

# { Angular }

Framework para la creación de Aplicaciones Web y Apps

Formador: Ezequiel Llarena Borges

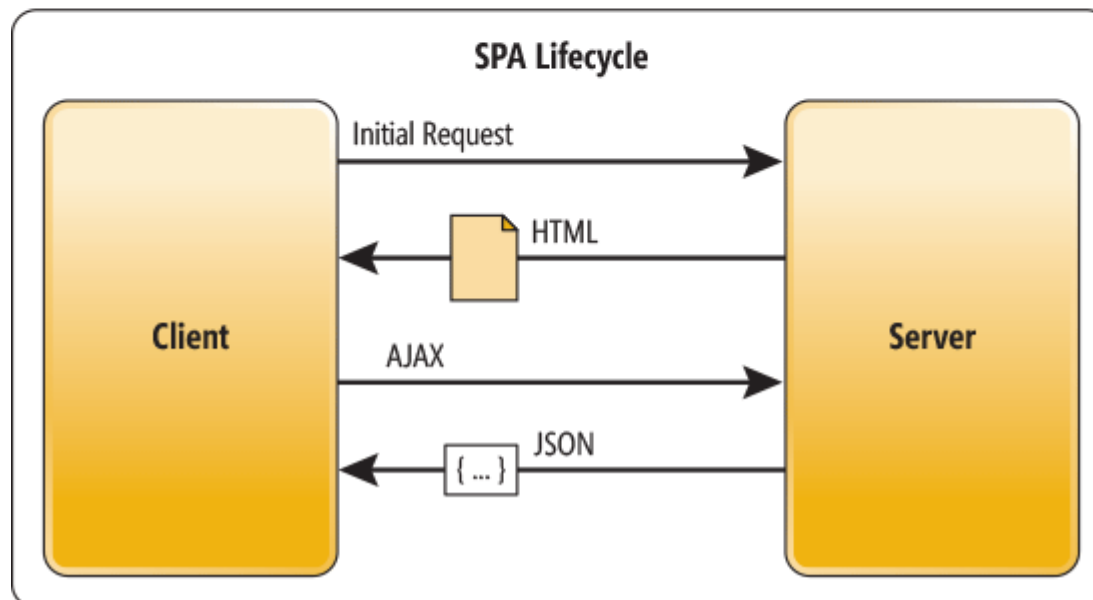
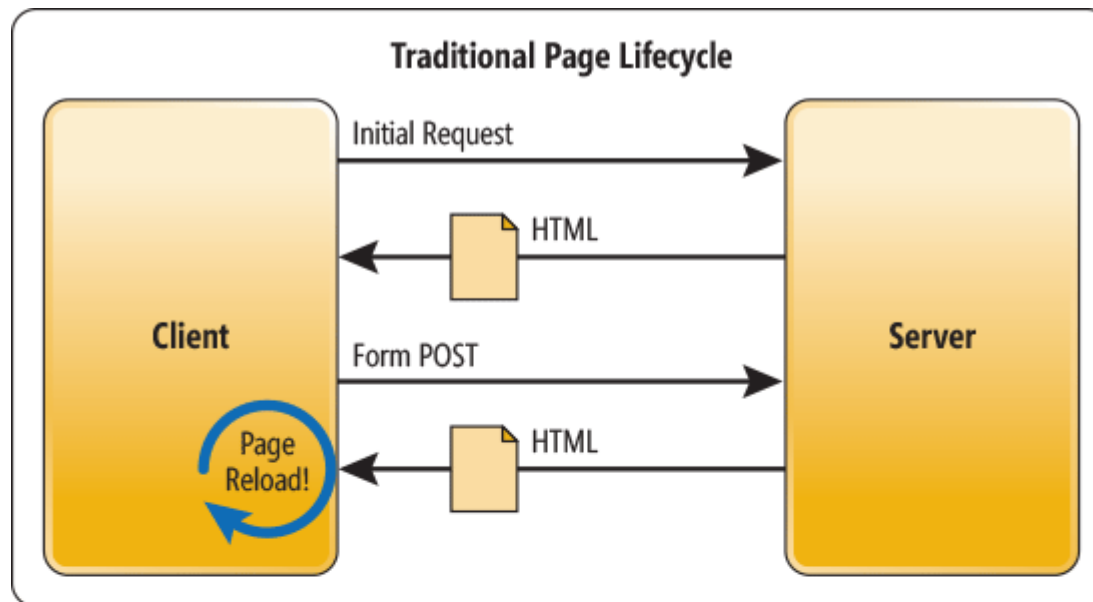
# { Angular }

Introducción, Instalación y Arquitectura de un proyecto Angular 4

Formador: Ezequiel Llarena Borges

## Características

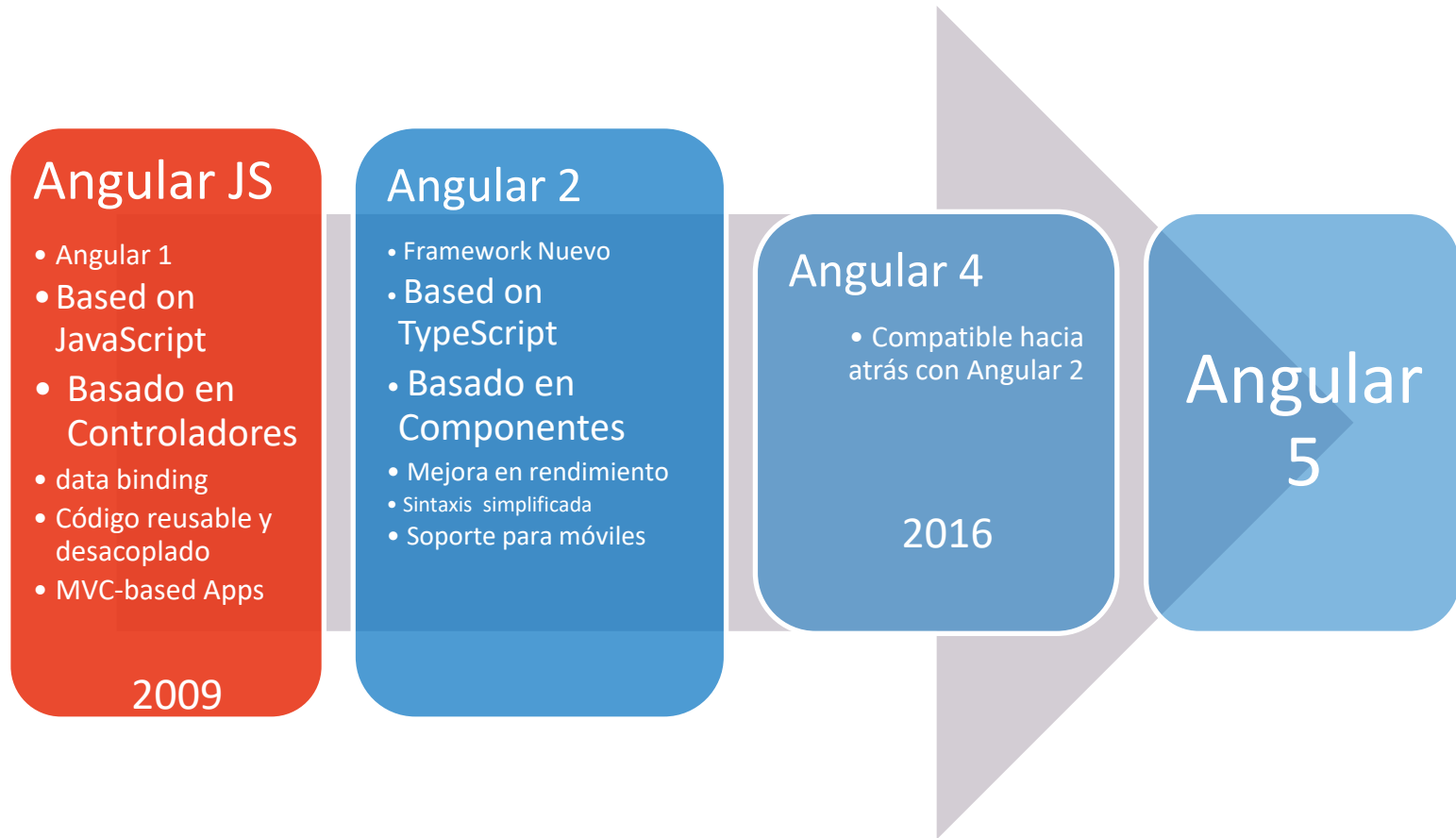
- Framework desarrollado y soportado por **Google**
- Proyectos **SPA** (Single-Page Application): carga de datos **asíncrona**
- Basado en el lenguaje **TypeScript** (desarrollado por Microsoft)
- TypeScript **basado en JavaScript** mejorado (funciones tipado y POO)



## Funcionamiento

- **Componentes** (controlador en Angular JS) que representan la vista/presentación de la webapp
- Componentes soportados por **clases**
- Utiliza **TypeScript** para lógica de las aplicaciones
- Sintaxis y desarrollo incorpora mejoras y características de **ECMAScript 6** (estandarización de JavaScript)
- Angular **transpila** el código TypeScript directamente a JavaScript ya que ECMAScript 6 aún no es soportado por los navegadores web
- Permite **trabajar en tiempo real** sobre la aplicación (podemos levantarla mediante un servidor disponible con Node JS)

# Angular JS vs Angular



## Herramientas de desarrollo

- **Node JS** (Entorno de ejecución para JavaScript)
- **NPM** (Gestor paquetes; *el Maven de Angular*)
- **Angular CLI** (Agiliza y facilita instalación de Angular + Inicio proyecto Angular)
- **Sublime Text** (Entorno de desarrollo)
- **Angular** (Framework)

## Pasos

- Bases fundamentales de Angular
- Instalar Node JS
- Instalar Angular con *quickstart* de su repositorio GitHub
- “Hola Mundo” de Angular
- Crear proyecto con Angular CLI
- Componentes (AppComponent)
- Creación de componentes
- Trabajar con múltiples componentes (Binding y Selector Tag)



## Comandos básicos Angular CLI

- `npm -v / npm --version`
- `node -v / node --version`
- `tsc -v`
- `ng new <nombre_proyecto>` Crear aplicación nueva
- `ng serve [--port <puerto>|--open]` Lanzar aplicación en el servidor

# API Angular

<https://angular.io/api>

# { Angular }

## Data Binding

Formador: Ezequiel Llarena Borges

## Data Binding

### One Way Binding

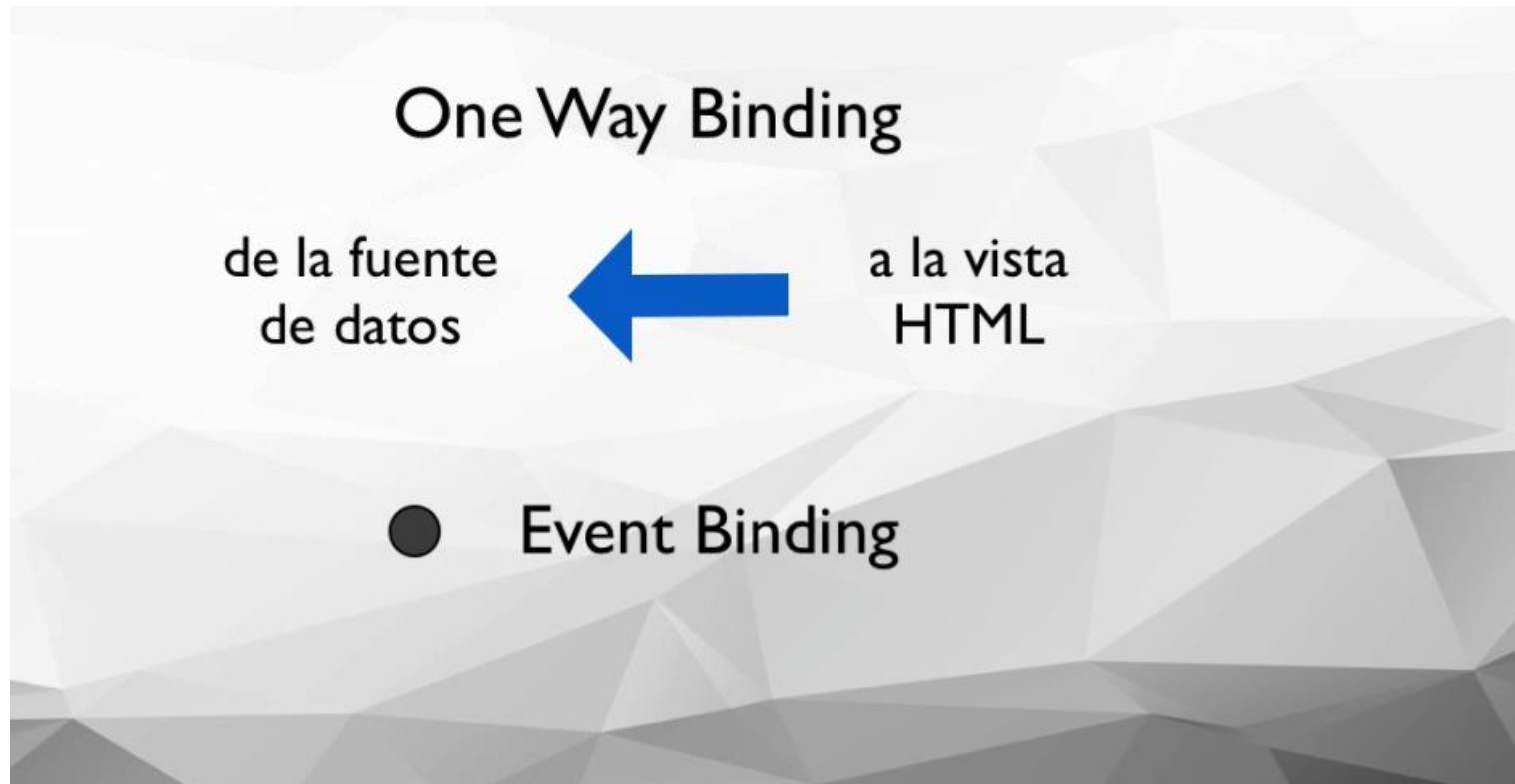
de la fuente  
de datos



a la vista  
HTML

- Interpolación
- Property Binding

## Data Binding



## Data Binding

### Two Way Binding

de la fuente  
de datos



a la vista  
HTML

● Two Way Binding

# { Angular }

## Directivas

Formador: Ezequiel Llarena Borges

## Directiva

Clase Angular con código para crear, formatear, interaccionar elementos HTML con el DOM. Tipos:

- **Componentes**
- **Estructurales** (alteran el layout del HTML en el que se encuentran)
- **Atributos** (funcionan como un atributo HTML, similar al *property binding*)



# { Angular }

## Componentes

Formador: Ezequiel Llarena Borges

## Comandos Angular CLI

**ng generate component <nuevo-componente>**

- Crear un nuevo componente

**ng g c <nuevo\_comp> --spec false**

- Versión simplificada anterior
- Omite archivo de pruebas `app.component.spec.ts`

# { Angular }

## Formularios

Formador: Ezequiel Llarena Borges

## Técnicas de control de formularios

- **Template Driven**
  - Lógica de captura de datos + Validación desde HTML
- **Reactive**
  - Gestión del Formulario desde TypeScript
  - Mayor control de la gestión de formularios
  - En tiempo real se van actualizando los valores de los campos del formulario
  - Clases: **FormControl**, **FormGroup**, **FormBuilder**

## Validación HTML

- Por defecto, Angular elimina la validación HTML5
- **NgForm** tiene su propio sistema de validación mediante **Estados**:
  - Dirty
  - Pristine
  - Touched
  - Valid
  - Invalid