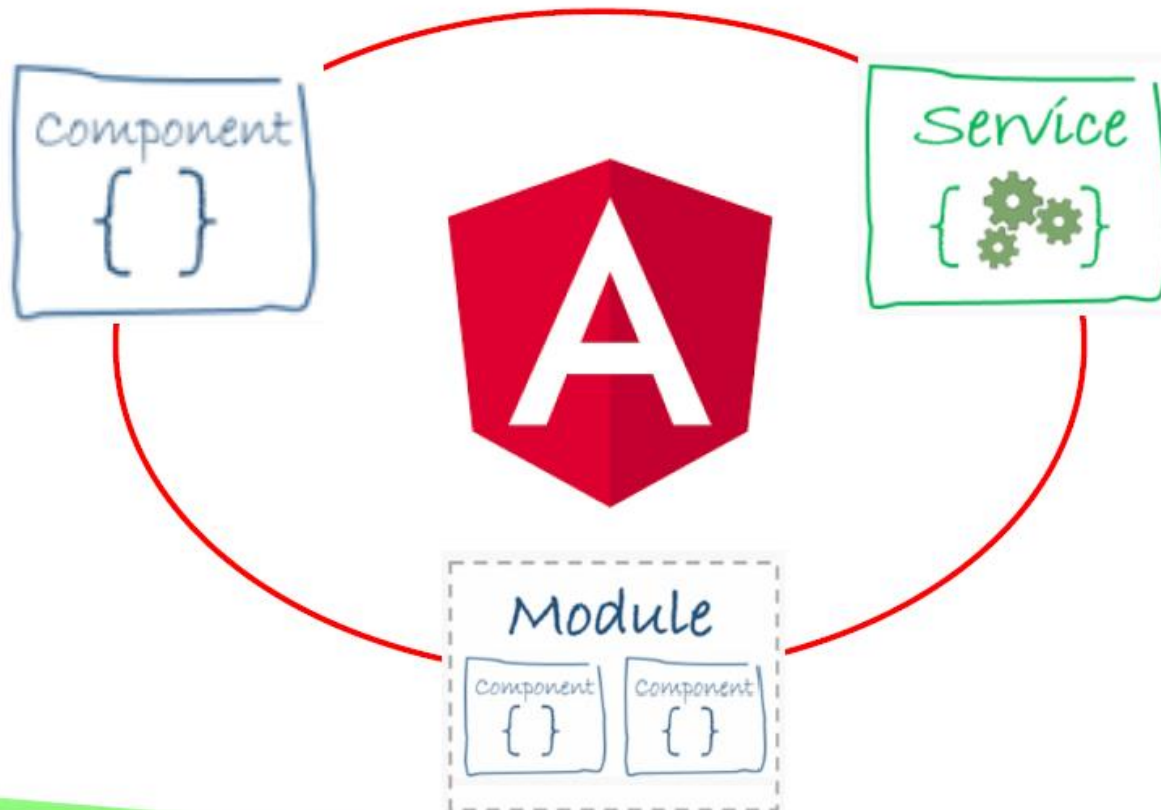


Arquitectura de una app Angular

Arquitectura de una app Angular

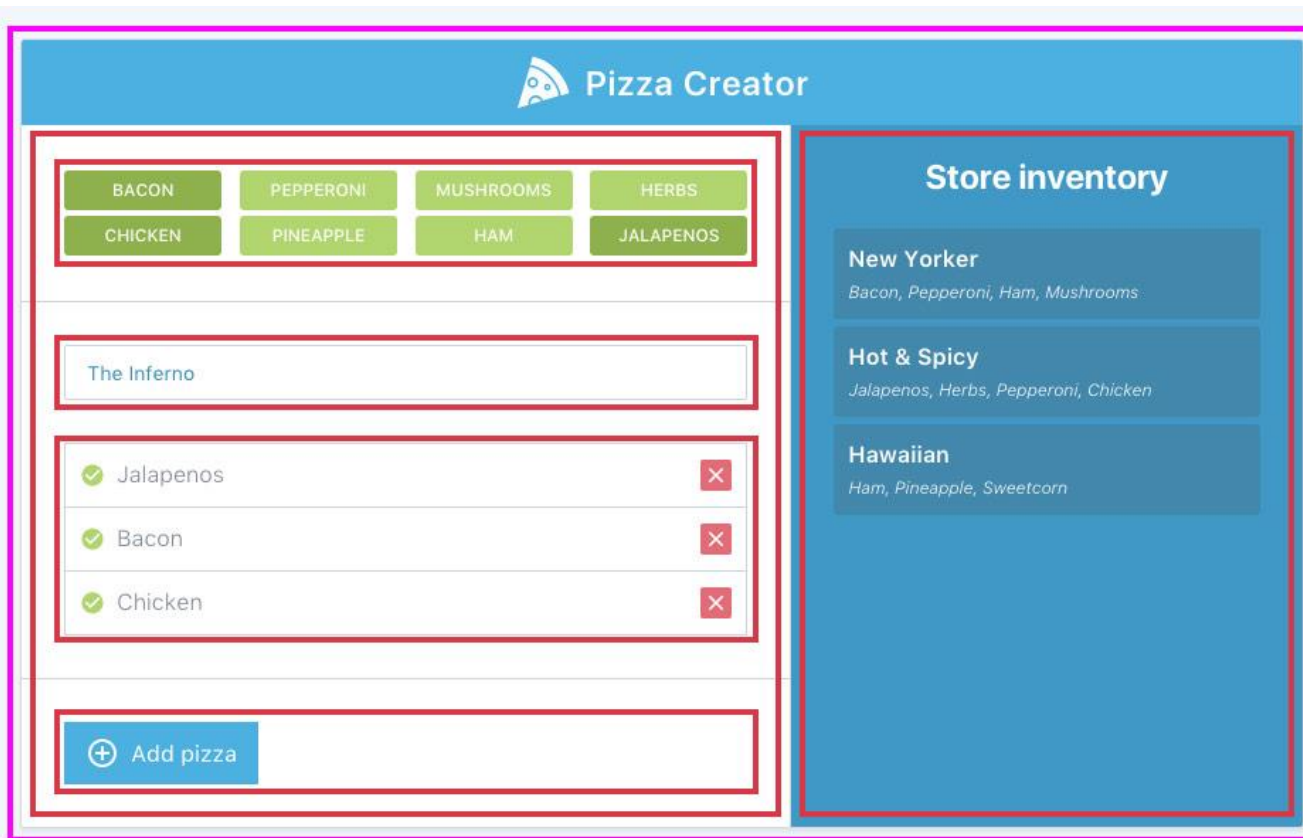
- Componentes + Servicios + ModulosEjemplo:



Arquitectura de una app Angular

Componentes

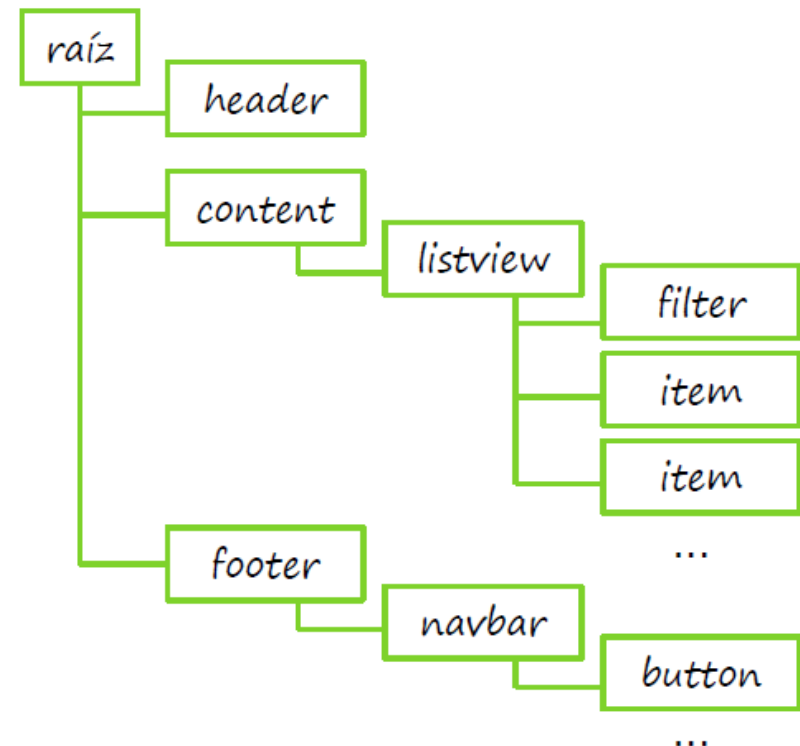
- Una aplicación Angular está basada en un conjunto de **COMPONENTES**.
- El usuario interactúa con los **COMPONENTES**.



Arquitectura de una app Angular

Componentes

- Un componente puede ser **básico** o **agregado**
 - Los agregados contienen otros componentes formando un **ÁRBOL DE COMPONENTES**
 - Tenemos un **componente RAÍZ** que carga otros componentes que, a su vez, cargan otros componentes, ... y así sucesivamente.



Arquitectura de una app Angular

Componentes

➤ La mayor parte de nuestro trabajo, por tanto, consistirá en:

1. Definir los componentes

2. Trabajar las interacciones entre ellos:

- ✓ Un componente **padre pueda pasarle datos al componente hijo**
- ✓ Un componente **hijo pueda notificar situaciones, eventos e incluso datos al componente padre.**

Arquitectura de una app Angular

Componentes

➤ Definen:

1. Interfaz gráfica (UI)
2. Interacción del usuario con la interfaz gráfica

➤ Y los datos de mi app,

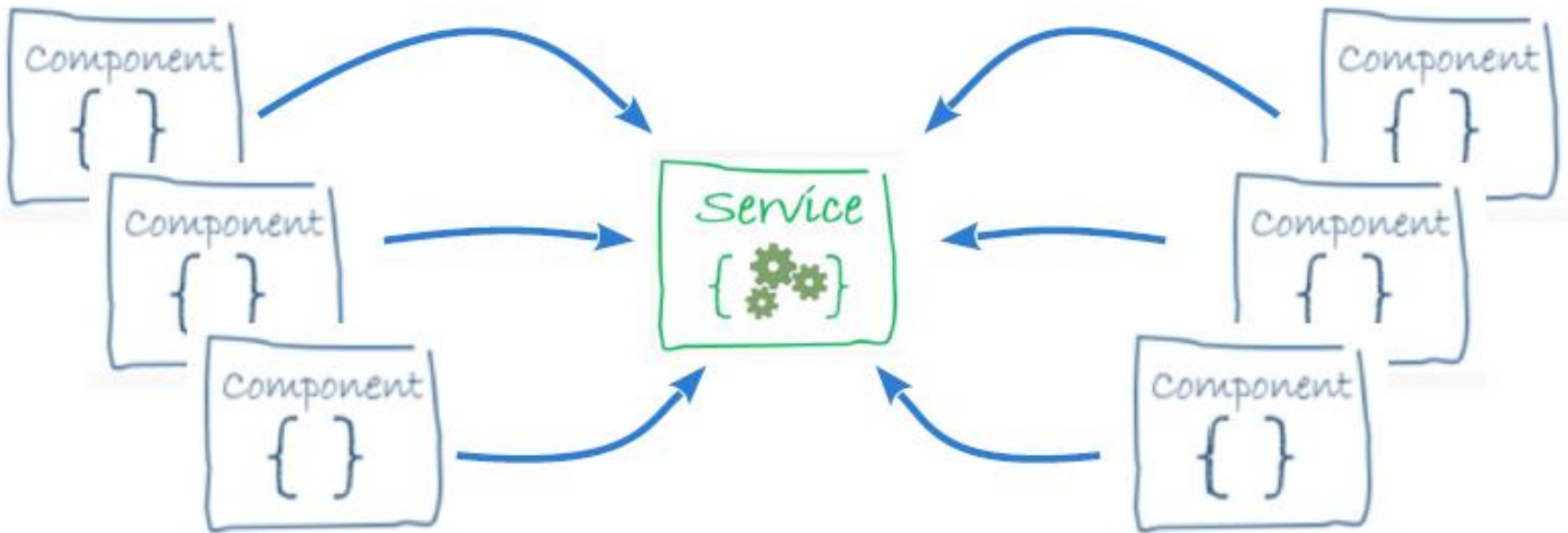
¿dónde están? ¿la lógica de negocio?

LOS SERVICIOS

Arquitectura de una app Angular

Servicios

- Contienen la lógica de la aplicación y/o los datos (depende de la app)
- Son consumidos por otros componentes/servicios
- *Lo que no es interfaz de usuario = SERVICIO*



- Los **USUARIOS** interactúan con los **COMPONENTES** y estos notifican a los **SERVICIOS** para que modifiquen los **DATOS**

Arquitectura de una app Angular

Módulos

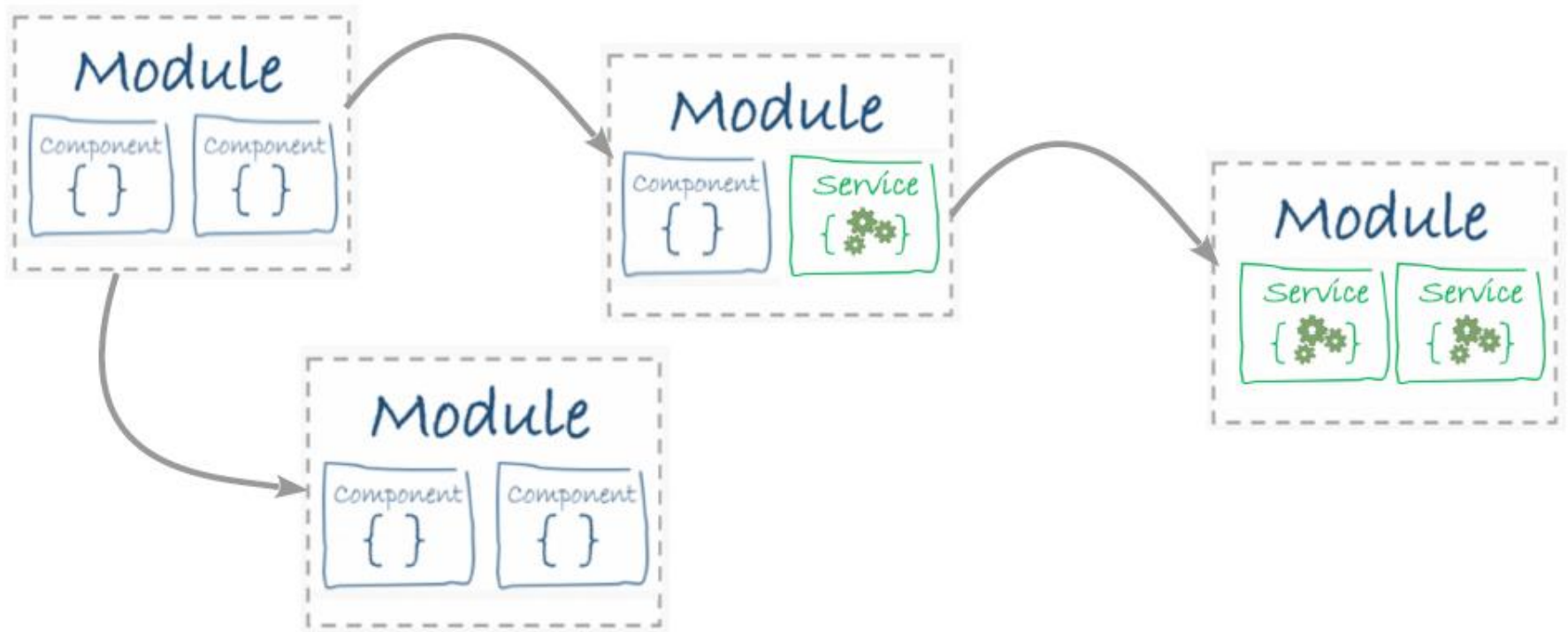
- Componentes y servicios se **ORGANIZAN EN MÓDULOS**
- Angular implementa **su propio sistema de módulos**
 - **TOTALMENTE DISTINTO** del sistema de módulos visto en **TYPESCRIPT**
- Cuando las apps Angular se complican, se organizan los componentes y servicios en **ÁREAS FUNCIONALES = MÓDULOS** o **LIBRERÍAS** (*en Angular*)
 - ✓ *Algunas librerías/módulos dependen de otras*
 - ✓ *Se crean **dependencias** entre librerías/módulos*
- Todo **sistema de gestión de módulos** necesita:
 - ✓ *Mecanismo para exportar “**cosas**”*
 - ✓ *Mecanismo para importar “**cosas**”*
- *Veremos más adelante cómo se exporta y importa en los módulos de Angular*

Arquitectura de una app Angular

Módulos

- Por defecto, **SIEMPRE** hay un **MÓDULO RAÍZ**

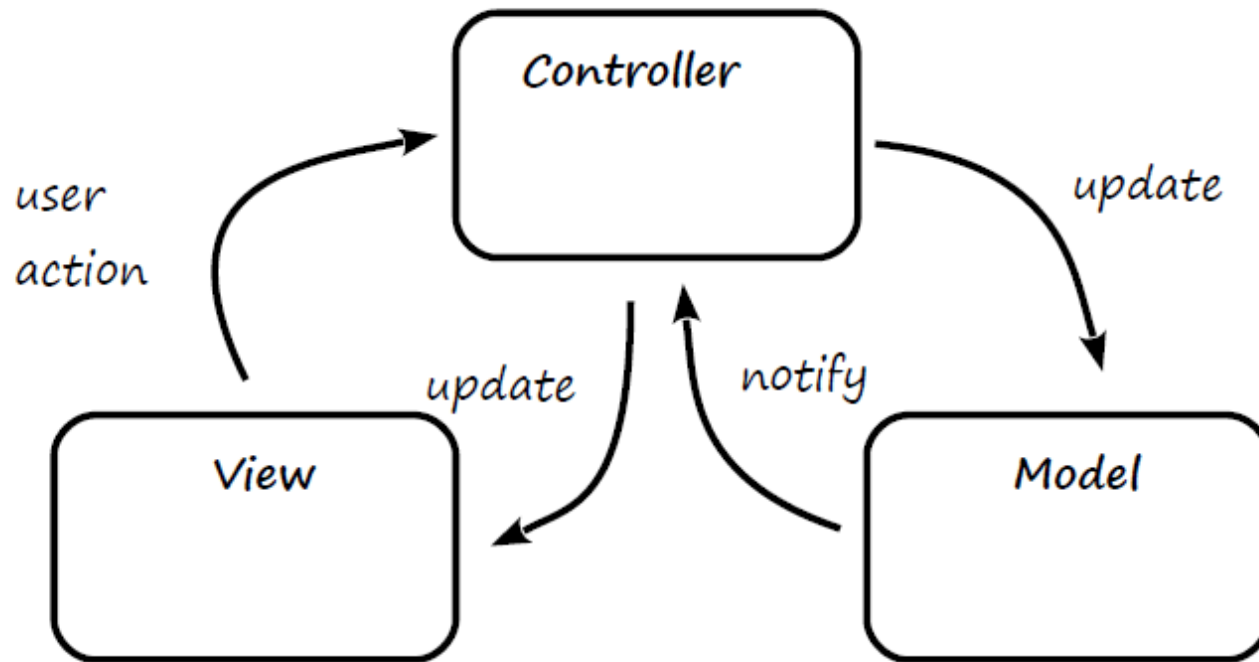
AppModule (fichero: *app.module.ts*)



Arquitectura de una app Angular

Patrón MODELO–VISTA–CONTROLADOR

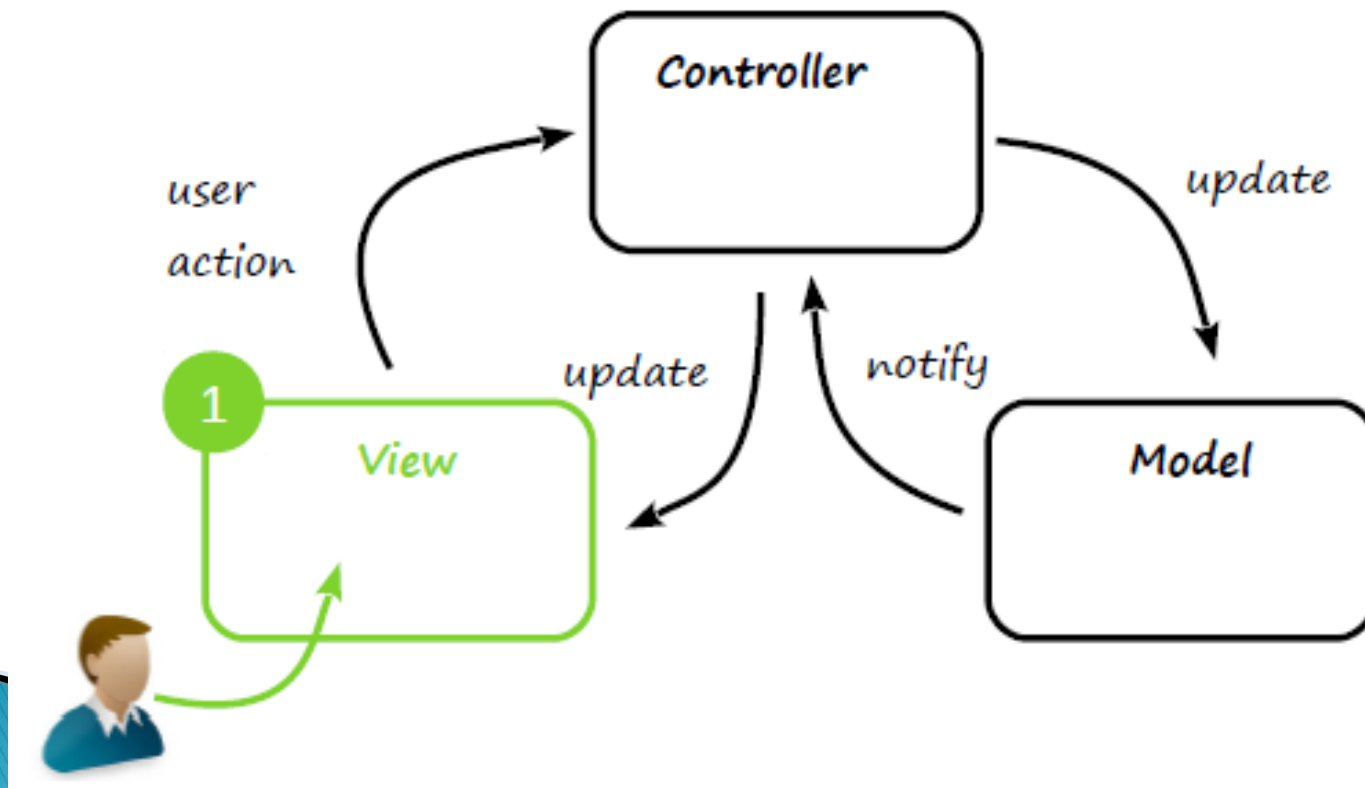
- Componentes y servicios contribuyen a implementar el patrón MVC (Model – View –Controller)



Arquitectura de una app Angular

Patrón MODELO–VISTA–CONTROLADOR

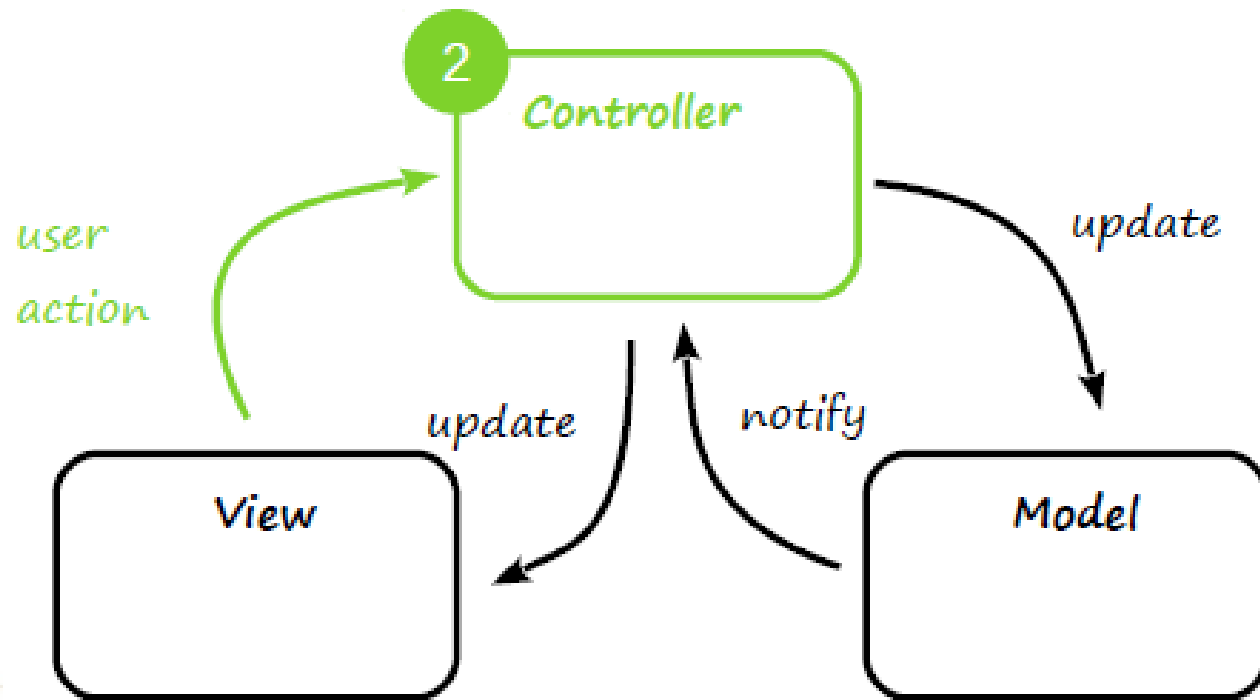
- Componentes y servicios contribuyen a implementar el patrón MVC (Model – View –Controller)



Arquitectura de una app Angular

Patrón MODELO–VISTA–CONTROLADOR

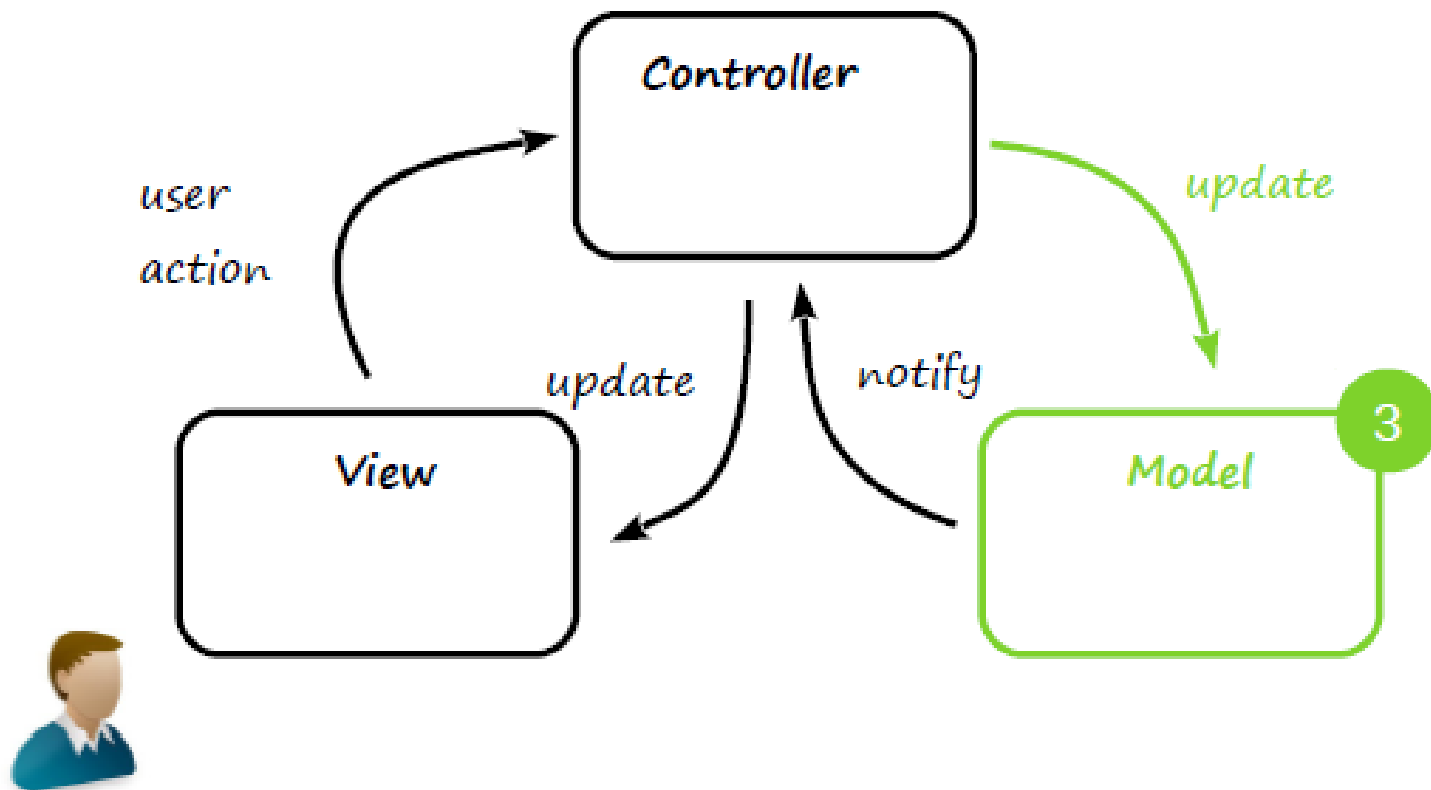
- Componentes y servicios contribuyen a implementar el patrón MVC (Model – View –Controller)



Arquitectura de una app Angular

Patrón MODELO–VISTA–CONTROLADOR

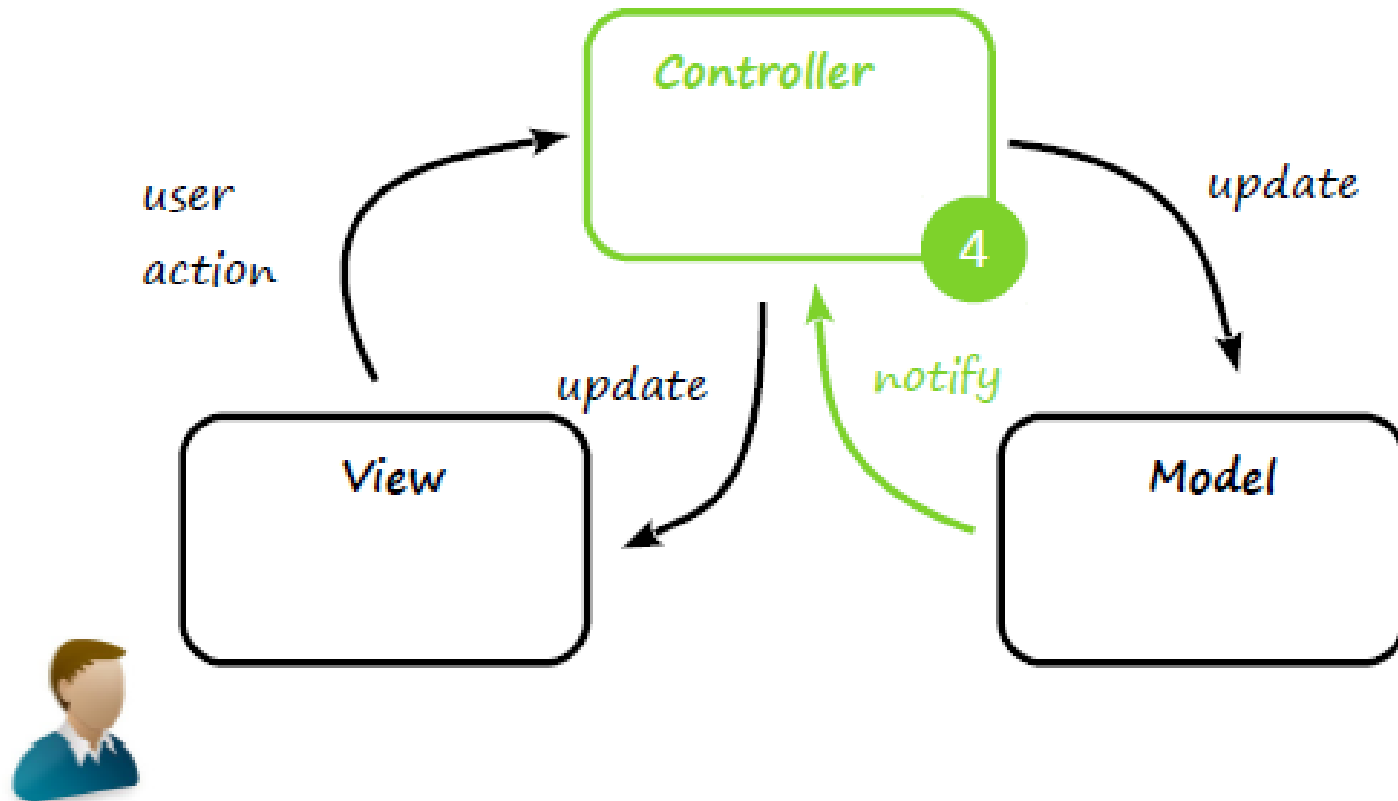
- Componentes y servicios contribuyen a implementar el patrón MVC (Model – View –Controller)



Arquitectura de una app Angular

Patrón MODELO–VISTA–CONTROLADOR

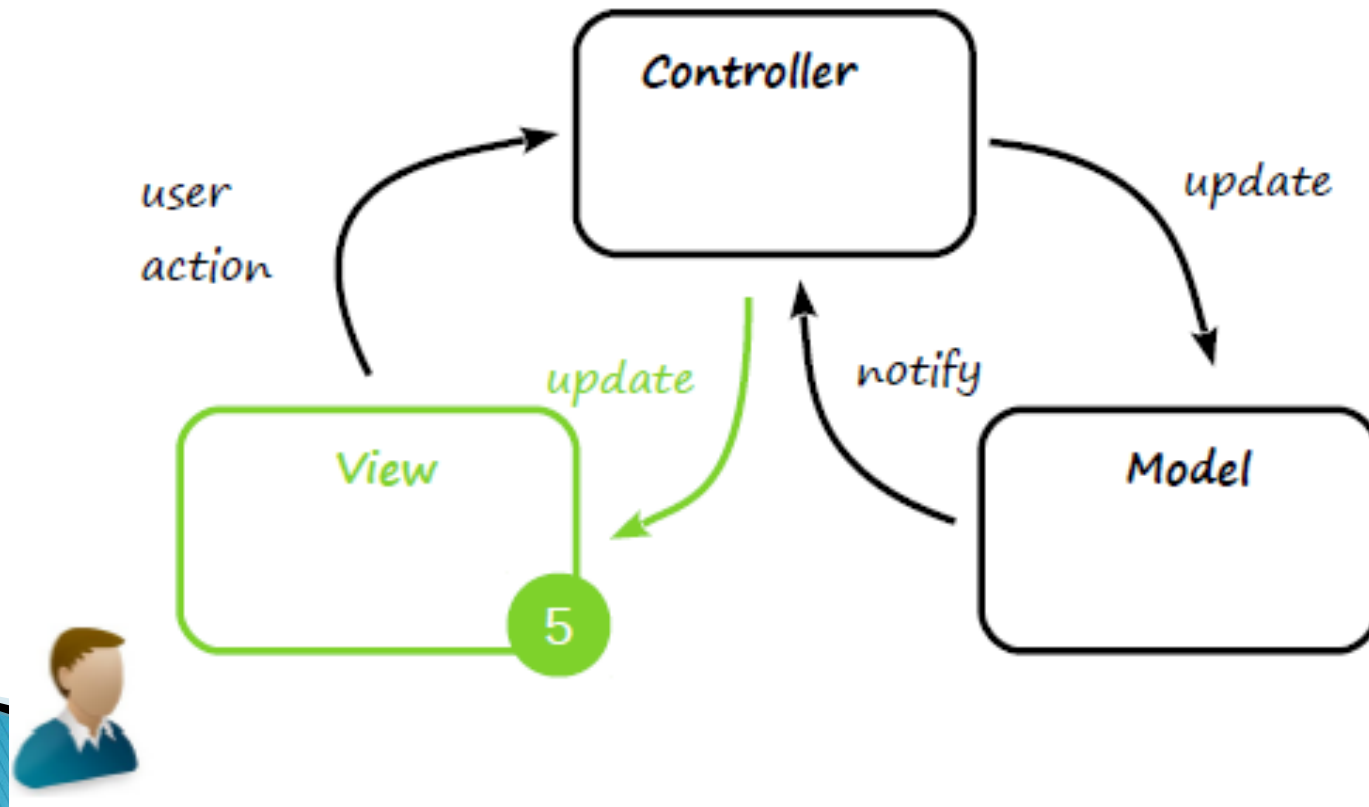
- Componentes y servicios contribuyen a implementar el patrón MVC (Model – View –Controller)



Arquitectura de una app Angular

Patrón MODELO-VISTA-CONTROLADOR

- No hay CONEXIÓN DIRECTA ENTRE MODELO y VISTA



Arquitectura de una app Angular

Patrón MODELO–VISTA–CONTROLADOR

- **Componente** = Vista + Controlador
- **Servicio** = Modelo (*datos y reglas de negocio*)

