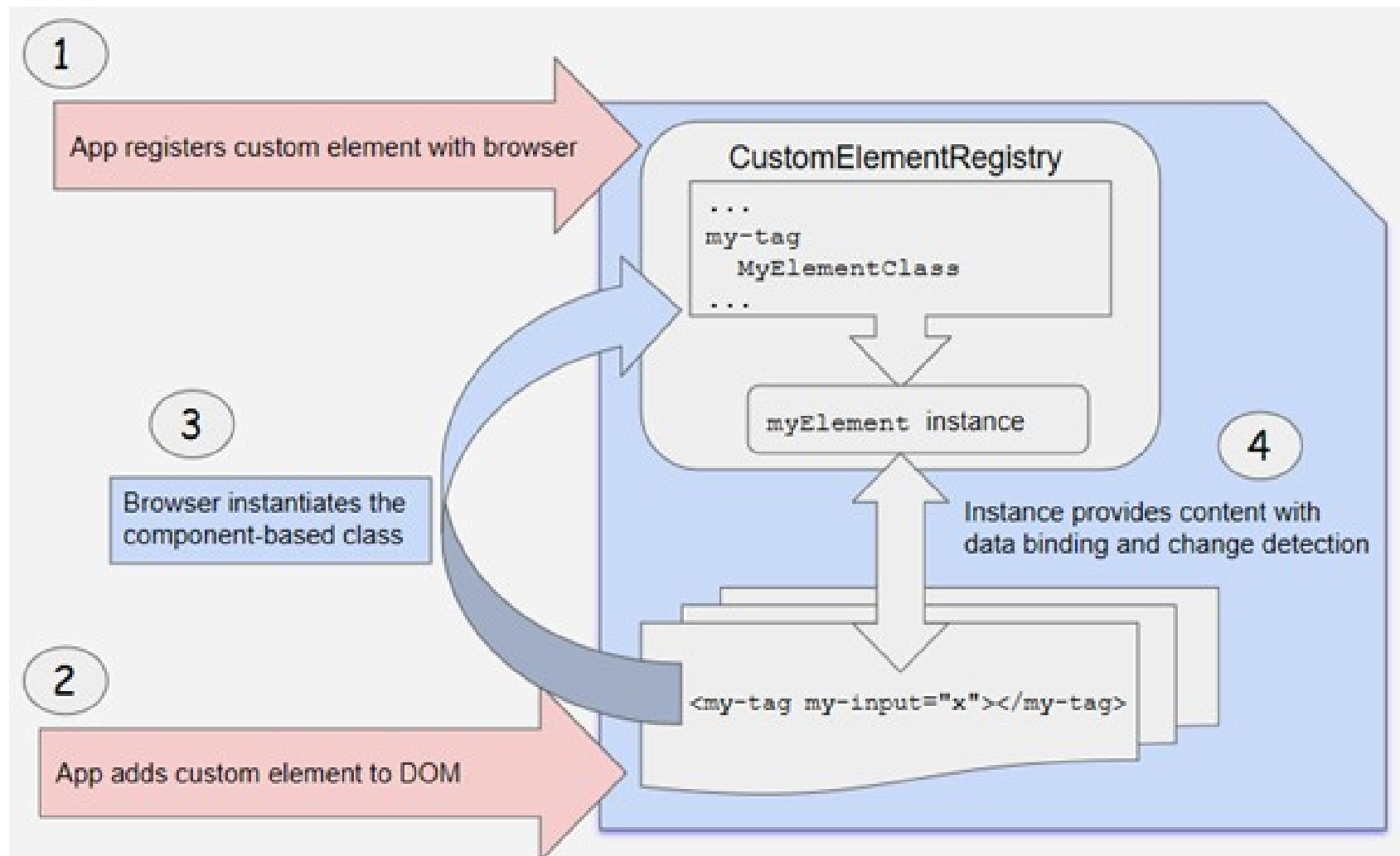


# LABORATORIO – Angular Custom Components

---



# LABORATORIO – Angular Custom Components



# LABORATORIO – Angular Custom Components

---

CREAR un nuevo COMPONENTE:

```
cd <dir padre>
```

```
ng g component <path/nombre>
```

```
ng g component main
```

```
ng g component about
```

```
ng g component clientes
```

```
ng g component proveedores
```

# LABORATORIO – Angular Custom Components

---

**En APP03 y MP3 vamos a aprender a:**

1. Cómo componer una vista a partir de varios componentes .
2. Cómo utilizar y renderizar unos componentes dentro de otros.
3. Cómo definir propiedades personalizadas de nuestros componentes.
4. Cómo usar componentes para definir contenido dentro de otros.
5. Cómo obtener información de los componentes.
6. Cómo emitir eventos desde nuestro componente hacia el mundo exterior.

# LABORATORIO – Angular Custom Components

---


WEB COMPONENTS con ANGULAR

Ver APP 03 !

# LABORATORIO – Angular Custom Components

---

Descargar proyecto de Stackblitz:

1. Descargar ZIP del proyecto, pulsando sobre botón  del proyecto.
2. Descomprimir en una carpeta <proy>
3. Abrir terminal en <proy>
4. `npm install` (para regenerar `node_modules`).
5. *`npm install @angular/cli@8 --save-dev`*
6. *`npm install @angular/compiler-cli@8 --save-dev`*

# LABORATORIO – Angular Custom Components

---

Audit:

1. `npm audit` (solo si es necesario)
2. `npm audit fix` (solo si es necesario)

Actualizando paquetes a su versión más actual: (esta es la opción más recomendable)

`npm install -g npm-check-updates`

1. `ncu` (otra opción: `ng update --all --force`)
2. `ncu -u -x typescript`
3. `npm install typescript@3.5.2`
4. `npm install`

Package.json

(puede ser necesario cambiar:) Typescript a 3.5.2

`npm install`

# LABORATORIO – Angular Custom Components

---

ng build --prod:

Puede ser necesario copiar src/environments de otro proyecto o crearlo nuevo!



# LABORATORIO – Angular Custom Components

---

## Micropráctica 03 – Selects combinados

### Micropráctica 03!

Definiendo nuestros propios Custom Elements:

Comunidad:  Provincia:

Comunidad:1

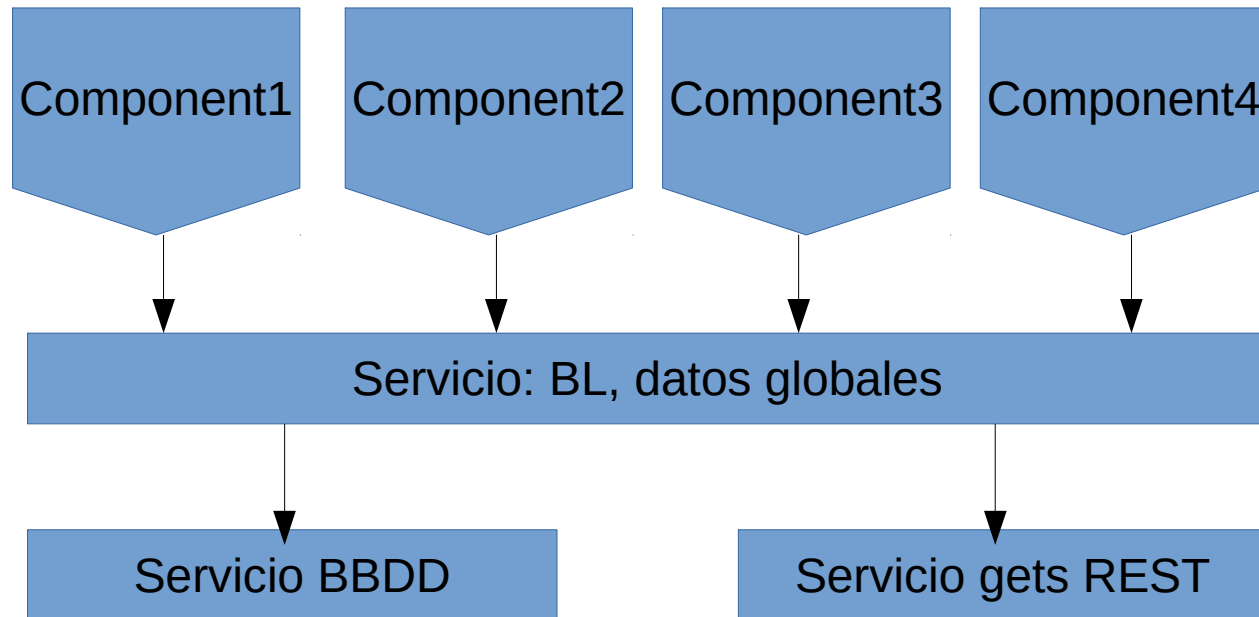
Provincia:3

Get propiedades del CustomElement

# LABORATORIO – Angular Custom Components

---

**Services:** componentes que “viven” globalmente...



# LABORATORIO – Angular Custom Components

---

**Services:**

ng g service global

# LABORATORIO – Angular Custom Components

---

## Service:

### Global.service.ts:

```
export class GlobalService {  
  constructor() { }  
  public datos_globales: Contacto = {  
    id: 0,  
    nombre: 'Ricardo Martín Manso',  
    telefono: '609117799',  
    email: 'ricardma@inf.uc3m.es',  
    tipo: 0,  
    habitual: true  
  };  
  public read_contacto(id:number):Contacto {  }
```

# LABORATORIO – Angular Custom Components

---

**Services:** Inyección de dependencias:

App.module.ts:

```
Exports: [  
  ...  
  providers: [GlobalService],
```

Detalle.component.ts:

```
constructor(private global:GlobalService) { }  
  
...  
  
console.log(this.global.datos_globales['usuario']);
```

# LABORATORIO – Angular Custom Components

---

**Routing:** Navegación entre las diferentes páginas del proyecto...

## app-routing.module.ts:

```
import { NgModule } from '@angular/core';
import { Routes, Router, RouterModule } from '@angular/router';

import { VerDetalleComponent } from './ver-detalle/ver-detalle.component';
import { ListadoComponent } from './listado/listado.component';
... importar todos los componentes "routables..."

const routes: Routes = [
    {path:'list',component:ListadoComponent},
    {path:"",redirectTo:'/list',pathMatch:'full'},
    {path:'edit',component:VerDetalleComponent},
    {path:'about',component:AboutComponent} ];

@NgModule({
  imports: [RouterModule.forRoot(routes)],
  exports: [RouterModule],
  providers: []
})
export class AppRoutingModule { }
```

# LABORATORIO – Angular Custom Components

---

**Routing:** Navegación entre las diferentes páginas del proyecto...

**app.module.ts:**

```
import { AppRoutingModuleModule } from './app-routing.module';
```

```
@NgModule({
```

```
  imports: [  
    AppRoutingModuleModule,  
    ...]
```

# LABORATORIO – Angular Custom Components

---

## Routing:

app.component.html

```
<router-outlet></router-outlet>
```

<http://localhost:4200/list>

<http://localhost:4200/about>



# LABORATORIO – Angular Custom Components

---

**Routing:** cómo pasar parámetros en la url...

<http://localhost:4200/edit/1>

## app-routing.module.ts:

```
const routes: Routes = [  
    ...  
    {path:'edit/:id',component:VerDetalleComponent},  
    ... ];  
  
@NgModule({  
  imports: [RouterModule.forRoot(routes)],  
  exports: [RouterModule],  
  providers: []  
})
```

## ver-detalle.component.ts:

```
ngOnInit() {  
  this.route.paramMap.subscribe(params=>{  
    this.id=params['params']['id'];  
    this.otro=params['params']['otro'];  
  });  
}
```

# LABORATORIO – Angular Custom Components

---

**Routing:** cómo pasar parámetros en la url...

<http://localhost:4200/edit/1>

app-listado.html

```
...  
<a [routerLink]="['/edit',id]">Editar</a>  
...
```

# LABORATORIO – Angular Custom Components

---

**Routing:** cómo navegar desde el código...

Inyectar dependencia (router) en todos los componentes donde queramos usar...

ver-detalle.component.ts:

```
export class VerDetalleComponent implements OnInit {  
  
    constructor(private router: Router, private route: ActivatedRoute,  
                private global: GlobalService) { }  
  
    public boton_aceptar()  
    {  
        ...  
        this.router.navigate(['/list']);  
    }  
  
    public boton_editar(id)  
    {  
        this.router.navigate(['/edit',id]);  
    }  
}
```

# LABORATORIO – Angular Custom Components

---

**Routing:** paso de parámetros más completo y ocultos en url...

Enviar y recibir parámetros:

listado.component.ts:

```
public editar(id)
{
  this.router.navigate( ['/edit',id,   {a:'XXXXXX',b:57 } ]
    ,{state:{otro:'pepe'}});
}
```

URL: .../edit/2;a=XXXXXX;b=57

ver-detalle.component.ts:

```
NgOnInit() {
  Console.log(history.state.otro);
  this.route.paramMap.subscribe(params=>{
    this.id=params['params']['id'];
    this.otro=params['params']['a'];
  })
}
```