Angular

Índice

Angular

1. Concepto
2. Decoradores
3. Componentes
4. Directivas
5. Módulos
6. Servicios
7. Inyección de dependencias
8. Diagrama

¿Que es Angular?

Angular es un framework para la creación de aplicaciones cliente en HTML y JavaScript.

Que es un framework

Un framework es un esquema o marco de trabajo que ofrece una estructura base para elaborar un proyecto con objetivos específicos, una especie de plantilla que sirve como punto de partida para la organización y desarrollo de software

Arquitectura de Angular

Angular es un framework que está basado en:

- Componentes
- Directivas
- Módulos
- Servicios

Decorador

Angular

1. Concepto
2. **Decoradores**
3. Componentes
4. Directivas
5. Módulos
6. Servicios
7. Inyección de dependencias
8. Diagrama

Decorado

Un decorador permite dotar funcionalidades y responsabilidades dinámicamente a objetos.

Decorador y Angular

Un decorador en Angular es la unión de HTML con la clase de TypeScript.

Decorado

Para identificar un decorador es “@” seguido del nombre de la función.

Angular Decorador

Ejemplo de decorador:

```
@Component({
  selector: 'app-root',
  templateUrl: 'miapp.component.html',
  styleUrls: ['miapp.component.css']
})
```

Angular Decorador

La estructura de un decorador de angular es:

- **selector:** es nombre de una nueva etiqueta.
- **templateUrl:** es la ruta del archivo HTML.
- **styleUrls:** es un arreglo de las rutas de los archivos CSS.

Componentes

Angular

1. Concepto
2. Decoradores
3. **Componentes**
4. Directivas
5. Módulos
6. Servicios
7. Inyección de dependencias
8. Diagrama

Componentes

Son pequeñas clases que cumplen funciones específicas.Además define la funcionalidad a una sección de html.

Componentes

Un componente puede ser:

- Barra de navegación.
- Formularios
- Campos de formularios.
- Tablas de datos.

Componentes

Un componente está compuesto por 4 archivos:

1. HTML
2. CSS
3. Clase TypeScript (es una plantilla para crear objetos)
4. Clase TypeScript Test (pruebas automatizadas que verifican su funcionamiento).

Componentes

La estructura de la clase de un componente está divida en:

1. Importaciones de módulos.
2. Decorador
3. Clase

Directivas

Angular

1. Concepto
2. Decoradores
3. Componentes
4. **Directivas**
5. Módulos
6. Servicios
7. Inyección de dependencias
8. Diagrama

Directivas

Una directiva básicamente es una plantilla dinámica de html.

Directivas Estructurales

Son instrucciones que alteran la estructura mediante la adición, eliminación y sustitución del DOM.

Directivas Estructurales

Las directivas estructurales más comunes son:

- Ngif (Esta directiva se utiliza para mostrar u ocultar elementos en función de una expresión booleana)
- Ngfor (si tienes una lista de nombres, puedes usar ngFor para recorrer cada elemento de la lista y crear un elemento HTML para cada uno).
- ngswitch (se utiliza para seleccionar un elemento de varios elementos posibles basados en una expresión. Es similar a una declaración switch/case en lenguajes de programación).

ARCHIVO.TS

ngIf (uso) “**se usa export class para su uso en cualquier otro archivo y la clase lleva el nombre del archivo Html)**

```
export class MiComponente {
  mostrarElemento = true;
}
```

Archivo: MiComponente.HTML

```
<div *ngIf="mostrarElemento">
  Este elemento se mostrará si mostrarElemento es verdadero
</div>
```

Archivo.ts

```
export class MiComponente {
  nombres = ['Juan', 'María', 'Pedro', 'Luisa'];
}
```

Archivo.HTML

```
<ul>
  <li *ngFor="let nombre of nombres">{{ nombre
}}</li>
</ul>
```

ngswitch

```
<div [ngSwitch]="opcionSeleccionada">
  <p *ngSwitchCase="'opcion1'">Contenido de la opción 1</p>
  <p *ngSwitchCase="'opcion2'">Contenido de la opción 2</p>
  <p *ngSwitchDefault>Contenido de la opción predeterminada</p>
</div>
```

Módulos

Angular

1. Concepto
2. Decoradores
3. Componentes
4. Directivas
5. **Módulos**
6. Servicios
7. Inyección de dependencias
8. Diagrama

Módulos en Angular

- Organiza una aplicación.
- Son escalables (se puede adaptar y crecer de manera eficiente sin afectar negativamente el resto del sistema..)
- Mantenibles (Debe tener una arquitectura clara y coherente para que otros desarrolladores puedan entender y trabajar en el código de manera eficiente..)

NgModule

- Es el módulo principal de la aplicación.
- Es una clase con el decorador @NgModule.
- Ayuda a organizar una aplicación en bloques de funcionalidad.

NgModule

Las propiedades más importantes son:

- **imports**: se importan los módulos para la aplicación.
- **declarations:** se declaran los componentes, y directivas creados.

NgModule

Las propiedades más importantes son:

- **providers**: exporta todos los servicios para el uso general en la aplicación.
- **bootstrap:** es la vista principal de la aplicación, aloja todas las vistas de la aplicación.

Ejemplo de NgModule

```
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import { AppComponent } from './app.component';

@NgModule({
  imports: [ BrowserModule ],
  declarations: [ AppComponent ],
  providers: [ ],
  bootstrap: [AppComponent]
})

export class AppModule { }
```

Servicios

Angular

1. Concepto
2. Decoradores
3. Componentes
4. Directivas
5. Módulos
6. **Servicios**
7. Inyección de dependencias
8. Diagrama

Servicios

Típicamente, es una clase con un propósito específico.

Servicios

Los servicios pueden ser:

- Servicios de datos.
- Servicios de registros.
- Configuraciones de la aplicación.

Inyección de dependencias

Angular

1. Concepto
2. Decoradores
3. Componentes
4. Directivas
5. Módulos
6. Servicios
7. **Inyección de dependencias**
8. Diagrama

Inyección de

- Es un método que administra la instancia de una clase.
- La mayoría de dependencias son servicios.
- Proporcionar nuevos servicios a los componentes.
- Proporciona objetos o funciones necesarios para un componente de manera automática y transparente.

Función

- Un Inyector mantiene los servicios en propiedades de la clase.
- Si un servicio no está instanciado en la clase, el inyector crea una nueva instancia del servicio requerido (la instancia se utiliza para acceder a los métodos y propiedades de la clase y realizar las operaciones necesarias en la aplicación app.)

Diagramas

Angular

1. Concepto
2. Decoradores
3. Componentes
4. Directivas
5. Módulos
6. Servicios
7. Inyección de dependencias
8. **Diagrama**

Arquitectura de Angular

Diagrama

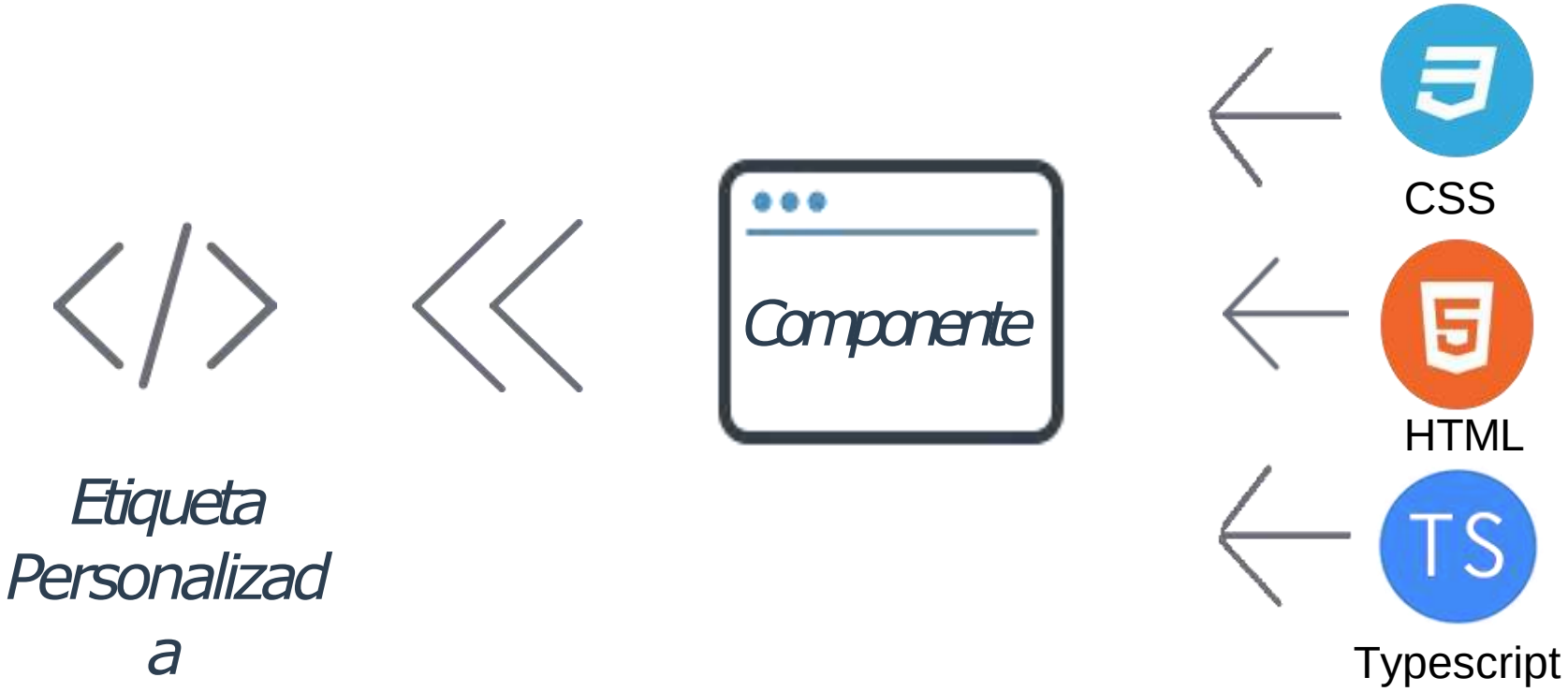
Es una representación visual de la estructura y relaciones entre los diferentes componentes y servicios de una aplicación, lo que ayuda a entender cómo funcionan juntos y cómo se pueden mejorar o escalar en el futuro.

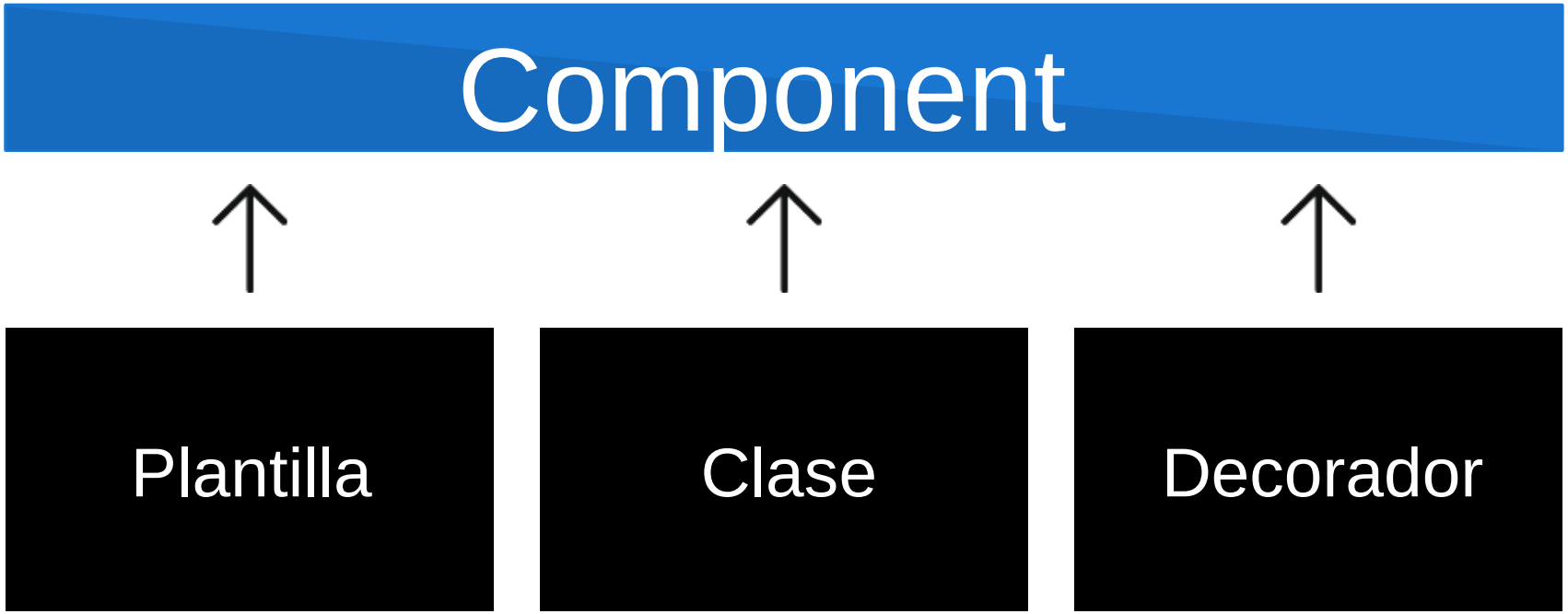
Servicio

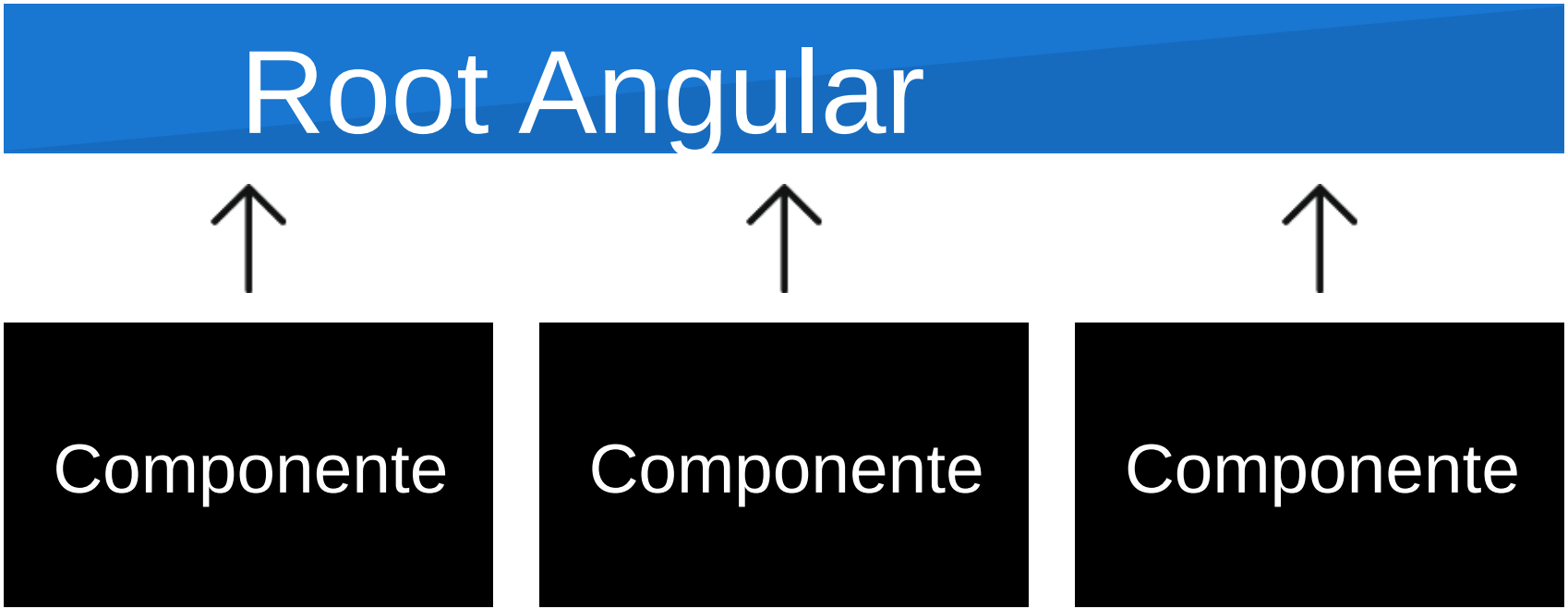
Componente

Componente

Componente


Componente
CSS
HTML
TS
Typescript
Etiqueta Personalizada


Component
Plantilla
Clase
Decorador


Root Angular
Componente
Componente
Componente

Gracias!