

Aleixer Alvarado Bernal

aleixer.alvarado@softtek.com

Angular aplicado

Se describe el uso y manejo de Angular como herramienta de desarrollo

Angular

app

Contenido

[Angular 2](#_Toc114722876)

[url: https://angular.io/ 2](#_Toc114722877)

[Que es 2](#_Toc114722878)

[Fabricante 2](#_Toc114722879)

[Características 2](#_Toc114722880)

[Lenguaje 2](#_Toc114722881)

[Ventajas 2](#_Toc114722882)

[Requerimientos 2](#_Toc114722883)

[Instalar Node 2](#_Toc114722884)

[Instalar Angular CLI 5](#_Toc114722885)

[Crear mi primera aplicación 6](#_Toc114722886)

[Comando “ng new” 6](#_Toc114722887)

[Comando “ng serve” 7](#_Toc114722888)

[Instalar Material 8](#_Toc114722889)

[url: https://material.angular.io/ 8](#_Toc114722890)

[Aplicaciones SPA vs No SPA: 14](#_Toc114722891)

[Peticiones Comunes (No-SPA) 14](#_Toc114722892)

[Peticiones (SPA) 15](#_Toc114722893)

[Angular - Estructura y flujo 15](#_Toc114722894)

[Crear Componentes “ng g c” 17](#_Toc114722895)

[Manual 17](#_Toc114722896)

[Automático con el comando NG de CLI 17](#_Toc114722897)

# Angular

### url: <https://angular.io/>

## Que es

Un framework para crear aplicaciones WEB SPA (Single page application)

## Fabricante

Google

## Características

* Framework de desarrollo frontend
* Estructura y simplifica el código
* Usa el patrón MVC
* Basados en componentes
* Código abierto

## Lenguaje

Typescript (derivación o mejora de JavaScript, + tipado)

## Ventajas

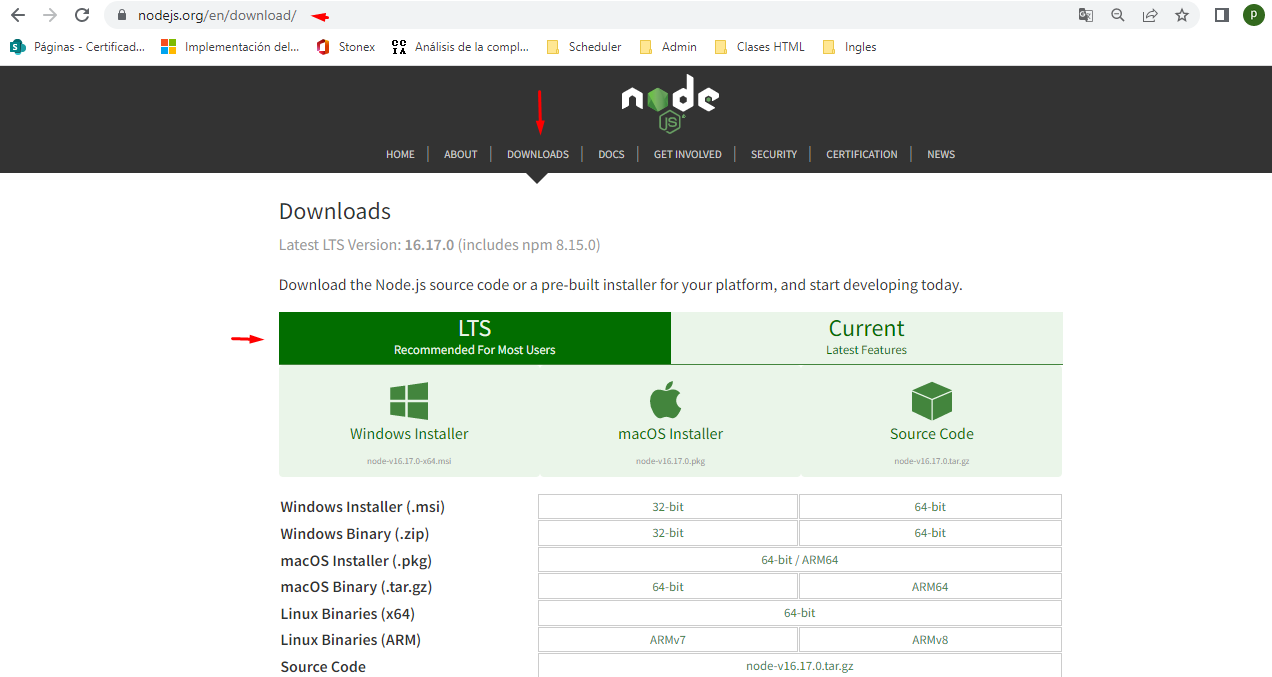
* Desarrollo y ejecución rápido
* Es modular o reutilizable
* Multiplataforma
* Fácil escalable
* Soporte y futuro estable
* Gran demanda

## Requerimientos

* Node.js (Entorno JavaScript para servidor) “nodejs.org/es”
* Npm (Node package manager)
* Angular (CLI Command line Interface) “angular.io”
* Visual studio Code (Editor)
* Git (Repo)

# Instalar Node

Desde la url <https://nodejs.org/en/download/> descargar el paquete según la maquina en la que te encuentres



Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

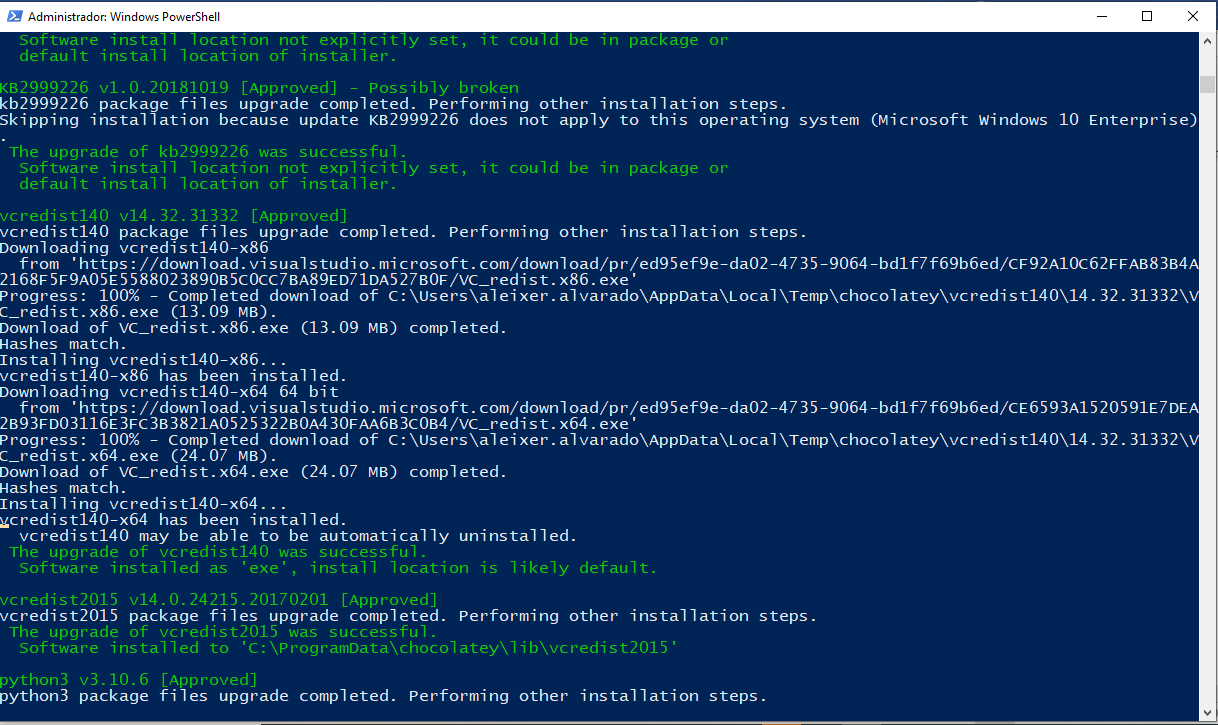
Texto

Descripción generada automáticamente

Al darle continuar con cualquier tecla, este inicia la instalación de paquetes

Para activar los script por consola

“Set-ExecutionPolicy -Scope CurrentUser -ExecutionPolicy RemoteSigned”



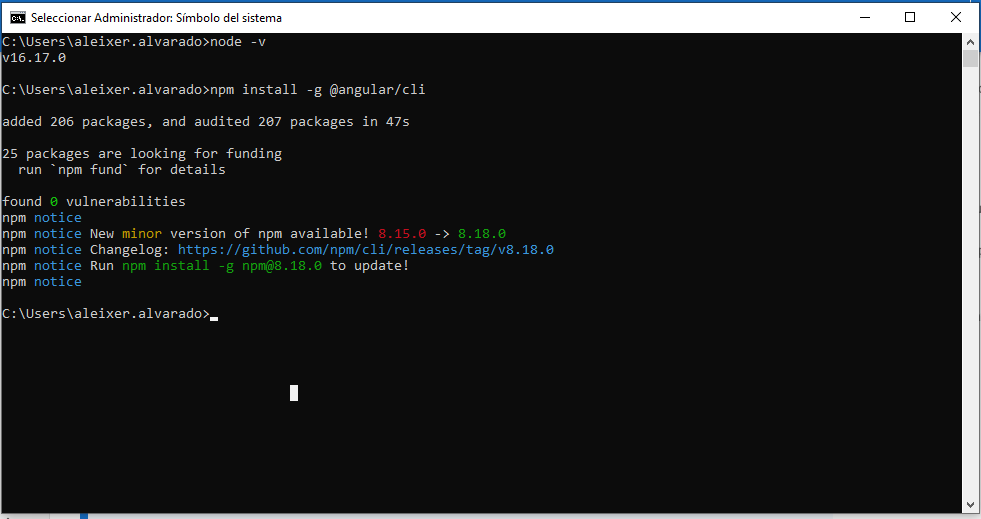
Después de la instalación, se abre una consola y se ejecuta “node -v” para ver si instalo la última versión

Texto

Descripción generada automáticamente con confianza baja

# Instalar Angular CLI

Desde la consola instalar con el comando “npm install -g @angular/cli”



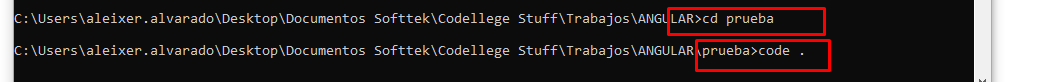
# Crear mi primera aplicación

## Comando “ng new”

1. Desde la consola dirigirse a la ruta donde se creará la aplicación
2. Para crear una aplicación se ejecuta el comando “ng new my-app” donde my-app es el nombre que desee darle a la aplicación.
3. Ejecutar y responder las preguntas de preferencias al generar la aplicación.
4. ¿Le gustaría usar un archivo routing? (‘S’ le creara un archivo para el sistema de rutas) (‘N’ sin rutas)
5. ¿Agregar hoja de estilos? Escoger el que se requiera y darle “enter”
6. Por medio del comando “cd my-app” donde “my-app” es el nombre de la aplicación para ingresar a la carpeta.
7. Por medio del comando “code .” se abre visual estudio code.

Texto

Descripción generada automáticamente



Se debe generar lo siguiente Texto

Descripción generada automáticamente

# Ejecutar con el Comando “ng serve”

Para levantar el servidor con nuestra aplicación en funcionamiento, ejecutamos el comando “ng serve -o” esto genera la url que debemos abrir en el navegador   
Texto

Descripción generada automáticamente

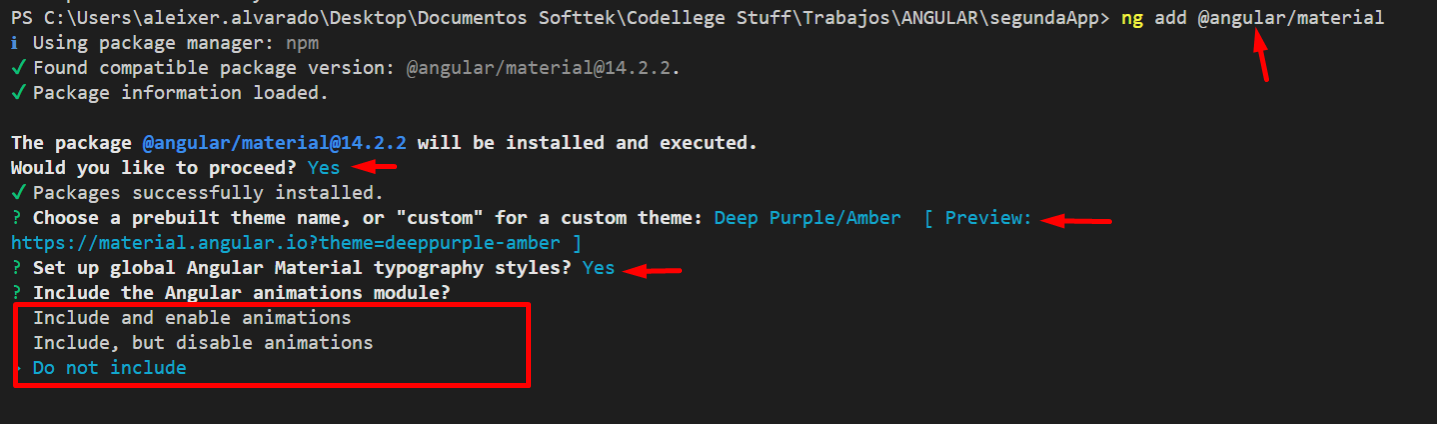
# Instalar Material

### url: <https://material.angular.io/>

la librería de material permite usar componentes prediseñados que nos ayudan a darle diseño y estilo a la aplicación.

Es importante siempre que estemos trabajando en material, que nos remitamos a la documentación, donde nos indicara paso a paso como hacer uso de sus herramientas.

Ejecutar en la carpeta del proyecto desde consola, el siguiente comando “ng add @angular/material”



Texto

Descripción generada automáticamente

Se agrega en el archivo “app.module.ts” los componentes de material que se requieran, según se muestra en el ejemplo.   
Texto

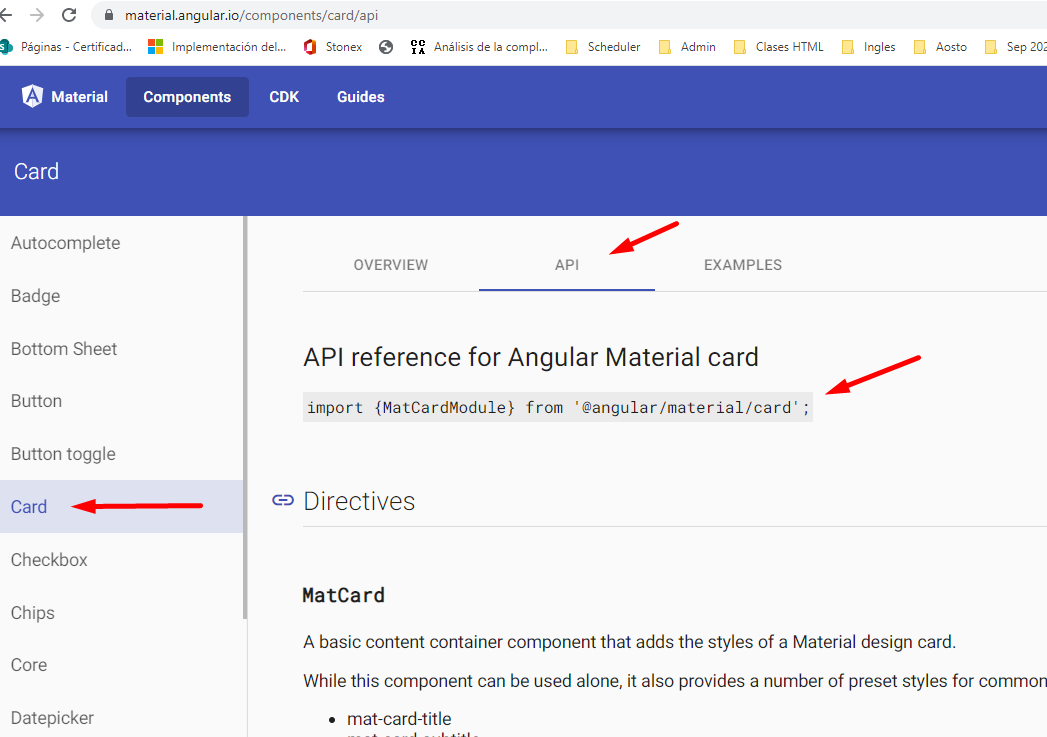
Descripción generada automáticamente

Luego se agrega en el componente html, según se muestra   


Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Otro ejemplo agregando una CARD y un NAV

Desde la documentación de material, seleccionamos lo que se requiere   


Texto

Descripción generada automáticamente

Interfaz de usuario gráfica, Aplicación, Correo electrónico, Sitio web

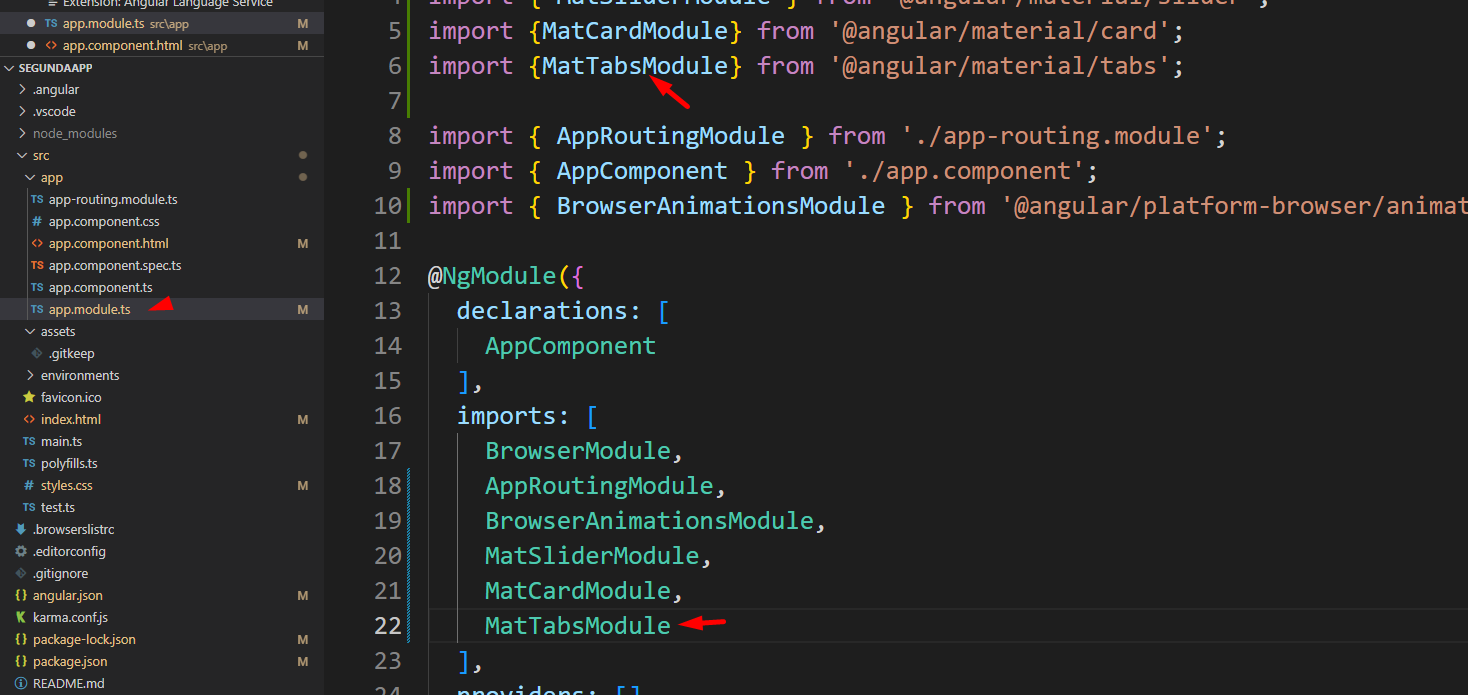
Descripción generada automáticamente

Captura de pantalla de computadora

Descripción generada automáticamente

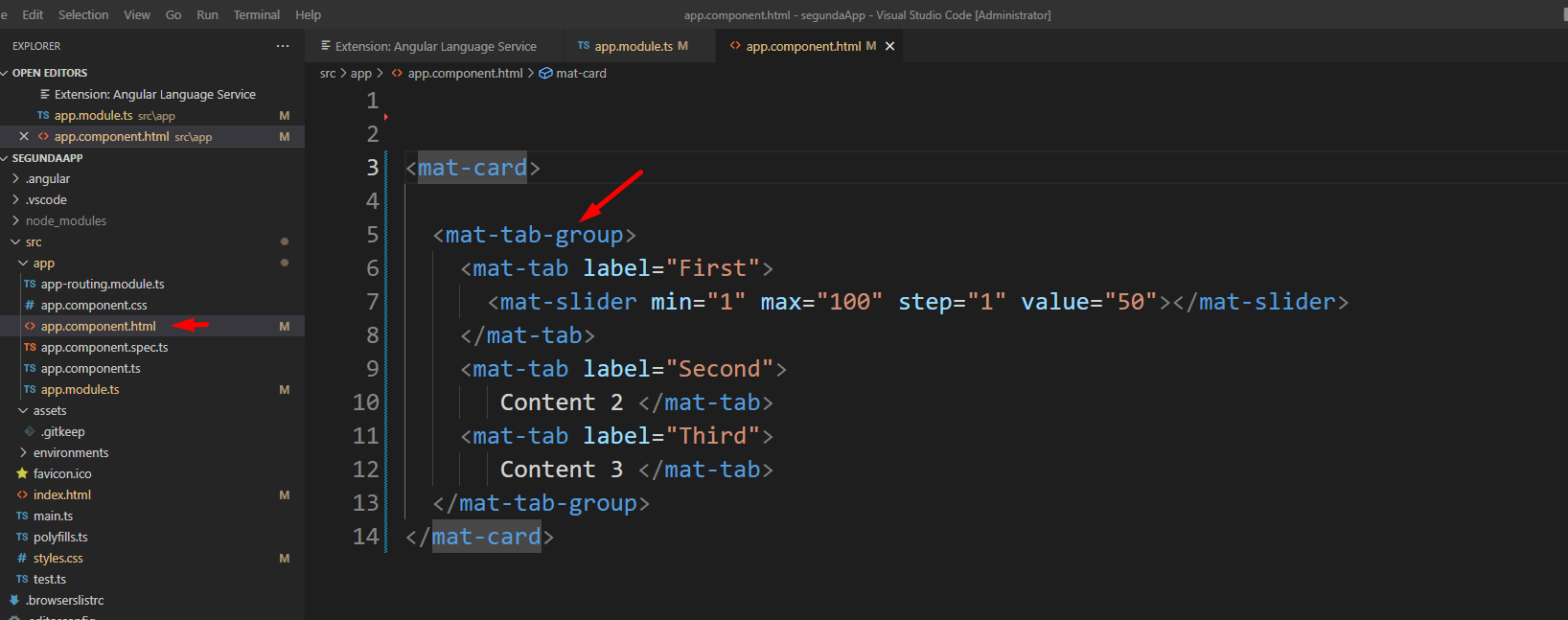
Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente



Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente



Resultado

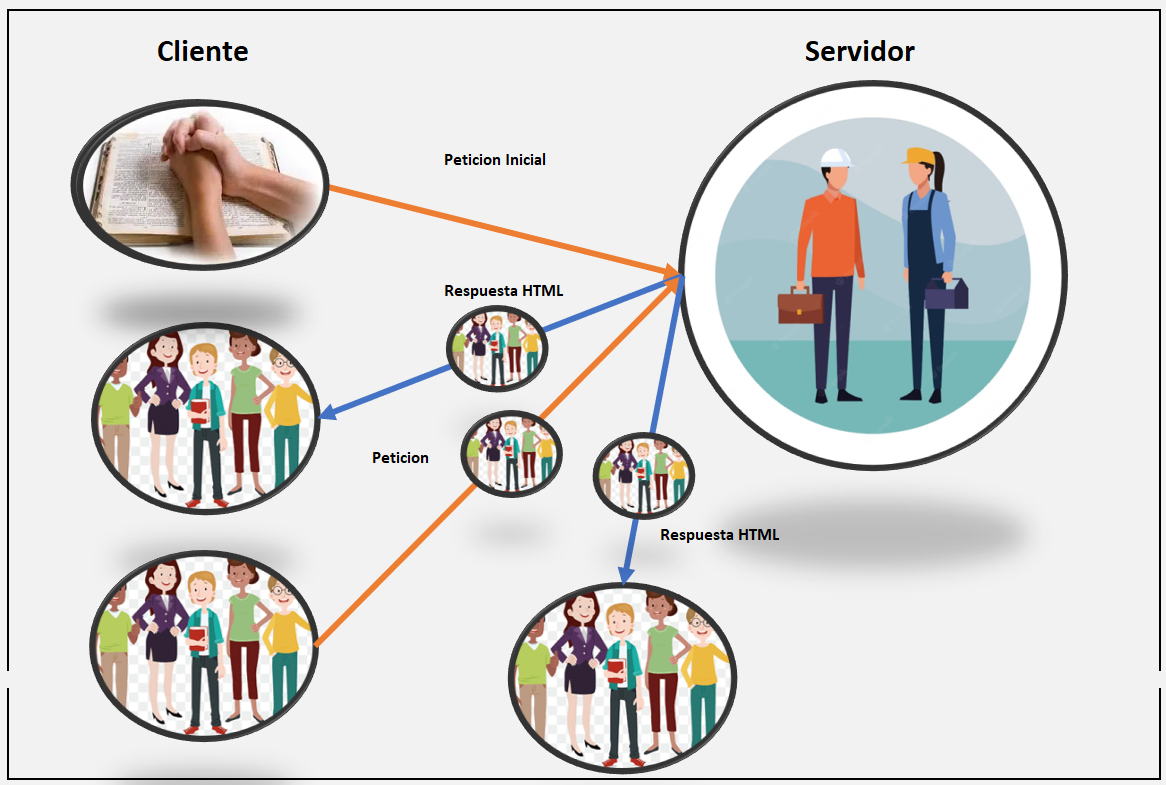
Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

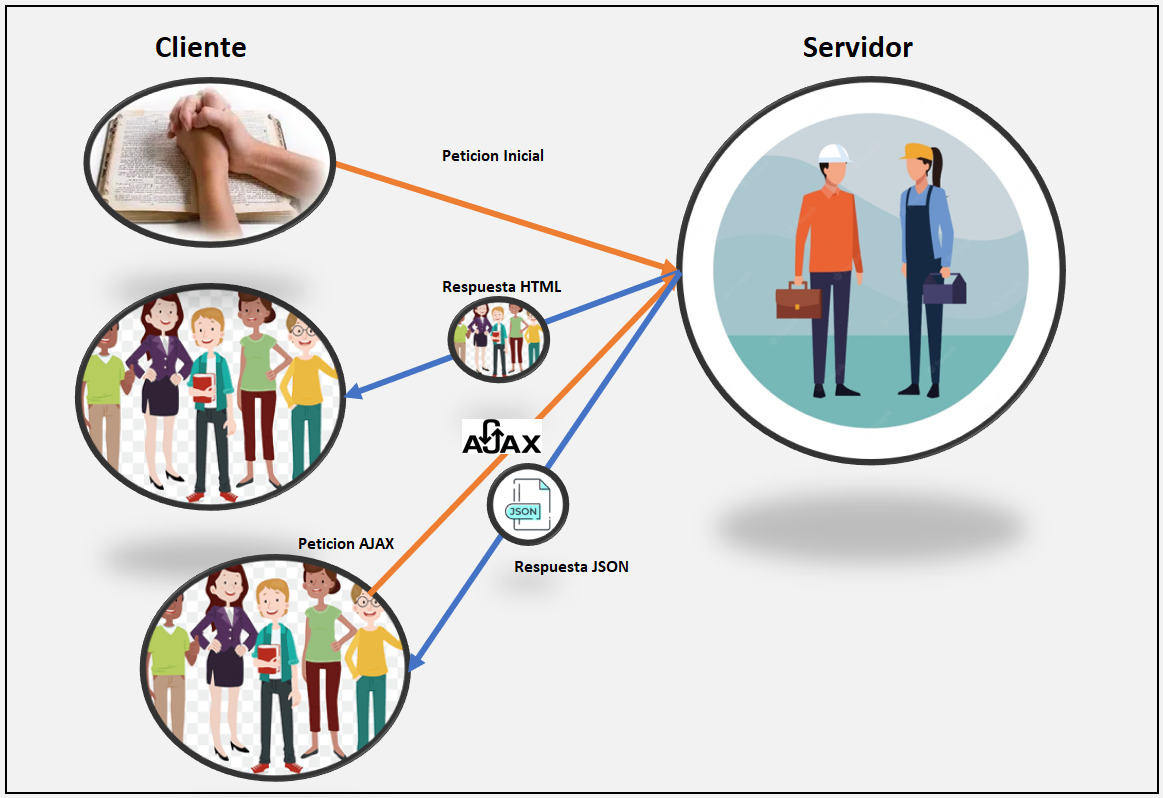
# Aplicaciones SPA vs No SPA:

Como funciona ANGULAR (SPA)

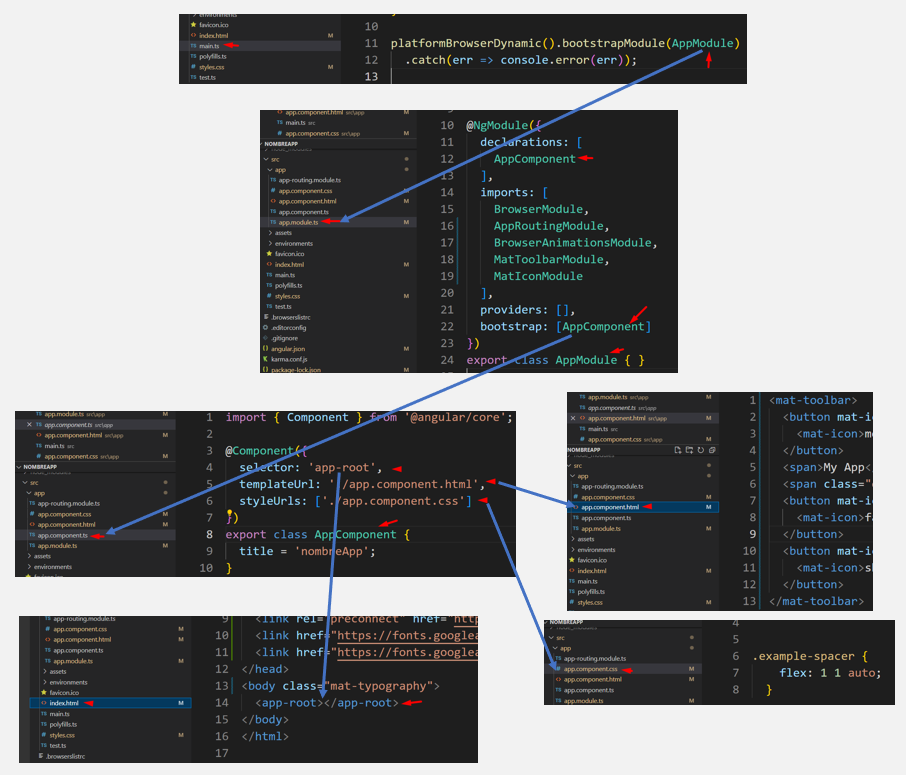
## Peticiones Comunes (No-SPA)



## Peticiones (SPA)



# Angular - Estructura y flujo



# Agregar Componentes “ng g c”

## Manual

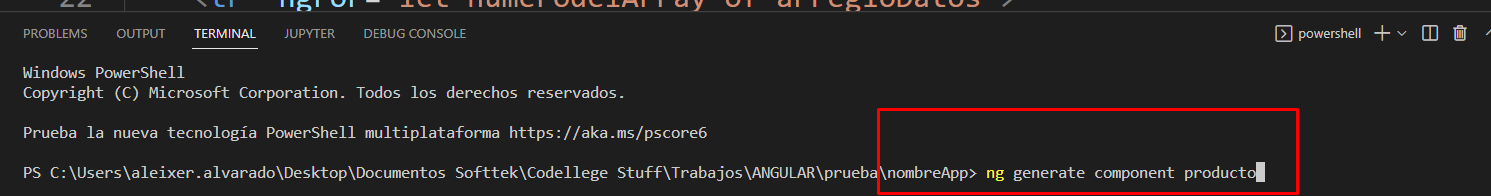
Se crean de forma manual, archivo por archivo y se agrega el componente .ts al módulo principal

## Automático con el comando NG de CLI

Crear componente de forma Automática con la herramienta de CLI

Desde la consola o la terminal de Visual estudio

1. Con el comando “ng generate component nombreComponente” donde nombreComponente es el nombre del componente a crear



Resultado

Texto

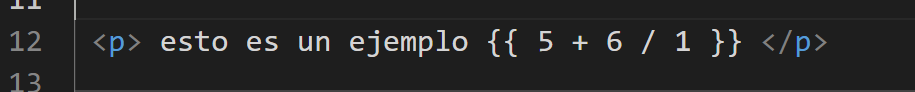
Descripción generada automáticamente

# Interpolación {{}} en Angular

url: <https://angular.io/guide/interpolation>

Angular usa la interpolación para agregar dinamismo al contenido, Cuando Angular ve en un “template” algo escrito entre dobles llaves {{}} lo evalúa y lo trata de convertir en una cadena, para luego pasarlo al “template”. Esto puede ser desde la lectura de una variable a la ejecución de una expresión o función.

Ejemplo:



Resultado:



# Analizando las directivas estructurales

Uso de ciclos por medio de la directiva estructural “\*ngFor”

Uso de condicionales por medio de la directiva estructural “\*ngIf”

