

Desarrollo Web

# Acta



## ***Fase 1***

Marzo 25, 2025

## **Pontificia Universidad Javeriana**

Carlos Eduardo Rojas Martínez  
Alejandro Caicedo Caicedo  
Juan Andrés Rodríguez Rubio  
Carlos David Mercado Gallego  
Juan Sebastián Forero Moreno

1. Se realiza el control de asistencia, el día 25 de marzo.



2. Como instancia se verifican las fechas detalladas de sustentación en la cual se sustentará el proyecto tal y como está estipulado.

Planned	Carlosmercq	High	Sustentación de proyecto. Actualizada	27 de marzo de 2025
	Sebastian Forero			
	Carlos Rojas			
	Alejandro Caicedo			
	Juan Rodríguez			

3. Se introduce clase de teoría en la cual se tratan los siguientes temas:

#### 1. Angular y TypeScript

- **Angular:** Es un **framework** para desarrollo de **frontends**. Facilita la creación de aplicaciones web, integrando HTML, CSS y JavaScript de forma más eficiente.

- **TypeScript:** Angular utiliza TypeScript, que es un **superset de JavaScript** que incluye tipado estático. Esto ayuda a mejorar la calidad del código y la depuración, aunque algunas personas se sienten intimidadas por el tipado.

## 2. Ventajas de Angular

- **Arquitectura y rendimiento:**
  - Reduce la carga en el servidor, ya que no es necesario enviar todo el HTML, CSS y JS con cada solicitud.
  - **SPA (Single Page Applications):** Las aplicaciones de una sola página cargan todo el contenido una vez y luego cargan solo los datos adicionales, mejorando la velocidad y reduciendo el tráfico innecesario.
- **Escalabilidad:**
  - Las aplicaciones en Angular son más escalables, ya que solo se carga la información necesaria, reduciendo el uso de recursos en cada petición.

## 3. Diferencias con páginas tradicionales

- **Páginas estáticas vs dinámicas:**
  - Las páginas tradicionales cargan todo el contenido en cada solicitud, lo que requiere una gran cantidad de recursos del servidor. En Angular, solo se cargan los datos necesarios (usando JSON), lo cual es mucho más eficiente.

## 4. Arquitectura del Proyecto

- **Backend y Frontend separados:**
  - En Angular, el **servidor de frontend** (por ejemplo, Apache) y el **servidor de backend** (por ejemplo, Spring) están separados. El backend se encarga de manejar la lógica de negocios, mientras que el frontend maneja la interfaz de usuario.

## 5. Procesos de Petición y Respuesta

- En una aplicación Angular, tras la primera carga de la página (HTML, CSS, JS), las siguientes solicitudes solo requieren los **datos en formato JSON**, lo que mejora la eficiencia al evitar la carga completa del contenido estático en cada solicitud.

## 6. Instalación de Angular

- **Requisitos previos:** Para comenzar a trabajar con Angular, primero se debe instalar **Node.js** y luego ejecutar una línea de comando para instalar Angular.
- **Creación de proyectos:** Una vez instalado, el siguiente paso es crear el primer proyecto en Angular y ejecutar la aplicación.

#### 4. El docente solicita la instalación de Angular por medio de estos comandos:

- Instalación de Angular
- Instalar Node.js
- → <a href="https://kinsta.com/es/blog/como-">https://kinsta.com/es/blog/como-</a>
- instalar-node-js/
- Instalar Angular
- → npm install -g@angular/cli

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <!-- Generator: Adobe Illustrator 19.2.1, SVG Export Plug-In . SVG Version: 6.00 Build 0) -->
3 <svg version="1.1" id="layer" xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink" x="0px" y="0px"
4   viewBox="0 0 652 652" style="enable-background:new 0 0 652 652;" xml:space="preserve">
5   <style type="text/css">
6     .st0{fill:#5333ED;}
7   </style>
8   <g>
9     <g>
10      <path class="st0" d="M276.3,281.1c-4.5-4.5-9.8-0.1-15.7-10.6c-6-2.6-12.5-3.9-19-3.9c-6-2.0-12.3-1-18.3,2
11      c-5.7,2.2-10.9,5.5-15.4,9.6-1.7-9.9c-0.9-1.2-1.1-5.3-4.1-5c-2.5,0-4.5,2.1-4.5,4.5V375c0,2.2,1.7,4.2,4.4,5.1h19.2
12      c2.5,0,4.5-2.1,4.5-4.5v-58.5c0-2.8,0.6-5.6,1.7-8.2c1.1-2.5,2.6-4.8,4.5-6.7c4-4.9,5-6.3,15.2-6.2c2.8,0,5.6,0.6,8.1,1.7
13      c5.1,2.2,9.1,6.3,11.2,11.4c1,2.5,1.5,5.2,1.5,7.9v58.5c0,2.5,2.4,5.4,5.4,5.1h19.2c2.5,0,4.5-2.1,4.5-4.5v-58.5
14      c0-6-1.3-13.2-3.9-19.3c284.1,291,280.6,285.7,276.3,281.1l276.3,281.1z M493.6,268.9h-75.4c-2.5,0-4.5,2.1-4.5,4.5v19.1
15      c0,2.5,4.5,4.5,4.5h52.5,0,4.5,2.4,5.4,5.4V308c0,6-6.1,3.1,13.2,3.9,19.2c5,1.1,11.0,14.5,21,3.26,3.26,46.6,1.2,6,12.6,3.9,19.2,3.9
16      M944.2,5,0,4.5-2.1,4.5-4.5v-19.1c0-2.5-2.4-5.4-5.4-5.1h-16.8c-2.9,0,5.7-0.6,8.3-1.7c-2.5-1.1,4.8-2.7,6.8-4.6
17      c-2-2-3.5-4.3-4.5-6.8c-1.1-2.6-1.7-5.5-1.6-8.4v-28.3c0-2.3,1.7-4.2,4.4,5.3,7.7c2.5,0,4.5-2.1,4.5-4.5v-19
18      C498,271,496,269,493.6,268.9l493.6,268.9z M616.3,266.9c-1.3,0-2.6,0.6-3.4,1.6l-6.7,8c-1.6,2.4,5.2-2.6,4.0,6
19      c-1.6-1.4-3.3-2.7-5.1-4c-3-2.1-6.2-4.9-5.5-6c-3.3-1.6-6.7-2.9-10.2-3.7c-3.5-0.9-7.1-1.3-10.7-1.3c-7.6,0-15.1,4-22.4,3
20      c-6.8,2.8-12.8,6.9-18,12.1c-5.2,5-9.3,11.7-12.1,18.6c-3.7,5-4.4,15.5-4.4,23.6c-0.1,8.5,1.3,16.8,4.4,24.7
21      c2.7,6.9,9.1,13.2,12.2,18.4c5-1.5,11.2,8.9,18,11.4c7,2.7,14.4,4.2,22.4c3.5,0,7.1-0.5,10.5-1.4c3.5-0.9,6.9-2.1,10.2-3.7
22      c3-1.6,6.4-3.4,9.4-5.5c1.7-1.2,3.3-2.4,4.9-3.7c2-1.6,5-1.2,6.4,0.9l6.7,9.3c2.5,3.6,8.1,1.8,8.1-2.6V271.4
23      C620.8,268.9,618.8,266.9,616.3,266.9l616.3,266.9z M590.1,332.3c-1.4,3.5-3.4,6.7-6.9,6c-2.5,2.8-5.6,5.1-8.9,6.7
24      c-3.4,1.6-7.1,2.5-10.9,2.5c-3.7,0-7.4-0.7-10.9-2c-3.4-1.3-6.4-3.2-8.9-5.8c-2.6-2.7-4.6-5.9-5.9-9.5c-1.5-4.2-2.3-8.5-2.1-13
25      c-0.1-4.4,0.7-8.8,2.2-12.9c1.3-3.4,3.4-6.5,5.9-9.1c2.5-2.6,5.5-4.5,8.9-5.8c3.5-1.3,7.2-2.2,10.9-1.9c3.8,0,7.5,0.8,10.9,2.5
26      c3.3,1.6,6.3,3.7,8.8,6.4c2.6,2.8,4.6,6.1,6.9,6.5c93.1,137,593.1,325,590.1,332.3l598.1,332.3z M44.6,268.9l0.5,3
27      c-2.5,0,4.5-2.4,5.4,5.4V375c0,2.5,2.4,5.4,5.4,5.1h19.3c2.5,0,4.5-2.1,4.5-4.5v273.2c49.1,270.8,47.1,268.7,44.6,268.7l44.6,268.7z
28      M123.8,276c2.3-2.9,0.3-7.2-3.4-7.3h-22c-1.3,0-2.6,0.6-3.4,1.6l51.6,321c-1.5,1.7-1.5,4.2,0.5,9.143,9.5,8.8
29      c0.9,1.2,1.6,3.4,1.6h22c3.7-0.1,5.7-4.4,3.5-7.3l-39.3-44.8c-1.5-1.7-1.5-4.2,0.5-9.1l23.8,276z M166.1,268.7
30      c-1.3,0-2.6,0.5-3.4,1.5l-0.9,11-18.3,20.9c-0.7,0.8-1.1,1.9-1.1,2.9V375c0,2.5,2.4,5.4,5.4,5.1h166.2,5,0,4.5-4.5V273.2
31      C170.4,270.8,168.5,268.8,166.1,268.7l166.1,268.7z M396.1,320.6c-3.2-3-7.5-3-11.1-6.9c-4.3-1.6-9-2.4-13.6-2.4h-29.3
32      c-1.9,0,3.7-0.8,5-2.1c-1.2-1.4-1.8-3.2-1.8-5c0-4.3-3.7,3.7-3.7-3.7-3.7c2.5,0,4.5-2.1,4.5-4.5v-18.8c0-2.5-2.4-5.4-5.4-5.1h-53.2
33      c-4.8-0.1-9.6,0.9-14.2,8c-8.3,3.6-14.8,10.3-18.3,18.6c-1.8,4.4-2.7,9-2.7,13.8c0,4.7,0.9,9.4,2.8,13.7
34      c3.4,8.4,9.1,15.2,18.2,18.9c4.3,1.9,9.2,8.1,7.2,8.8V375c-1.0,1.5,8.2,3.6,1.5,4c0.3,3.5-2.4,6.5-5.9,6.5h-60.7
35      c-2.5,0-4.5-2.4,5.4,5.4V375c0,2.5,2.4,5.4,5.4,5.1h59.9c9.3,0,18.2-3.7,24.7-10.3c3.1-3.2,5.6-7.3-11.2c1.9-4.3,2.8-9,2.8-13.7
36      c0-4.6-0.9-9.1-2.8-13.3c401.7,327.1,399.2,323.6,396.1,320.6l396.1,320.6z M396.1,320.6"/>
37   </g>
38 </g>
39 </svg>
40

```

- ```

      _\_\_\_\_\_|\_|_\_\_\_\_
      |||'\_/_/_/_/\_/_/_/_\
      ||_||||(|_|(|_||||_|/_
      |__|_|_|\\_/_\\_/_\\_/_|

```
- |                            |
|----------------------------|
| Angular CLI: 12.x.x        |
| Node: 16.x.x               |
| Package Manager: npm 7.x.x |
| OS: darwin x64             |

|                            |
|----------------------------|
| Angular CLI: 12.x.x        |
| Node: 16.x.x               |
| Package Manager: npm 7.x.x |
| OS: darwin x64             |

- Primer proyecto angular
- ng new <nombre del proyecto>
- Abrir navegador
- ng serve -o ng serve

- **Angular Live Development Server is listening on localhost:4200, open your browser on <http://localhost:4200>**

- **Angular Live Development Server is listening on localhost:4200, open your browser on <http://localhost:4200>**

**5.** Se realiza nuevamente la explicación de tema, donde se abarcan temas como:

## 1. Modelos y DTOs

- **Modelos en Angular:** Los modelos en Angular se refieren a las **clases TypeScript**, que están directamente relacionadas con los **DTOs (Data Transfer Objects)** en el backend. Esto facilita la **correlación entre el frontend y el backend**.

- **Mapeo de JSON:** El mapeo de **JSON a TypeScript** se hace de manera directa y automática, lo que simplifica la conversión de datos entre el backend y el frontend sin necesidad de traducciones complicadas.

## 2. Servicios

- **Lógica de negocio:** Los **servicios** en Angular se encargan de la **lógica de negocio** y de hacer solicitudes al **backend**, como peticiones HTTP.
- **Restricción de funciones en los servicios:** Los servicios **no deben** realizar operaciones como **sumas, restas o accesos a bases de datos**. Su única función es manejar **solicitudes** (peticiones HTTP).

## 3. Componentes

- **Definición de componentes:** En Angular, los **componentes** representan diferentes **partes de la interfaz** de la aplicación, como:
  - Menú
  - Footer
  - Otros elementos de la página
- **Estructura de un componente:** Cada componente tiene:
  - **HTML:** La estructura visual
  - **CSS:** El estilo
  - **TS:** La lógica en TypeScript
  - **Spec:** Pruebas unitarias para el componente

## 4. Arquitectura y MVC

- **Ventajas de Angular sobre React:** Angular **obliga** a trabajar con una **arquitectura estructurada** (modelos, servicios y componentes), lo que ayuda a respetar el patrón **MVC (Modelo-Vista-Controlador)** y proporciona una **arquitectura clara**.
- **Claridad en la arquitectura:** La estructura de Angular asegura una **arquitectura más organizada y sostenible**, facilitando el desarrollo y mantenimiento de la aplicación.

## 5. Creación de la Página en Angular

- **Componente principal:** El componente principal de la aplicación debe contener **HTML, CSS, TS y spec**. Este componente principal es típicamente `app.component.html`.
- **Datos dinámicos:** El docente menciona que se pasará un título y una lista de datos al componente principal. Utilizando la directiva `*ngFor`, se recorrerán los datos y se imprimirán los nombres y apellidos de los usuarios.
- **Ciclo `ngFor`:** El ciclo `*ngFor` se usa para iterar sobre una lista de datos (como diccionarios en Python) y mostrar los elementos en el HTML.

- **Importación de librerías:** Es crucial **importar las librerías adecuadas** para que las funcionalidades (como `*ngFor`) funcionen correctamente.

## 6. Próximos Pasos y Actividades

- **Ejercicio de práctica:** El siguiente paso para los estudiantes será crear el componente principal con los datos y utilizar el ciclo `*ngFor` para mostrar la lista de usuarios en el HTML.
- **Importación de librerías:** Asegurarse de **importar la librería** correspondiente para que todo funcione correctamente en el proyecto.

### 6. Componente Principal

|                                                            |
|------------------------------------------------------------|
| <code>&lt;h1&gt;Datos&lt;/h1&gt;</code>                    |
| <code>&lt;ul&gt;</code>                                    |
| <code>&lt;li *ngFor="let usuario of datos"&gt;</code>      |
| <code>{{ usuario.nombres }} {{ usuario.apellidos }}</code> |
| <code>&lt;/li&gt;</code>                                   |
| <code>&lt;/ul&gt;</code>                                   |

El docente nos indica que:

- **`<h1>Datos</h1>`:** Esto es simplemente un título que se muestra en la página, y aparecerá como un encabezado grande que dice "Datos".
- **`<ul>`:** Define una lista desordenada (bullet points).
- **`*ngFor="let usuario of datos"`:** Esta es la directiva de Angular `*ngFor`, que permite iterar sobre el arreglo `datos`. Por cada elemento en `datos`, se va a crear un `<li>` (un item de lista) y se asignará el valor de `usuario.nombres` y `usuario.apellidos` dentro de ese item.
- **`{{ usuario.nombres }} {{ usuario.apellidos }}`:** Aquí estamos utilizando **interpolación de datos** de Angular, lo que significa que **Angular reemplazará** `usuario.nombres` y `usuario.apellidos` con los valores de esas propiedades de cada objeto usuario.