



El futuro digital
es de todos

MinTIC

Universidad
Industrial de
Santander



Mision
TIC2022

FRONTEND

CONTENIDO

- Frontend
 - Angular
 - MVC
 - Creación proyecto

Angular

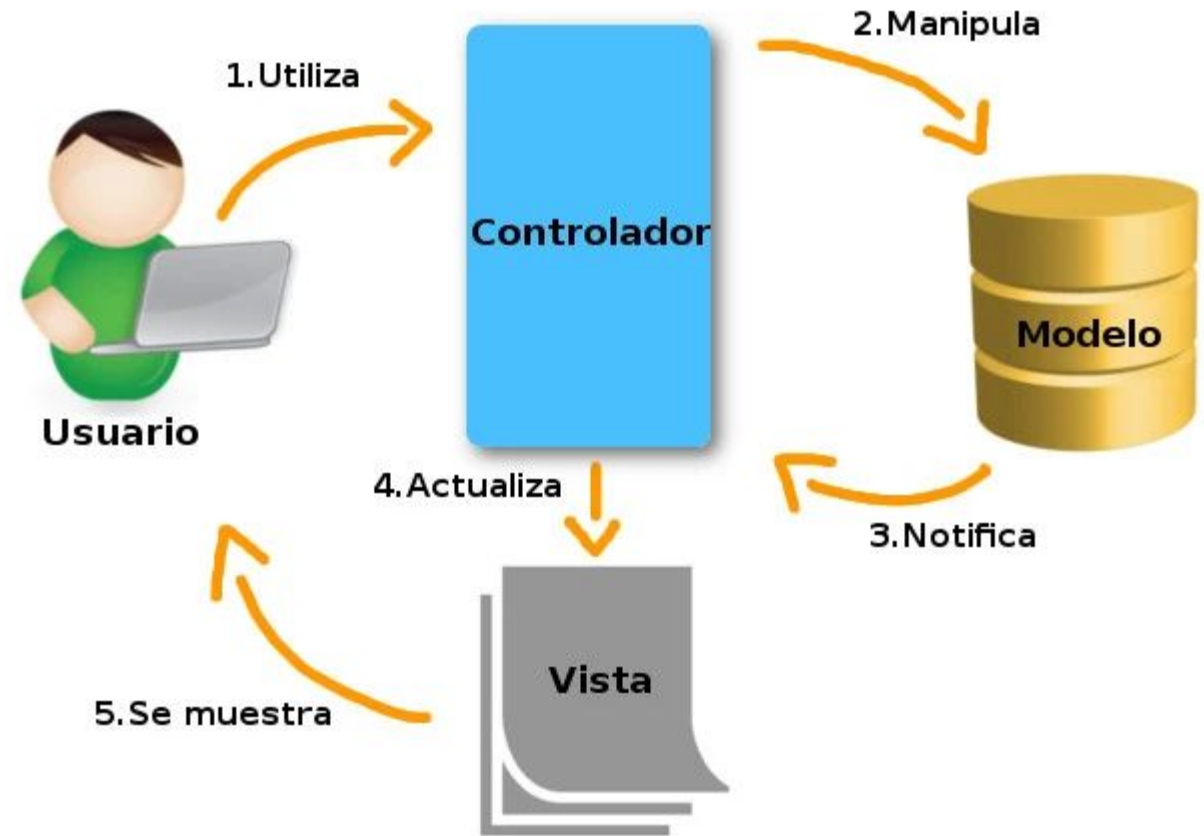
Es un framework opensource desarrollado por Google para facilitar la creación y programación de aplicaciones web de una sola página, las webs SPA (Single Page Application). Usa la arquitectura Modelo-Vista-Controlador (MVC).

- Marco de trabajo estandarizado
- Viene con todo lo que se necesita para trabajar
- Es modular



MVC

- **Modelo:** representa la estructura de datos de una aplicación de software.
- **Vista:** representa la interfaz del usuario, es decir, todo aquello que el usuario puede ver en la pantalla e interactuar.
- **Controlador:** se puede ver como el intermediario entre la vista y el modelo. El modelo y la vista nunca se relacionan directamente.



Angular CLI

Angular Cli es la herramienta con la que vamos a poder generar aplicaciones y administrirlas. Generar aplicaciones, componentes, directivas, pipes ...

Instalar: `npm install -g @angular/cli`
validar: `ng --version`



```
> npm install -g angular-cli
> ng new my-dream-app
> cd my-dream-app
> ng serve
```

Generando el nuevo proyecto

Comando: `ng new nombre-proyecto`



```
> npm install -g angular-cli  
> ng new my-dream-app  
> cd my-dream-app  
> ng serve
```

Bloques fundamentales

Componentes

Rutas

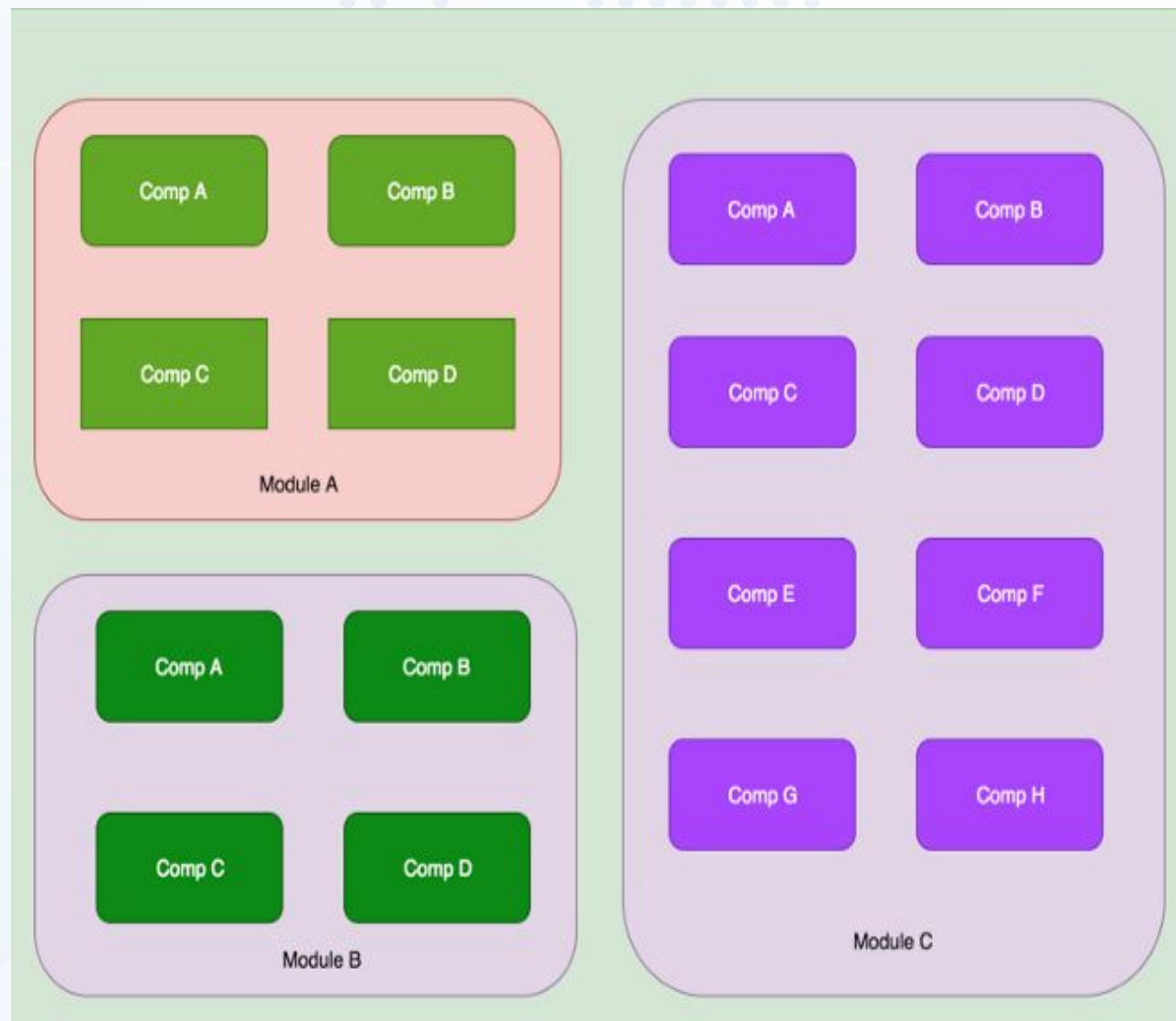
Directivas

Servicios

Módulos

Módulos

Agrupar un conjunto de artefactos Angular: componentes, directivas, pipes y servicios que forman parte de ese mismo módulo. Dicho esto, representa una agrupación lógica en lo que podríamos llamar área funcional de nuestra aplicación (ej. módulo de contactos, módulo de administración,...)



Componentes

Un componente en Angular es un bloque de código re-utilizable, que consta básicamente de 4 archivos: un CSS, un HTML y un TypeScript.

Comando: `ng generate component nombre-componente`

Componente

Typescript
(Lógica)

HTML
(Vista)

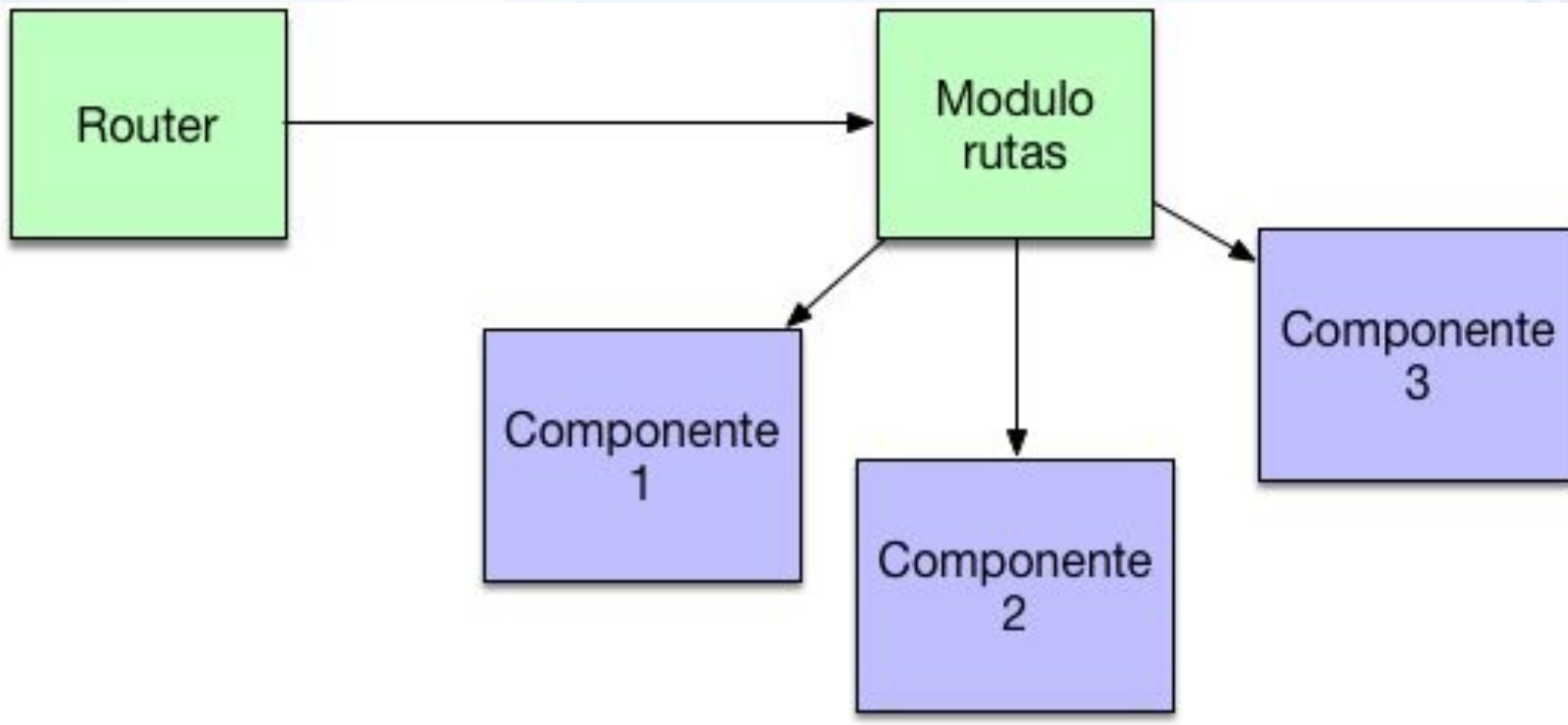
CSS
(Estilos)

ngOnInit vs Constructor

Constructor	ngOnInit
A constructor is not the concept of Angular. It is the concept of JavaScript's class.	ngOnInit is the second stage of Angular component life cycle hook whenever is called when angular is done which creating the component.
Constructor is best place to add all dependencies	ngOnInit function which guarantees you that the component has already being created.
Constructor is automatically called at the time of creating object of the class	Invoked by Angular when component is initialized
Used for Injecting dependencies	Actual business logic performed here

Rutas

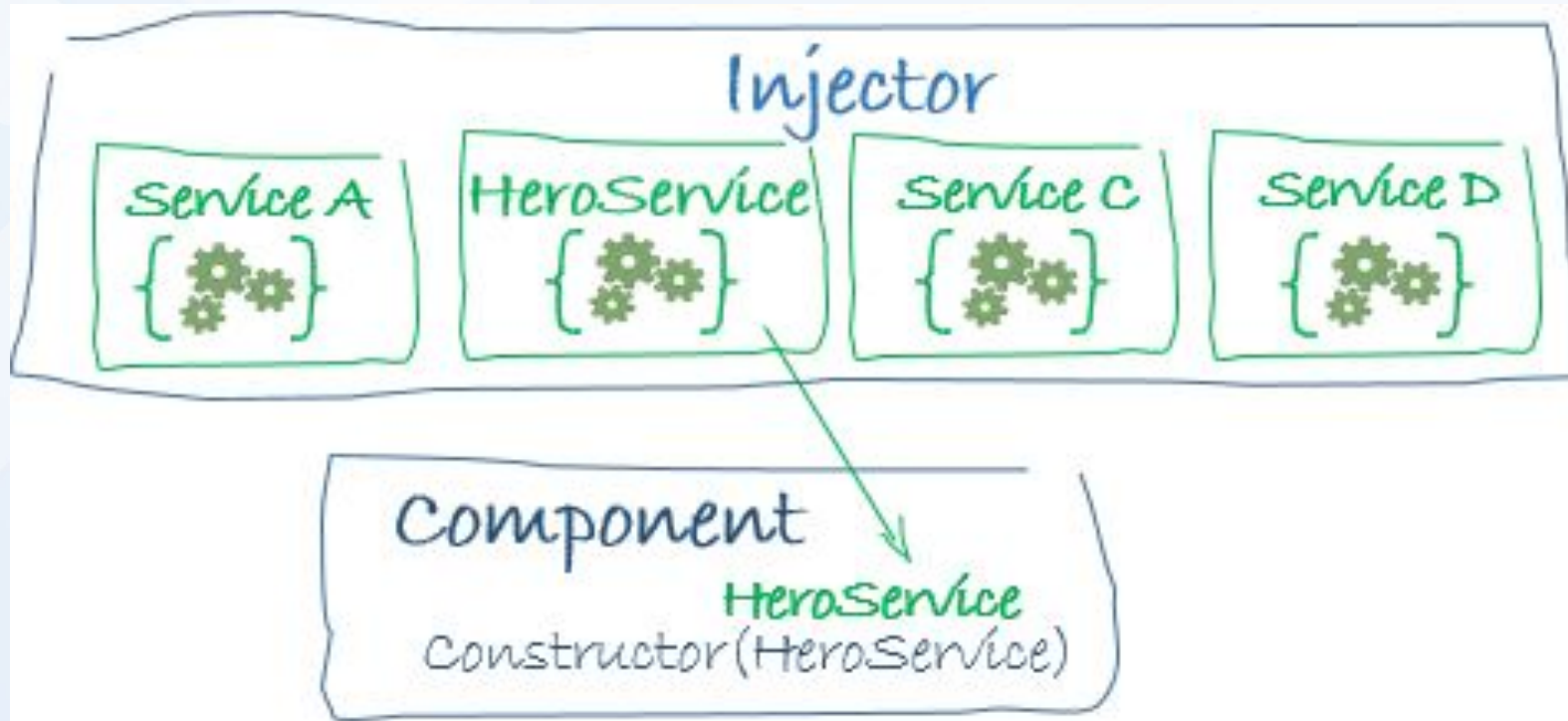
Las Rutas le indican al enrutador qué vista mostrar cuando un usuario hace clic en un enlace o pega una URL en la barra de direcciones del navegador.



Servicios

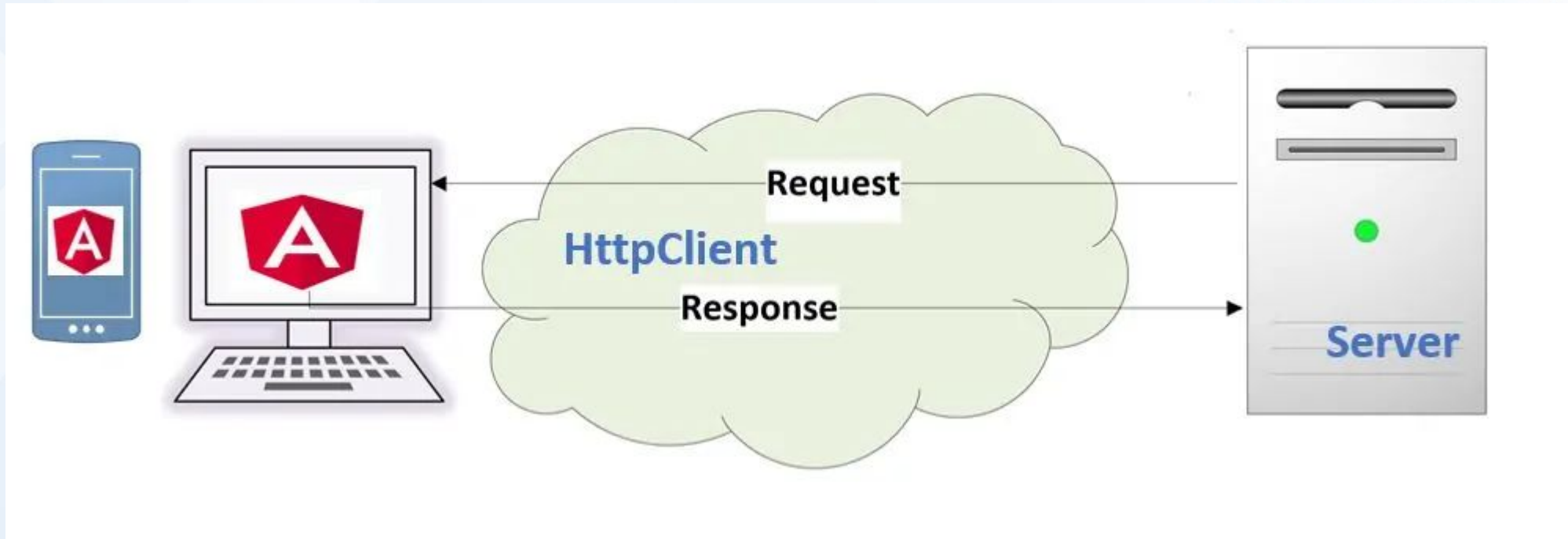
Son clases que se encargan de acceder a los datos para entregarlos a los componentes. Lo bueno de esto es que se puede utilizar un servicio para distintos componentes.

Comando: `ng generate service nombreServicio`



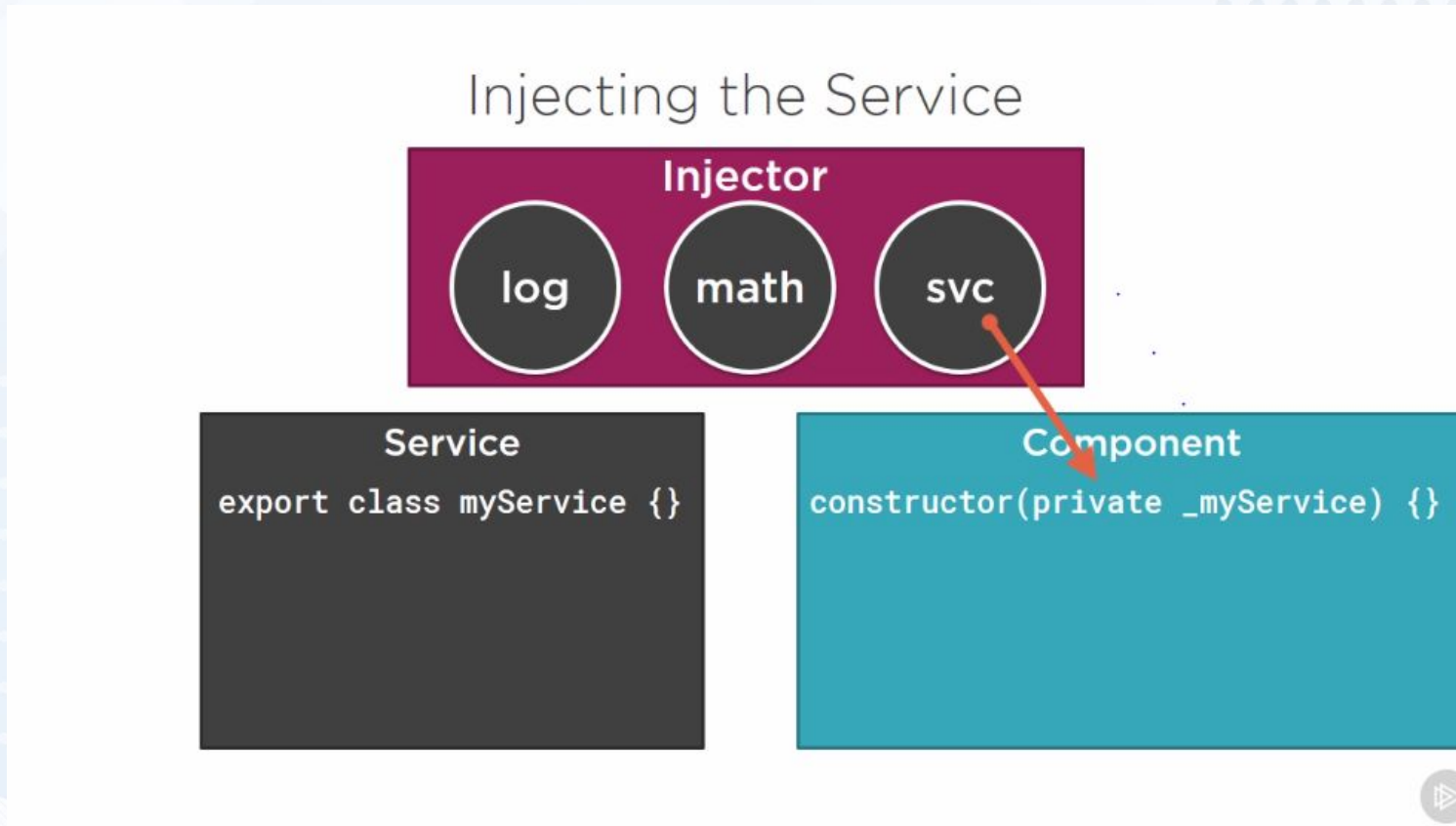
HttpClient

Es un cliente con los métodos REST habituales, que está basado en Observables. Básicamente es lo que se utiliza en angular para hacer llamadas a una API REST y obtener resultados de la misma.



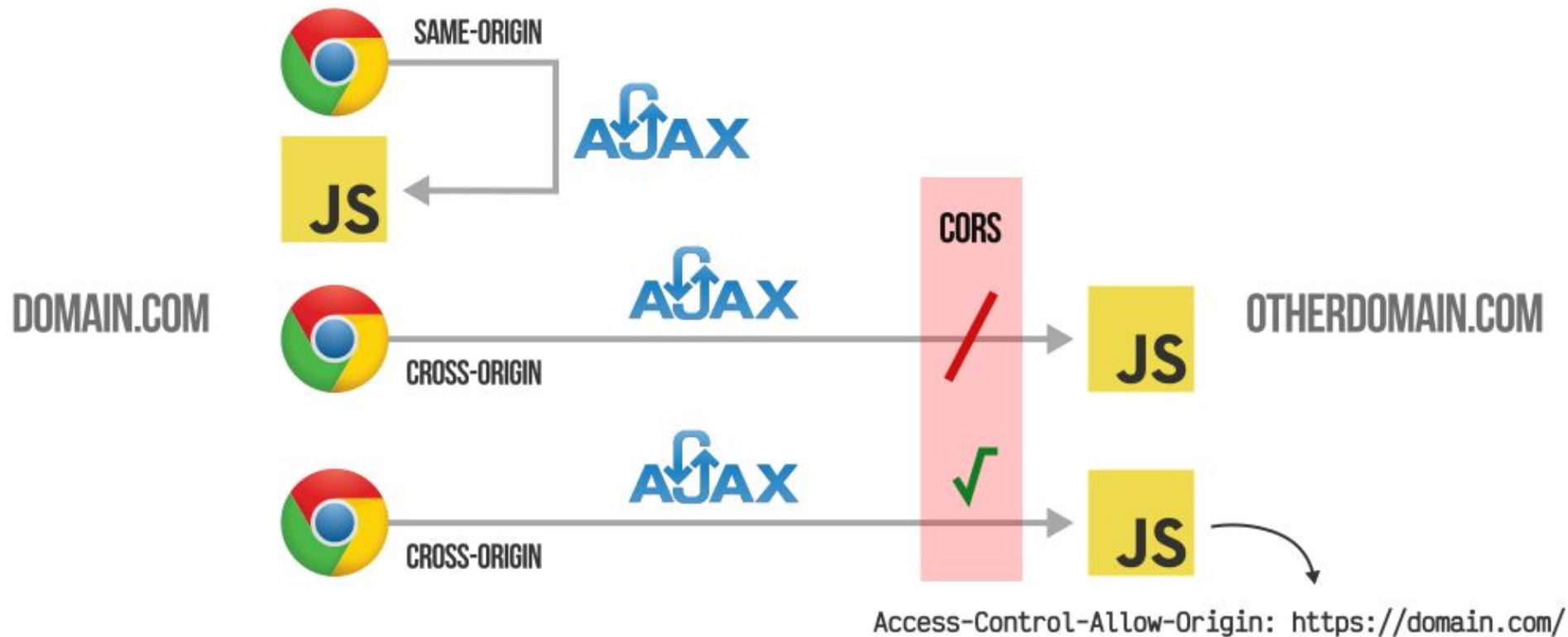
Inyección de dependencias

Es un patrón de diseño orientado a objetos, en el que se suministran objetos a una clase en lugar de ser la propia clase la que cree dichos objetos



Cors

Cross Origin (origen cruzado) es la palabra que se utiliza para denominar el tipo de peticiones que se realizan a un dominio diferente del dominio de origen desde donde se realiza la petición. De esta forma, por una parte tenemos las peticiones de origen cruzado (cross-origin) y las peticiones del mismo origen (same-origin).



Directivas

Son, esencialmente, instrucciones para manipular el DOM. Extienden la funcionalidad del HTML usando para ello una nueva sintaxis. Con ella podemos usar lógica que será ejecutada en el DOM (Document Object Model).



Directivas de componente

Una directiva de componente es aquella que administra una región de HTML de una manera nativa como un elemento HTML. Técnicamente es una directiva con un template.

Directivas estructurales

Alteran la estructura del DOM, agregando, eliminando y manipulando los elementos host a los que están unidos.

- ngIf:



```
<section class="img-section" *ngIf="selectedHero.name">
  <h2>{{selectedHero.name}}</h2>
  <img [src]="selectedHero.img" [alt]="selectedHero.name" />
</section>
```

Directivas estructurales

- ngFor:

```
<ul class="heroes">
  <li *ngFor="let hero of superHeroes" (click)="onSelect(hero)">
    <span class="badge">{{hero.id}}</span> {{hero.name}}
  </li>
</ul>
```

Directivas estructurales

- ngSwitch:

```
1 <section>
2   <h1>Type of Super Heroes</h1>
3   <ul *ngFor="let hero of superHeroes" [ngSwitch]="hero.type">
4     <li *ngSwitchCase="'Marvel'" class="text-opt1">{{ hero.name }} {{ hero.type }}
5   </li>
6     <li *ngSwitchCase="'DC'" class="text-op2">{{ hero.name }} {{ hero.type }}
7   </li>
8     <li *ngSwitchDefault class="text-op3">{{ hero.name }} {{ hero.type }}
9   </li>
10  </ul>
11 </section>
```


Directivas de atributo

ngModel: Implementa binding

ngClass: permite añadir/eliminar varias clases

ngStyle: permite asignar estilos inline

HTML

```
1 <p [ngClass]="{'rojo': flag, 'azul': !flag}">
2   Hola
3 </p>
4
5 <button (click)="cambiarFlag()">Cambiar clase</button>
```

```
<div [ngStyle]="{'background-color':person.country === 'UK' ? 'green' : 'red' }"></div>
```

HTML

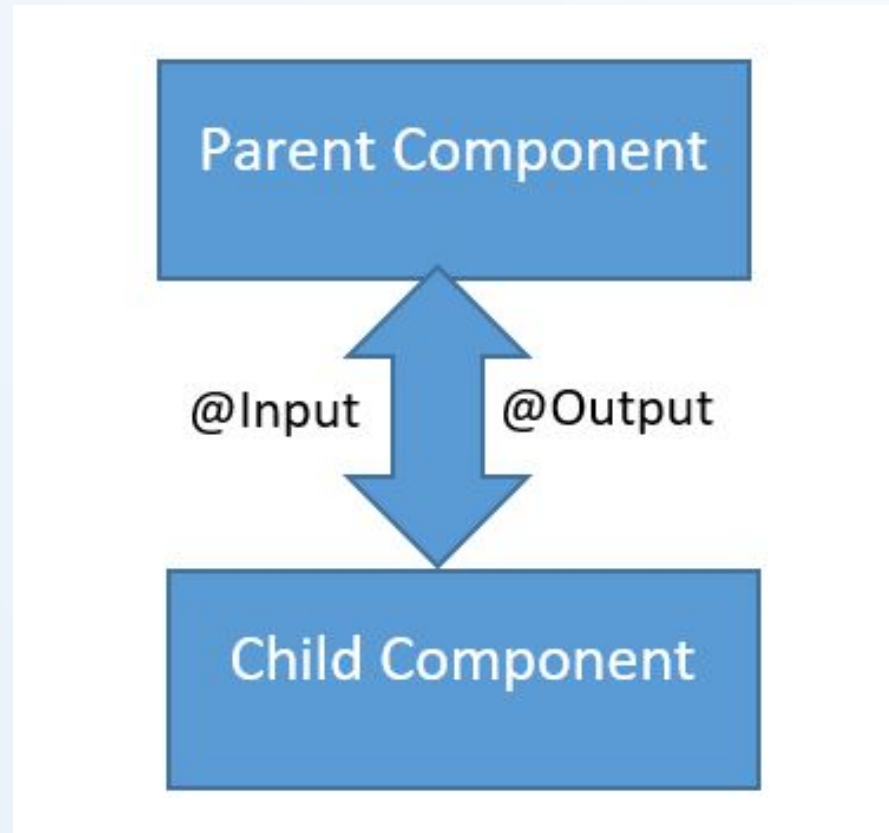
Formularios

Los formularios reactivos también se conocen como formularios controlados por plantillas. Esto significa que el contenido HTML cambia según el código en el componente.

Se necesita: `ReactiveFormsModule`

Comunicación entre componentes

- @Input: Pasar datos de padre a hijo
- @Output: Pasar datos de hijo a padre



Comunicación entre componentes

- Datos por el router
- LocalStorage - SessionStorage

