

# Libro Angular 2

Raúl López

Published  
with GitBook



# Tabla de contenido

Introducción	0
¿Por qué Angular 2?	1
Prerequisitos	2
Entorno	3
Dependencias	4
Arquitectura	5
TypeScript	6
Components	7
Lifecycle	7.1
Styles	7.2
Templates	8
Data Binding	9
Directives	10
Pipes	11
Routing	12
Resolve	12.1
ui-router	12.2
Http	13
Services	14
Firebase	14.1
RxJS	15
Observables	15.1
Forms	16
Tests	17
TDD	17.1
BDD	17.2

# Angular 2

Este libro se basa en la recopilación de información adquirida de diferentes fuentes (libros, tutoriales, experiencia personal) sobre Angular 2. Es más un libro personal que con intención de difusión, pero si en el futuro le sirve a alguien me sentiría agradecido si se recomienda el link y se comparte.

## Contenido del libro

El contenido será en castellano con la salvedad de los *títulos, palabras que no tenga sentido traducir o conectos de Angular/TypeScript* que serán en inglés.

Esto facilita la búsqueda de información por internet. Dejando únicamente la explicación de los conceptos en castellano.

Por supuesto muchas veces dentro del contenido se mezclarán terminos en inglés con explicaciones en castellano.

**Cabe destacar que el contenido está actualmente en desarrollo y seguirá así. Quiero que sea una fuente viva y que vaya evolucionando con las nuevas versiones de Angular.**

## Notas

**De ahora en adelante cuando hablemos de Angular nos estaremos refiriendo únicamente a Angular 2.**

## Código

Los contenidos de código se identificará así:

```
// app.component.ts
import {Component} from 'angular2/core';

@Component({
  selector: 'my-app'
})
export class AppComponent {}
```

Todos los códigos que se incluyan estarán probados y se adjuntará la versión de Angular con los que se han desarrollado.

Opcionalmente si el ejemplo lo necesita se añadirá un link hacía un editor online tipo: [plunker](#), [jsbin](#), [jsfiddle](#), etc.

## Enlaces

Y los links hacia la fuente original o de ampliación de información así:

[Página oficial de Angular](#)

## Erratas

Pido disculpa de antemano por las posibles faltas de ortografía que intentaré sean la mínimas posibles. De todas formas se realizará una revisión de textos y códigos cada cierto tiempo.

Cualquier error o amplicaición de contenido lo podéis notificar a través de mi cuenta de [github](#).

## Licencia

El contenido del libro se encuentra bajo licencia [MIT](#).

# ¿Por qué Angular 2?

## Ventajas de Angular 2 con respecto a Angular 1.x

1. Contruido para ser rápido
2. Código moderno
3. API simplificada
4. Mejora la productividad

# Prerequisitos

Los conocimientos necesarios para empezar a trabajar con Angular 2 son los siguientes:

## Requerido

- JavaScript
- HTML5
- CSS3

## Ayuda

- Conceptos de Orientación a objetos
- Familiarizado con Java, C++ o C#

## No requerido

- Experiencia previa con Angular 1.x
- Experiencia previa con TypeScript

## Semilla

- <https://jakearchibald.com/2014/offline-cookbook/#cache-then-network>

# Enviorement

La instalación del entorno de trabajo se basará únicamente en sistemas linux/mac. La diferencia para usuario windows no suele ser muy disitinta.

Instalar: node.js + npm

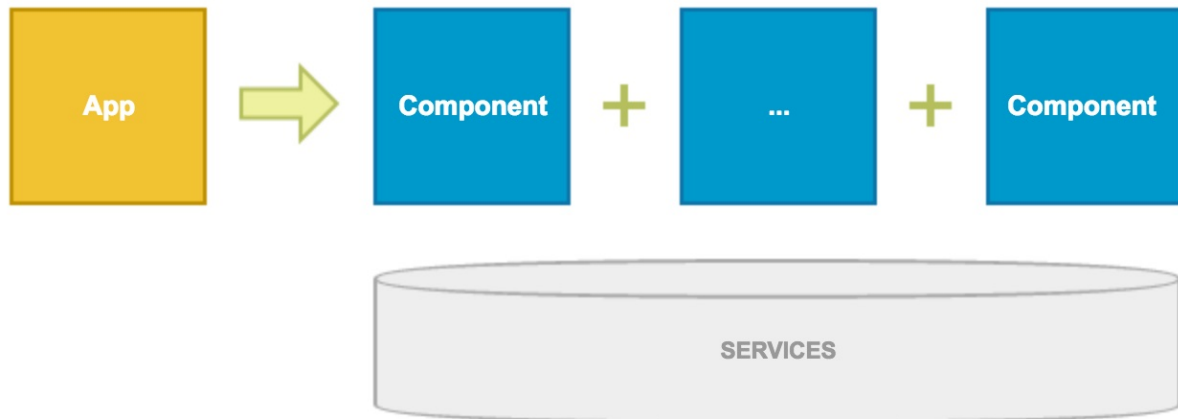
# Dependencias

Listado de depencias de un proyecto *Angular 2*.

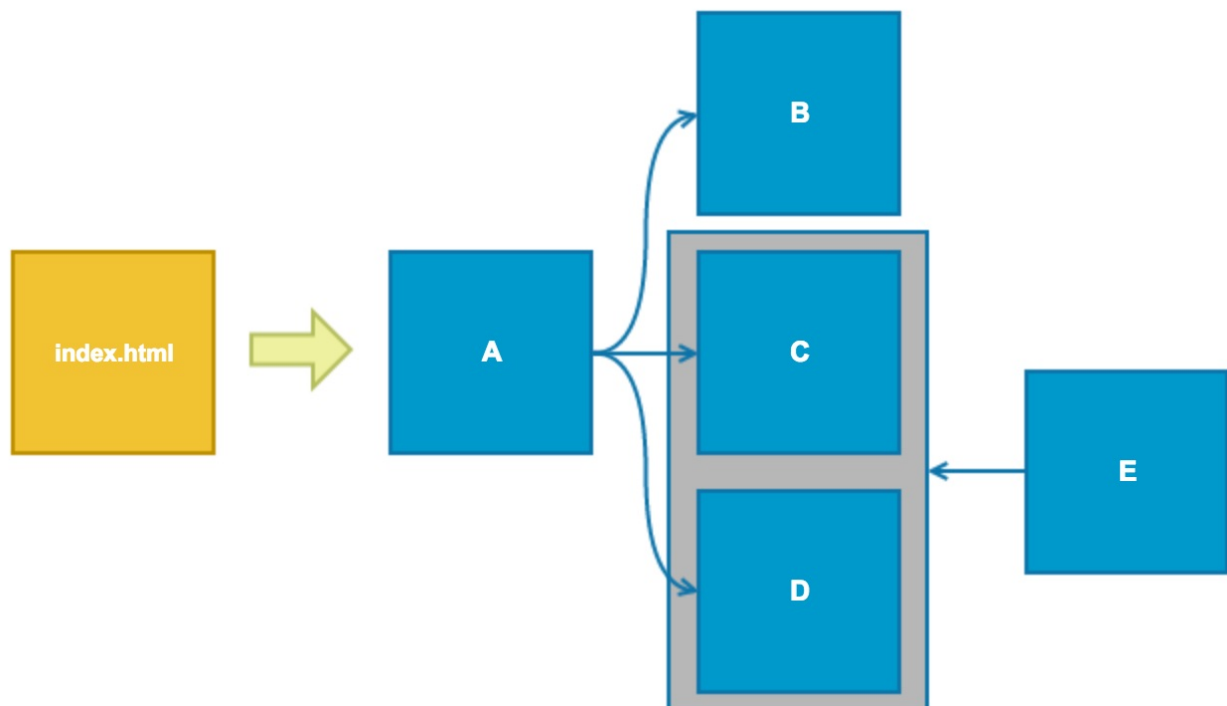


# Arquitectura

Una app de *Angular 2* está basada en *components*



Una arquitectura de ejemplo de una app con *Angular 2*



# TypeScript

## ¿Por qué TypeScript?

- Es un lenguaje Open Source
- Es un superset de JavaScript
- Transpila a JavaScript nativo
- Fuertemente tipado
- Orientación a objetos basado en clases (Java, C#, C++...)

## Editores

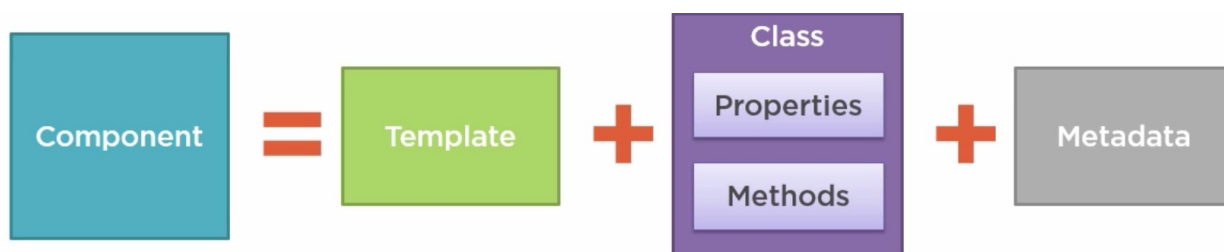
- [Visual Studio](#)
- [Visual Studio Code](#)
- [WebStorm](#)
- [Atom](#) + [package](#)
- [Eclipse](#)

## Características:

- Interfaces
- Inheritance
- Modules

# Components

**Angular 2 está basado en componentes. Su estructura básica es la siguiente**



# Lifecycle

## El ciclo de vida de un componente



# Styles

¿Cómo añadir estilos CSS a lo *template* de un *component*?

# Templates

- `template`
- `templateUrl`

# Data Binding

Formas de renderizado y comunicación entre un *component* y su *template*.

# Directives

Básicamente son *components* sin *template*.



# Pipes

En **Angular 1.x** eran conocidos por *filters*.

# Routing

¿Cómo se realiza la navegación en una app de *Angular 2*?

# Resolve

¿Cómo realizar navegación entre *templates* con el los *datos* previamente cargados?

# ui-router

- ¿Tenemos disponible *ui-router* en *Angular 2*?
- ¿Tiene sentido?

## Existe una versión aplha

<https://github.com/angular-ui/ui-router/tags>

# Http

Recuperando datos a través de Http.

# Services

Creando servicios para organizar las llamadas *Http* y como realizar la inyección de dichos servicios.

# Firestore

¿Cómo integrar **Firestore** con *Angular 2*?

# RxJS



# Observables

¿Cómo se comunican los *components*?

# Forms

En el desarrollo web los formularios son complicados.

¿Cómo funcionan los formularios en Angular?

# Tests

Los tests hoy en día son imprescindibles para el desarrollo de software.

Cómo testear una app Angular con:

- Karma
- Jasmine

# TDD

# BDD