

# ANGULAR

JHON ALEXANDER LÓPEZ BOHÓRQUEZ  
IBM: Associate Developer



# ÍNDICE

- Información general
- Configuración de ambiente
- Creación de proyecto
- Implementación de proyecto “My online store”
- Firebase
- Compilación del proyecto

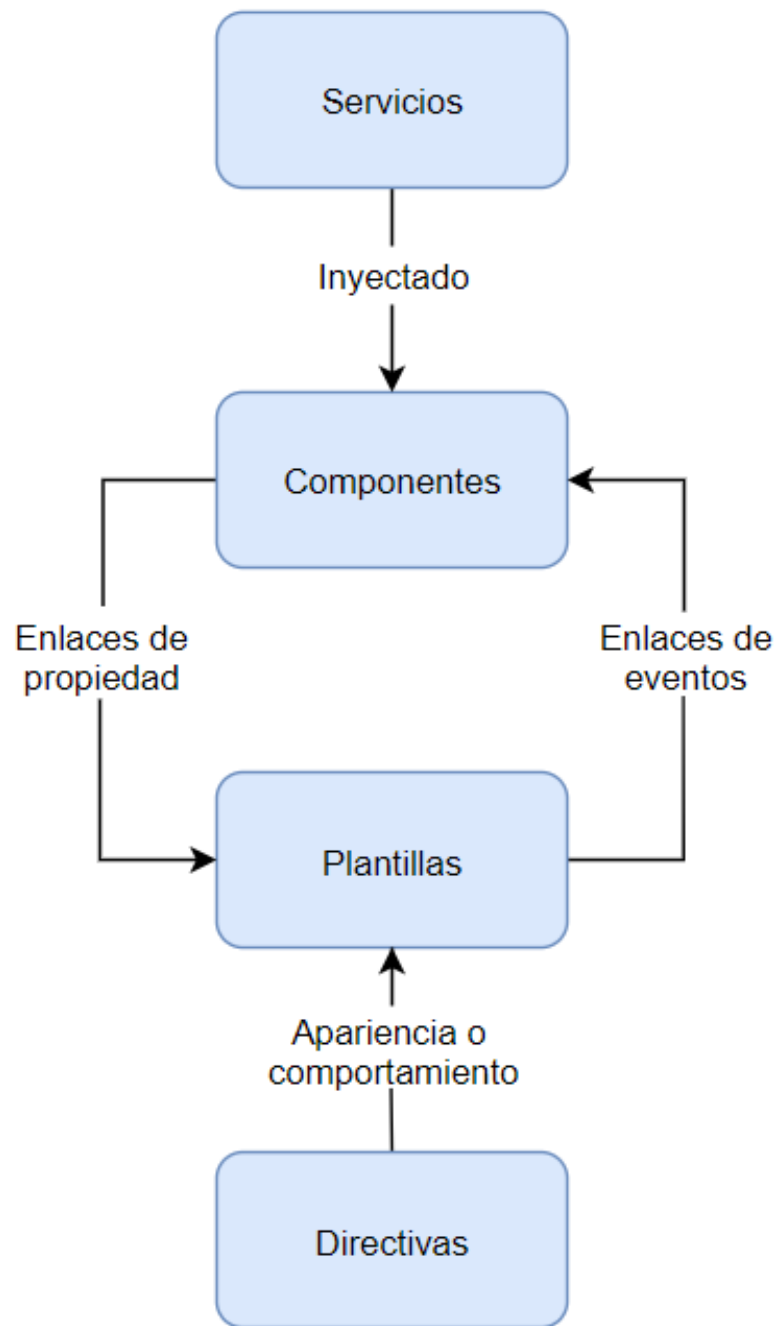
# Angular

- Es un framework para la creación de aplicaciones web de código abierto creado por Google. Usa TypeScript como lenguaje de programación.



# Angular

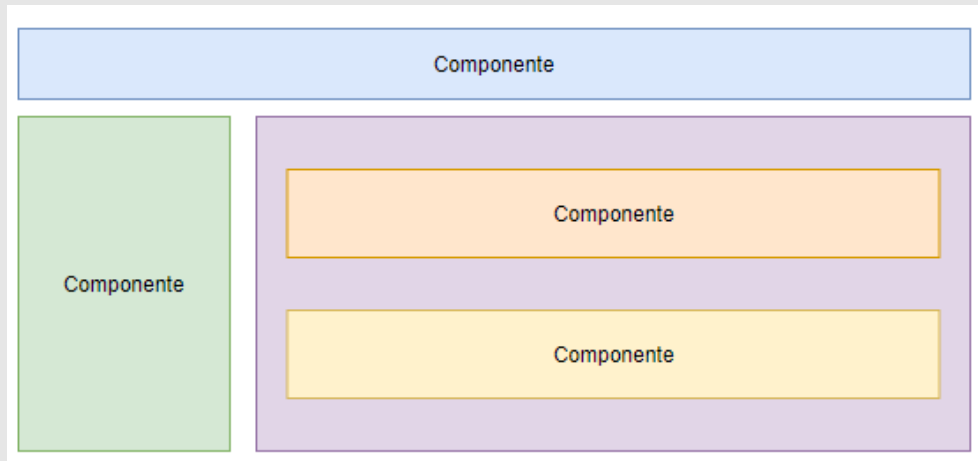
- Al ser un framework, Angular ofrece más funcionalidades que una simple biblioteca. Con otros software similares, lo más común es tener que utilizar varias bibliotecas de terceros.
- Angular cuenta con una estructura predefinida y desde el primer momento regulariza cómo organizar el código y la arquitectura de la aplicación.
- Angular usa TypeScript como lenguaje de desarrollo, que tiene permite un mejor entendimiento del código por su consistencia en la documentación, mejora el mantenimiento de las aplicaciones y evita confusiones y sobrecarga en la toma de decisiones al empezar con un proyecto Angular.
- Angular adopta un estándar de los componentes web, lo que permite un desarrollo de aplicaciones mucho más ágil.
- Angular almacenan por separado el código de la interfaz de usuario y el de la lógica de negocio. Otros frameworks como React, por ejemplo, mezclan en un mismo archivo todo el código, lo cual dificulta el uso de herramientas estándar de desarrollo.



# Angular

Arquitectura básica de angular

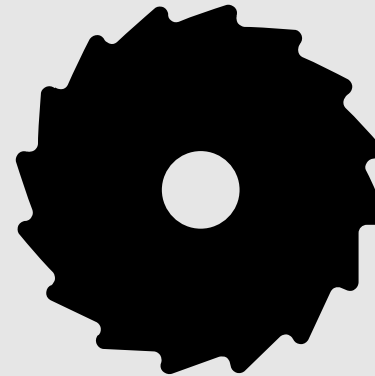
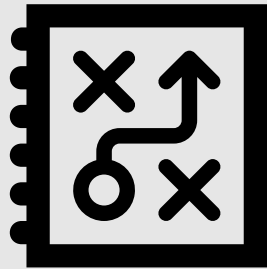
# Componentes



- Los componentes son las vistas de la aplicación Angular, estos se encargan de controlar qué, cuándo y cómo deben mostrarse los diferentes elementos en la pantalla.

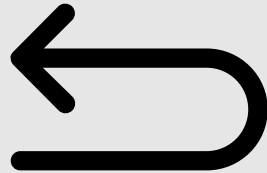
# Servicios

- Los servicios son clases con un designio único enfocado a la lógica que pueda llegar a tener el componente. Este debe ser limitado y bien definido, es decir, un servicio no debe contener la lógica de todas las aplicaciones.



# Directivas

- Las directivas se encargan de la interacción con las plantillas y sus respectivos componentes directos. Las directivas interactúan con las plantillas a través de enlaces de propiedad y eventos. Las directivas pueden cambiar estilos o dar formatos a elementos de textos que se ingresan a través de eventos.





# Pipes

- Los pipes son clases que se enfocan en la transformación de datos, por tanto, convierte cualquier entrada en una salida deseada. A diferencia de las directivas, los pipes implican el uso de una interfaz proporcionada por Angular, *PipeTransformy*. Esta se encarga de definir un método único donde ocurrirá la transformación de los datos.
- Ejemplo: Transformar "763000" a "\$763.000".

