

Sesión 01

Introducción a Angular 17

Instructor:
ERICK ARÓSTEGUI
earostegui@galaxy.edu.pe



17

ANGULAR

FULL-STACK DEVELOPER

ÍNDICE

01 ¿Qué es Angular?

02 Evolución y ventajas de Angular.

03 Arquitectura de Angular

04 Herramientas de desarrollo
(Node.js, npm y VSC).

05 Mi primera aplicación.

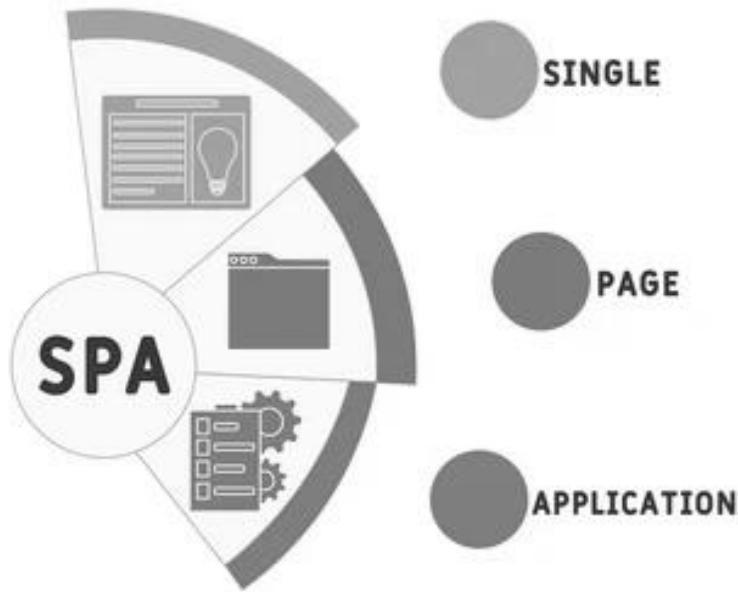
01



¿Qué es Angular?

→ ¿Qué es Angular?

¿Qué es una aplicación de una sola página (SPA)?



- Una aplicación de una sola página es un **sitio web o aplicación web que interactúa dinámicamente** con el usuario reescribiendo la página actual en lugar de cargar la página completa desde el servidor.
- Una aplicación de una sola página (SPA) es utilizada por la mayoría de las aplicaciones populares, incluidas Facebook, Gmail, Twitter, Google Drive, Netflix y una gran cantidad de otras.

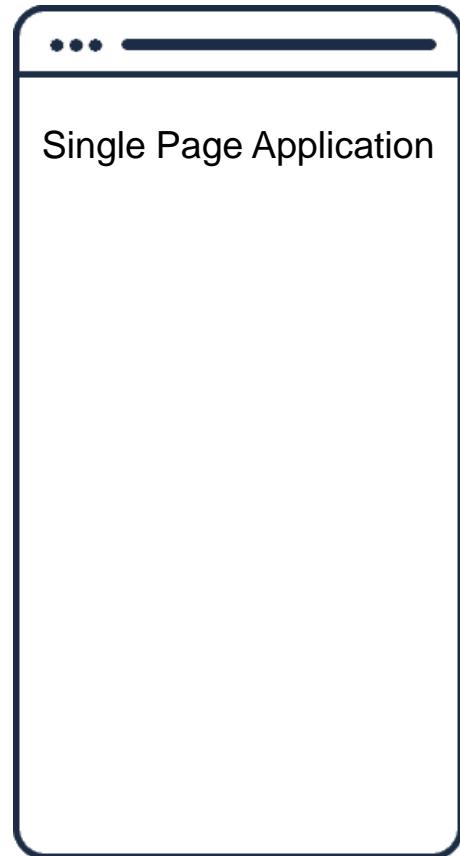
→ ¿Qué es Angular?

SPA vs Aplicación web tradicional



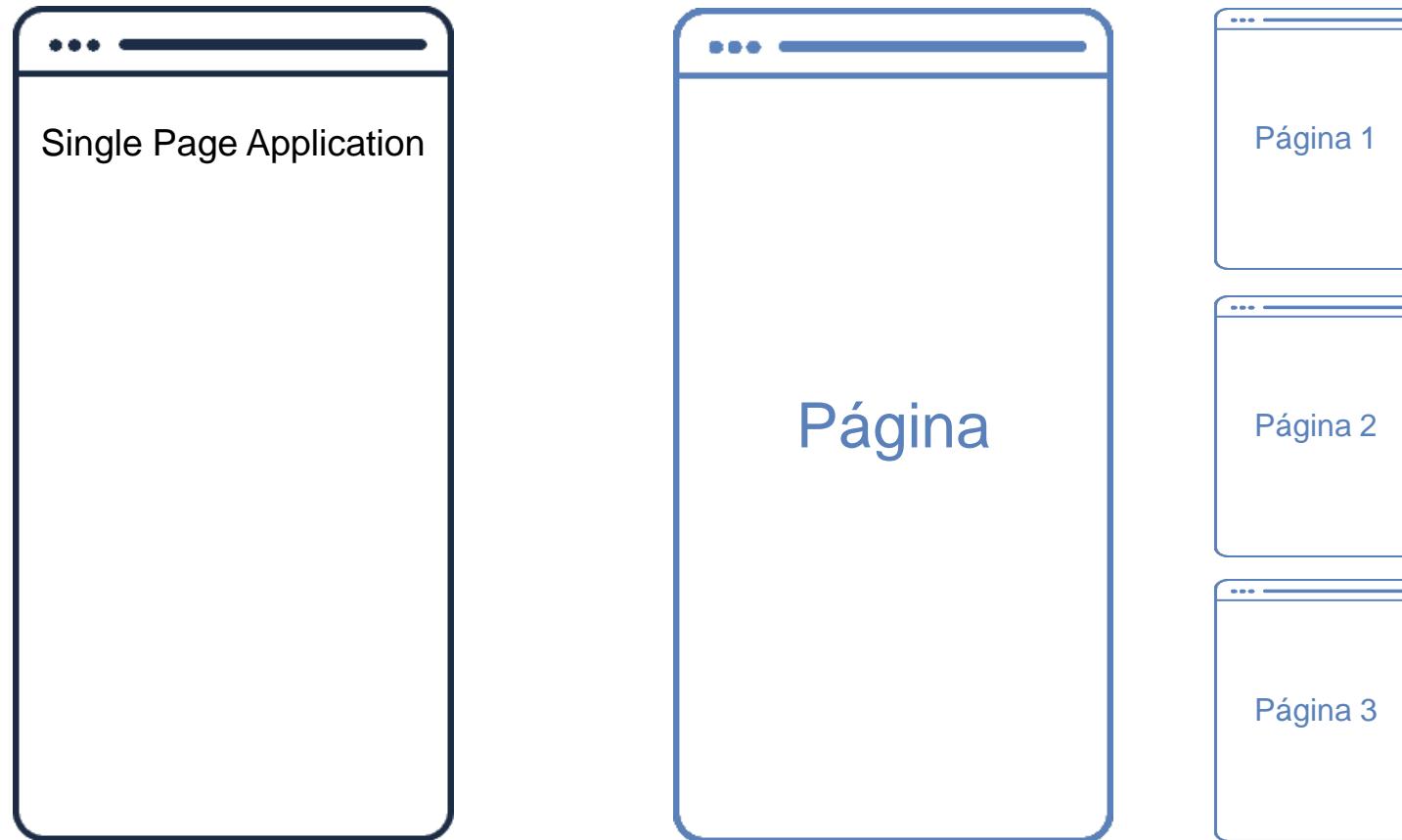
→ ¿Qué es Angular?

SPA vs Aplicación web tradicional



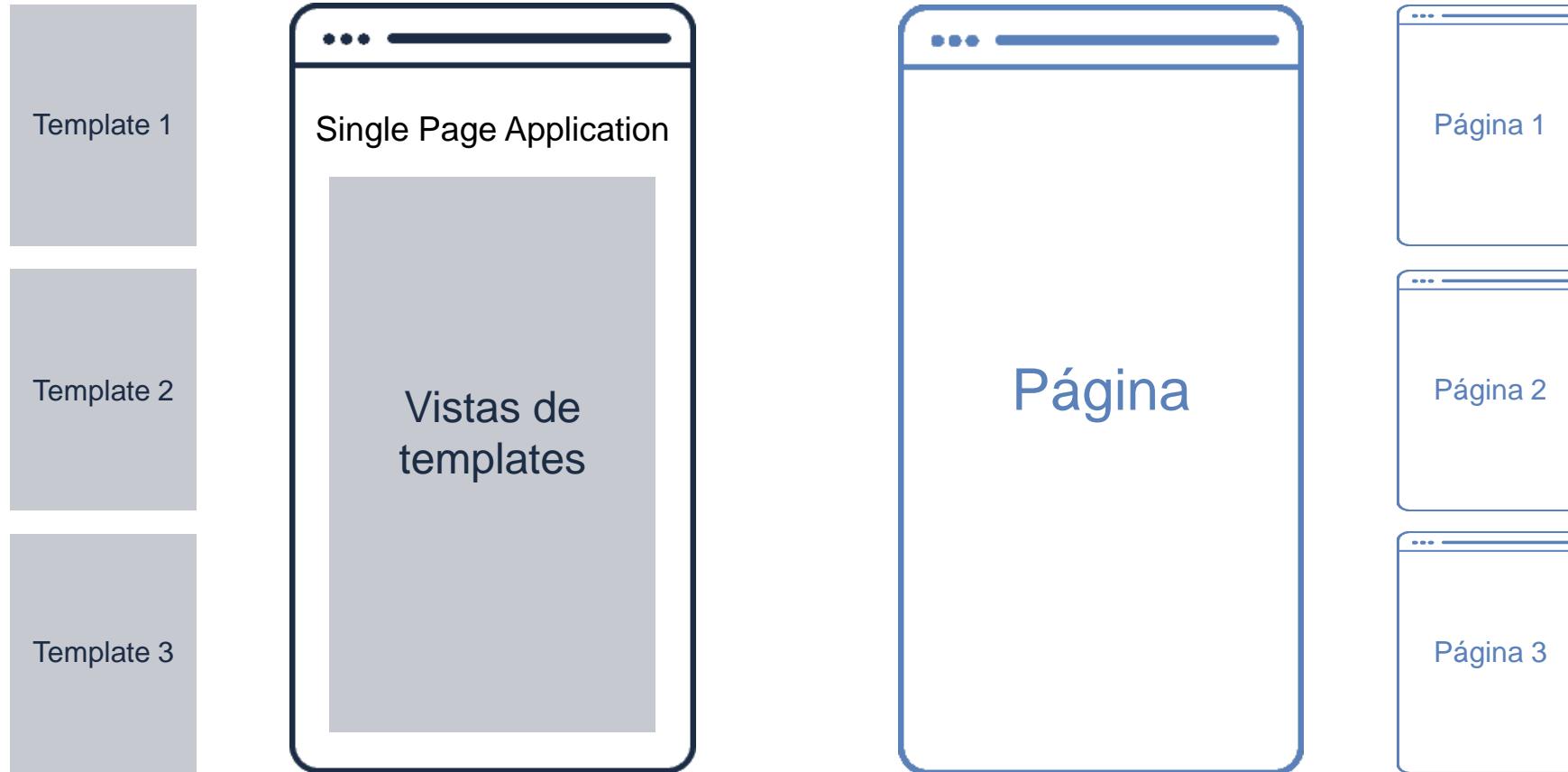
→ ¿Qué es Angular?

SPA vs Aplicación web tradicional



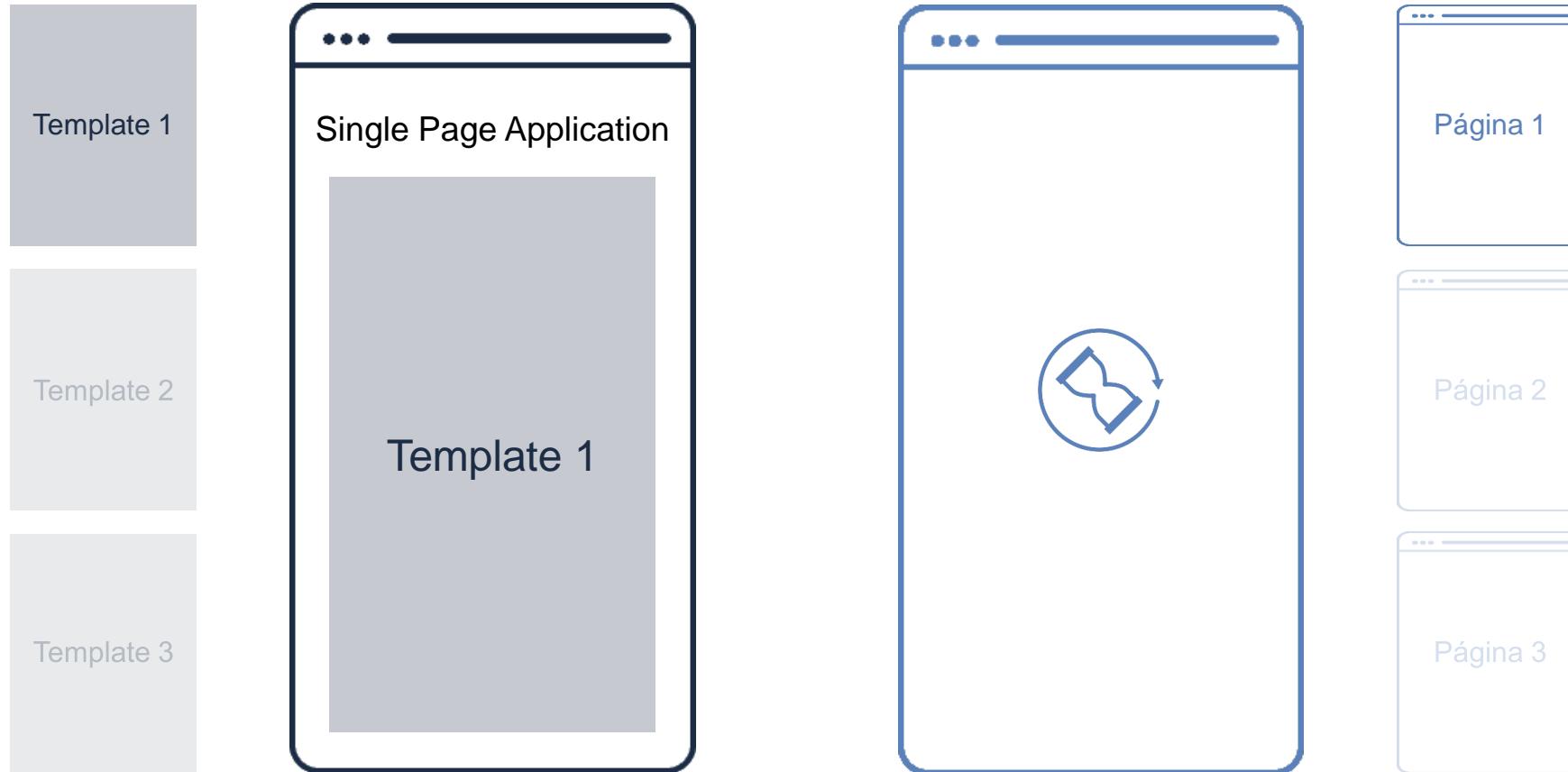
→ ¿Qué es Angular?

SPA vs Aplicación web tradicional



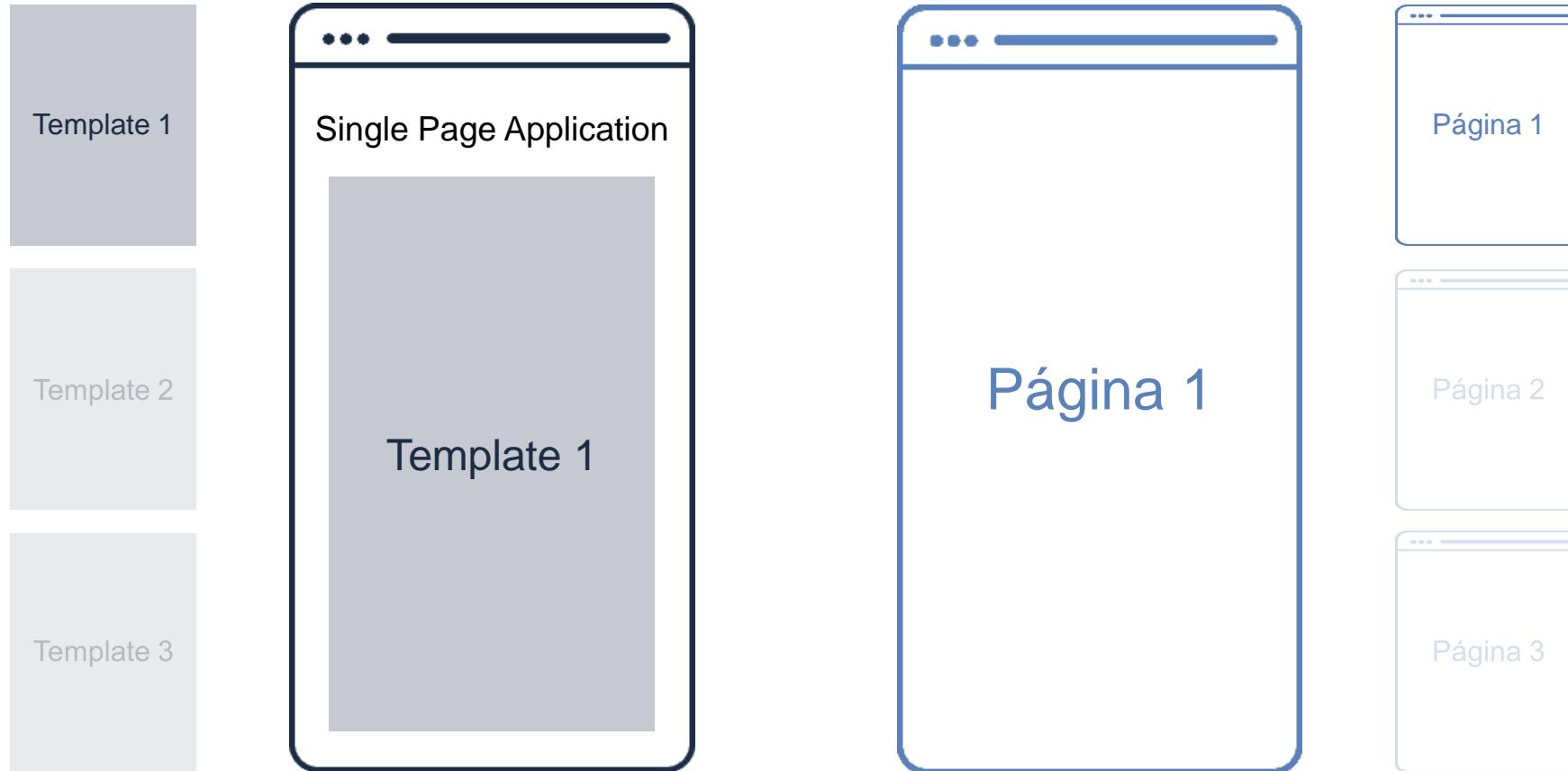
→ ¿Qué es Angular?

SPA vs Aplicación web tradicional



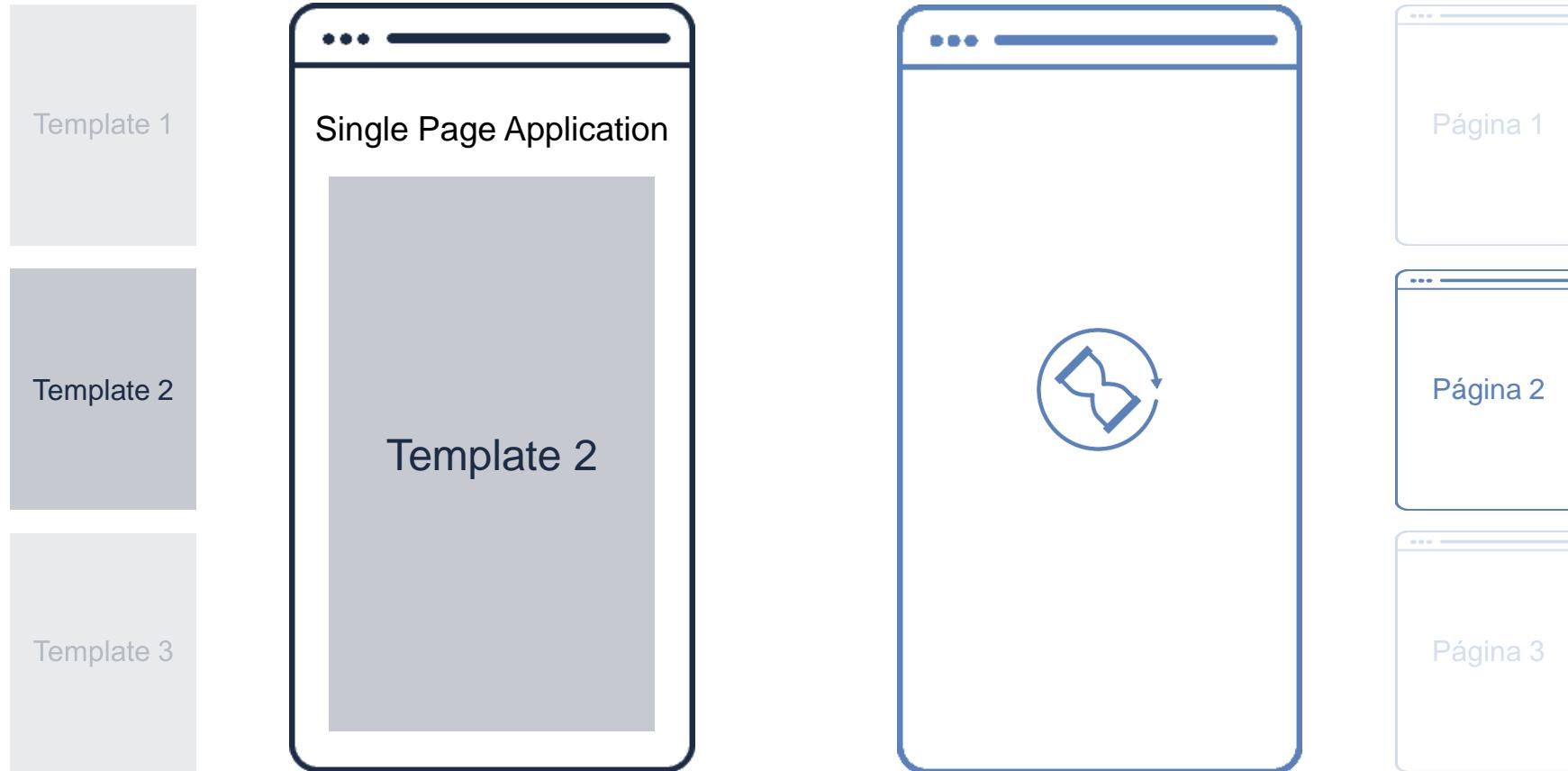
→ ¿Qué es Angular?

SPA vs Aplicación web tradicional



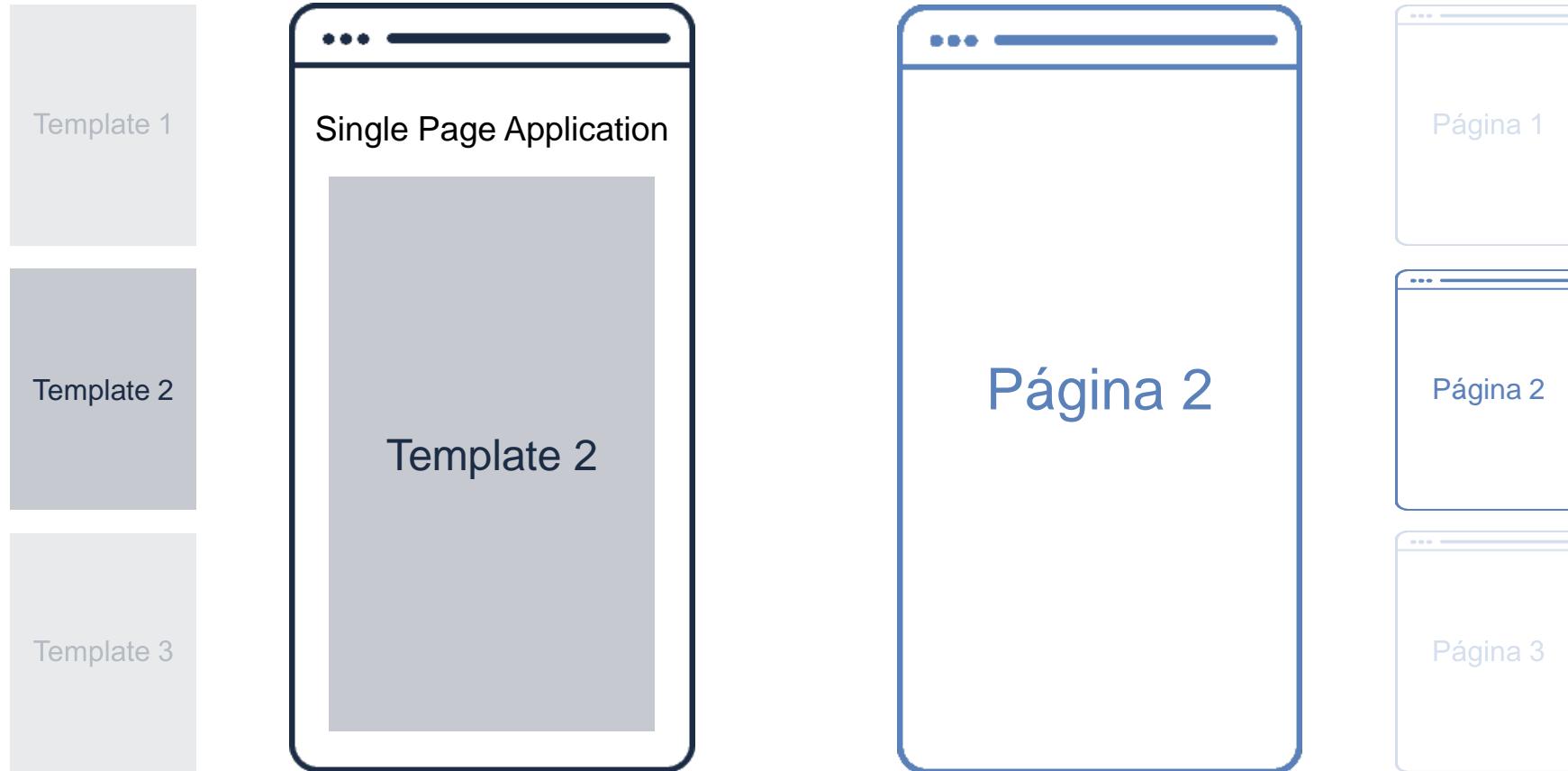
→ ¿Qué es Angular?

SPA vs Aplicación web tradicional



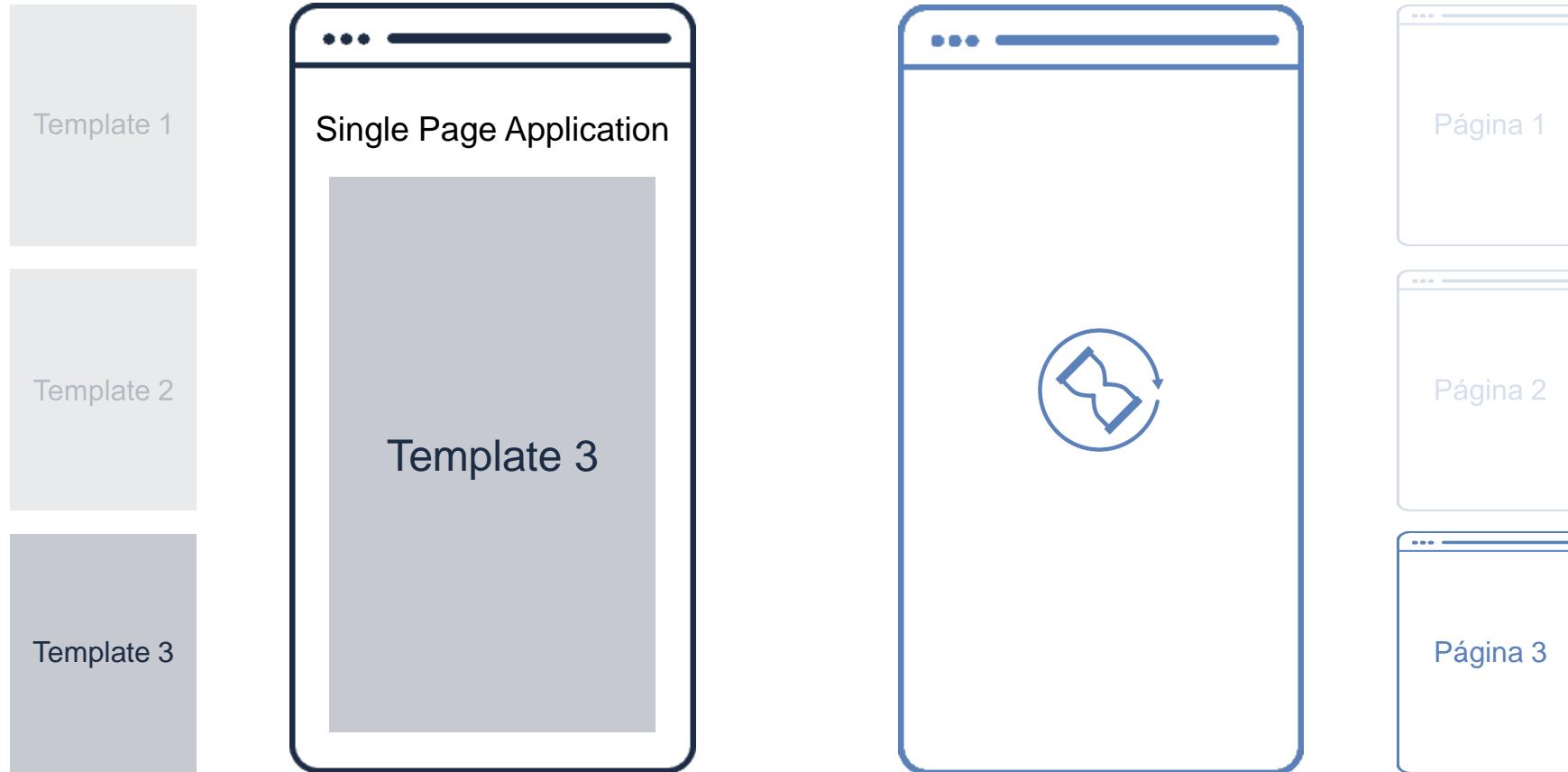
→ ¿Qué es Angular?

SPA vs Aplicación web tradicional



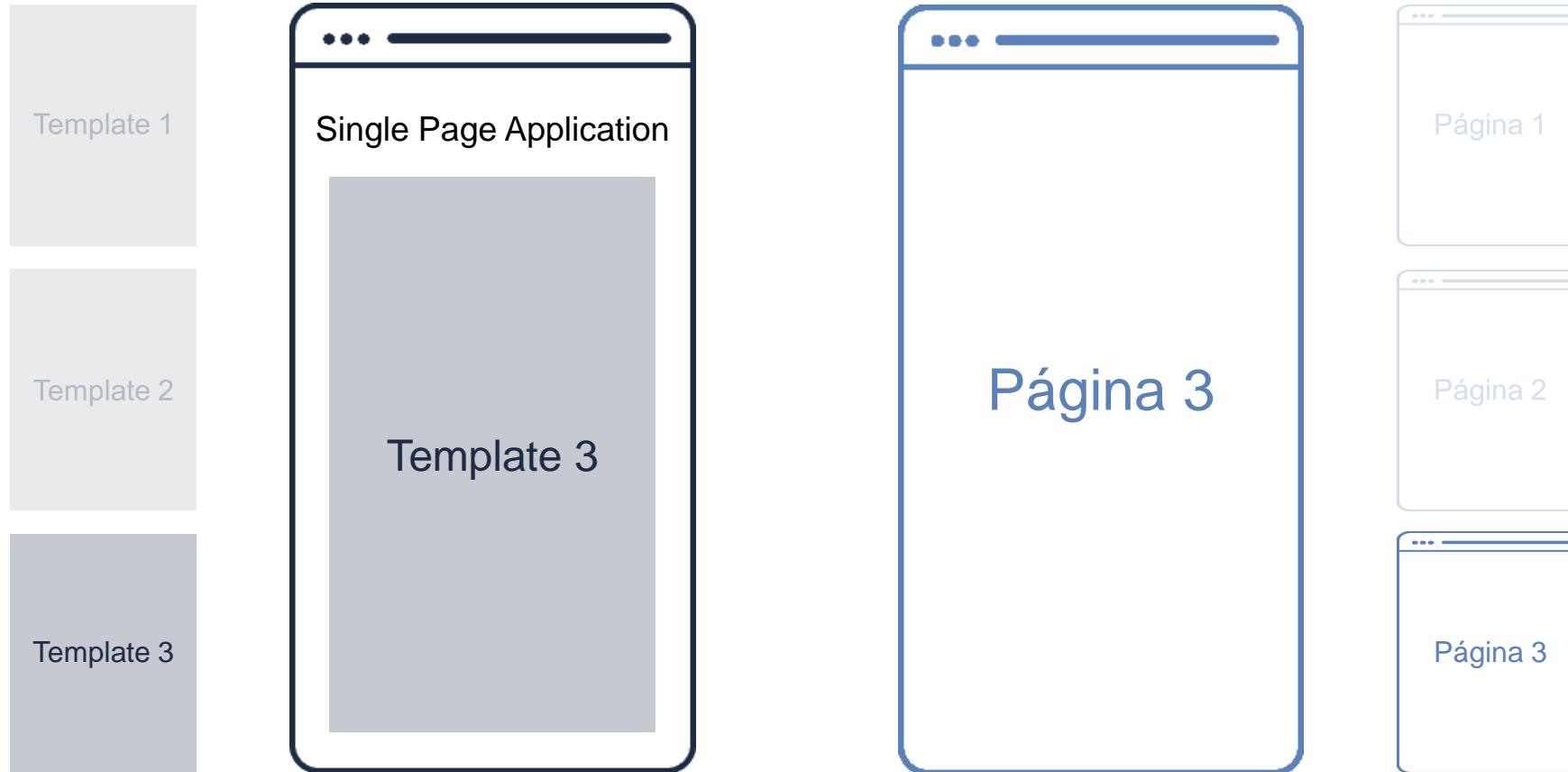
→ ¿Qué es Angular?

SPA vs Aplicación web tradicional



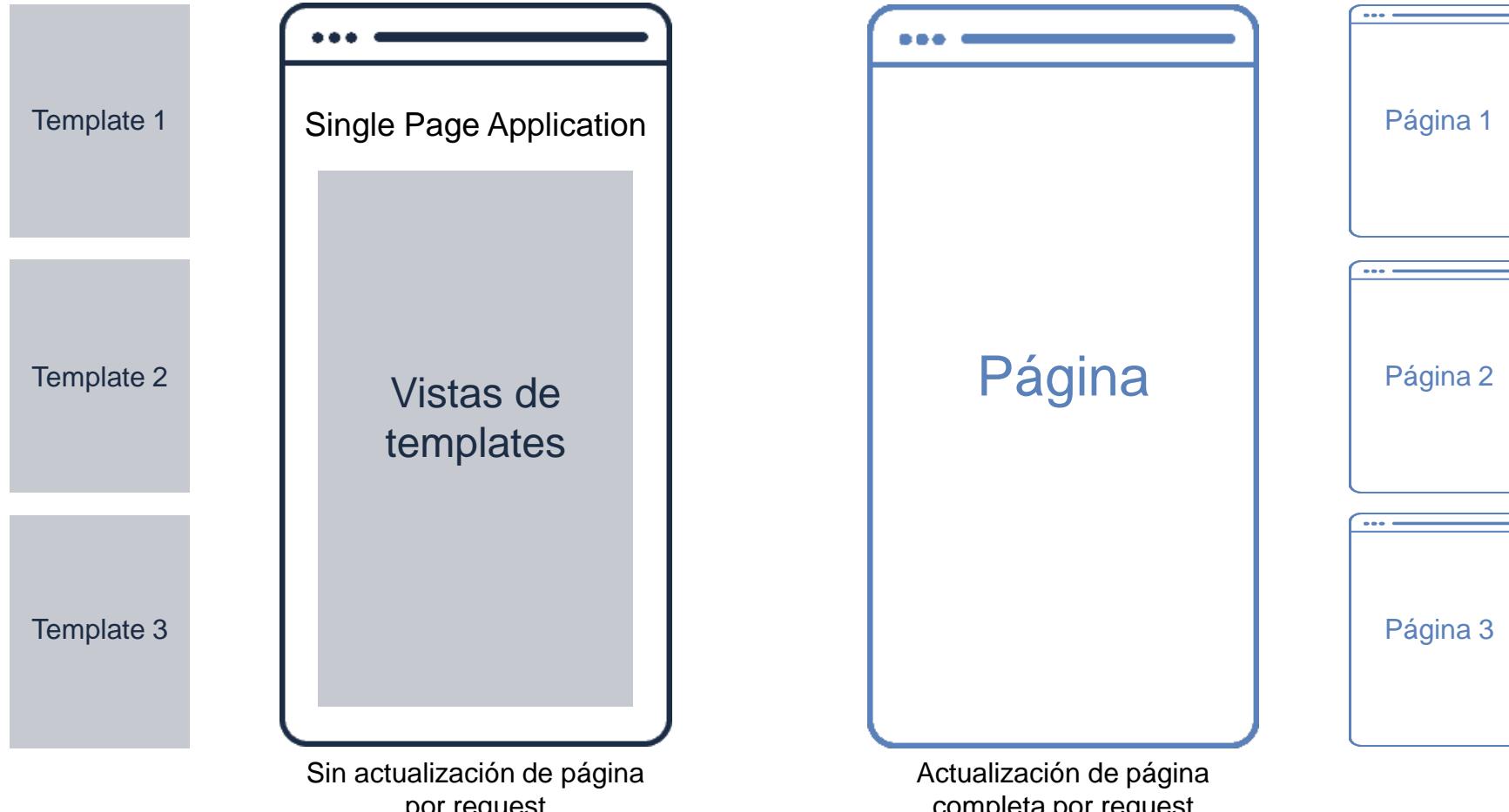
→ ¿Qué es Angular?

SPA vs Aplicación web tradicional



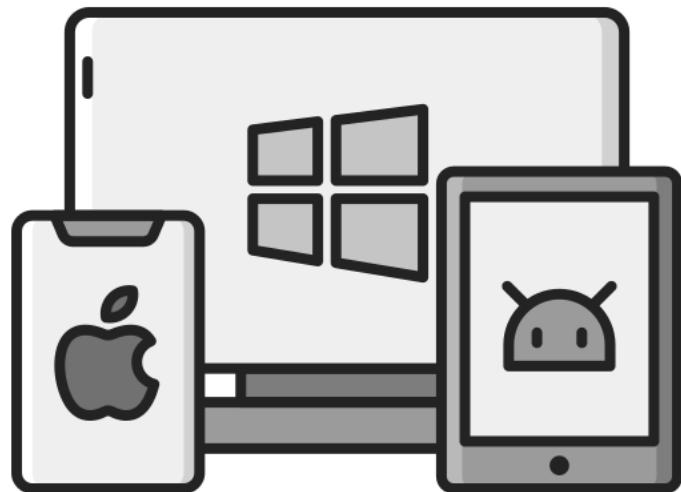
→ ¿Qué es Angular?

SPA vs Aplicación web tradicional



→ ¿Qué es Angular?

¿Qué es una aplicación multiplataforma?



- Una aplicación multiplataforma es una pieza de **software que puede ejecutarse en una variedad de arquitecturas de computadoras y sistemas operativos.**
- Las actualizaciones son simples y convenientes en aplicaciones multiplataforma, ya que están basadas en Internet. Aquí, la aplicación se actualiza automáticamente para todos los usuarios para garantizar que siempre tengan la versión más actualizada, lo que eventualmente aumenta el rendimiento general.

→ ¿Qué es Angular?

Multiplataforma



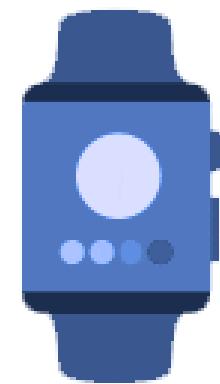
Desktop



Tablet



Mobile



Smart Watch

→ ¿Qué es Angular?

Renderizado del lado del cliente (CSR)

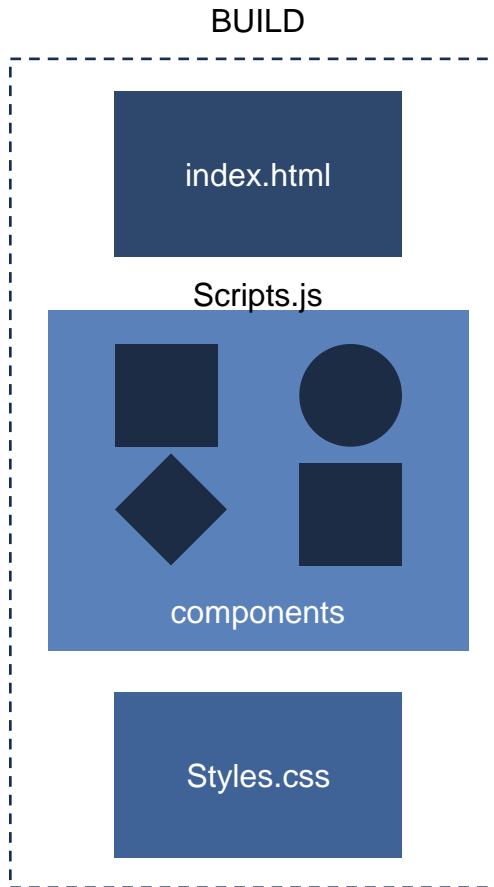
Esto es lo que la mayoría de los frameworks y librerías web como **Angular** y React soportan de forma inmediata. Esto suele ser adecuado para aplicaciones de una sola página (SPA) y aplicaciones con muchas interacciones de usuario (por ejemplo, juegos) y contenido altamente dinámico, como formularios y aplicaciones de chat.

Básicamente, el navegador web cargará inicialmente un archivo HTML vacío. Javascript y los estilos que se cargan después, son responsables de representar la página completa y fácil de usar en el navegador web.



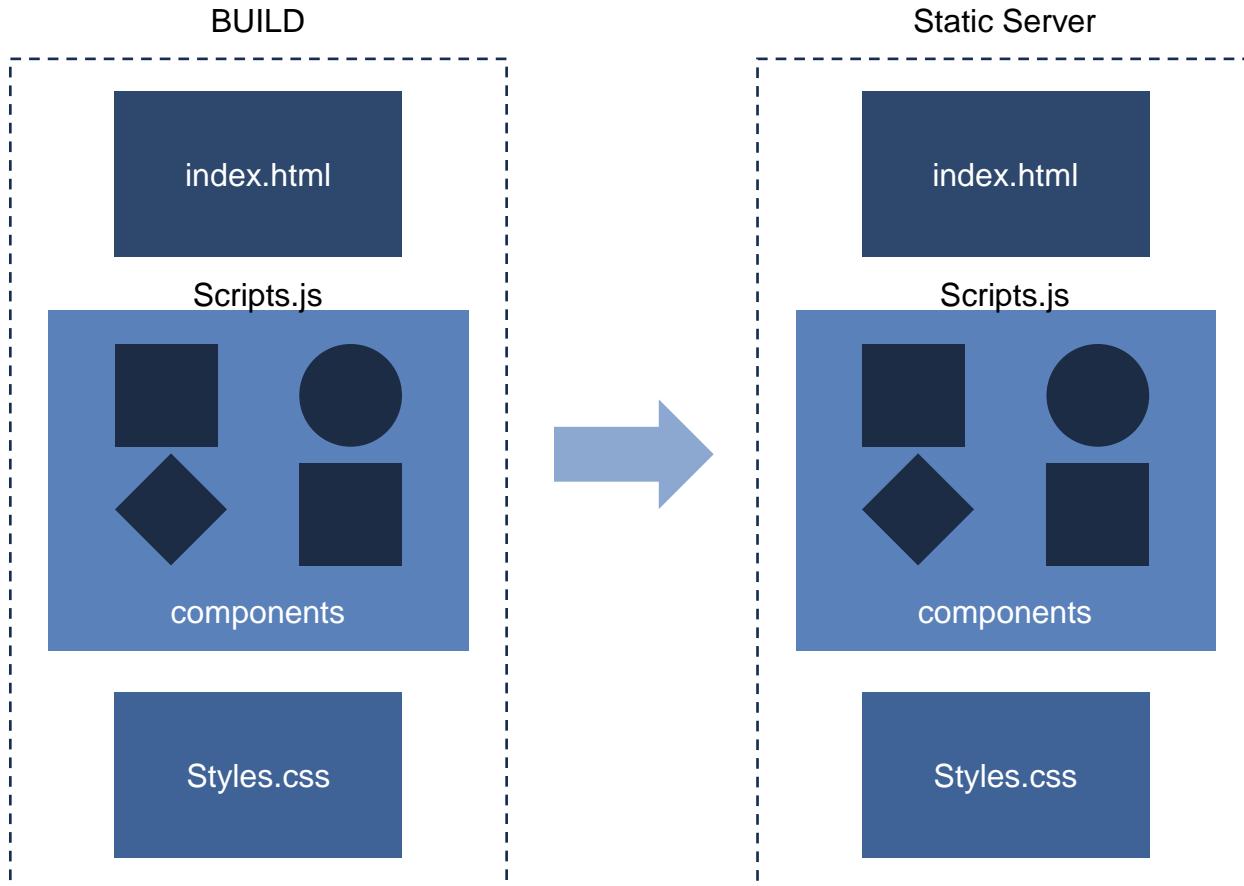
→ ¿Qué es Angular?

Renderizado del lado del cliente (CSR)



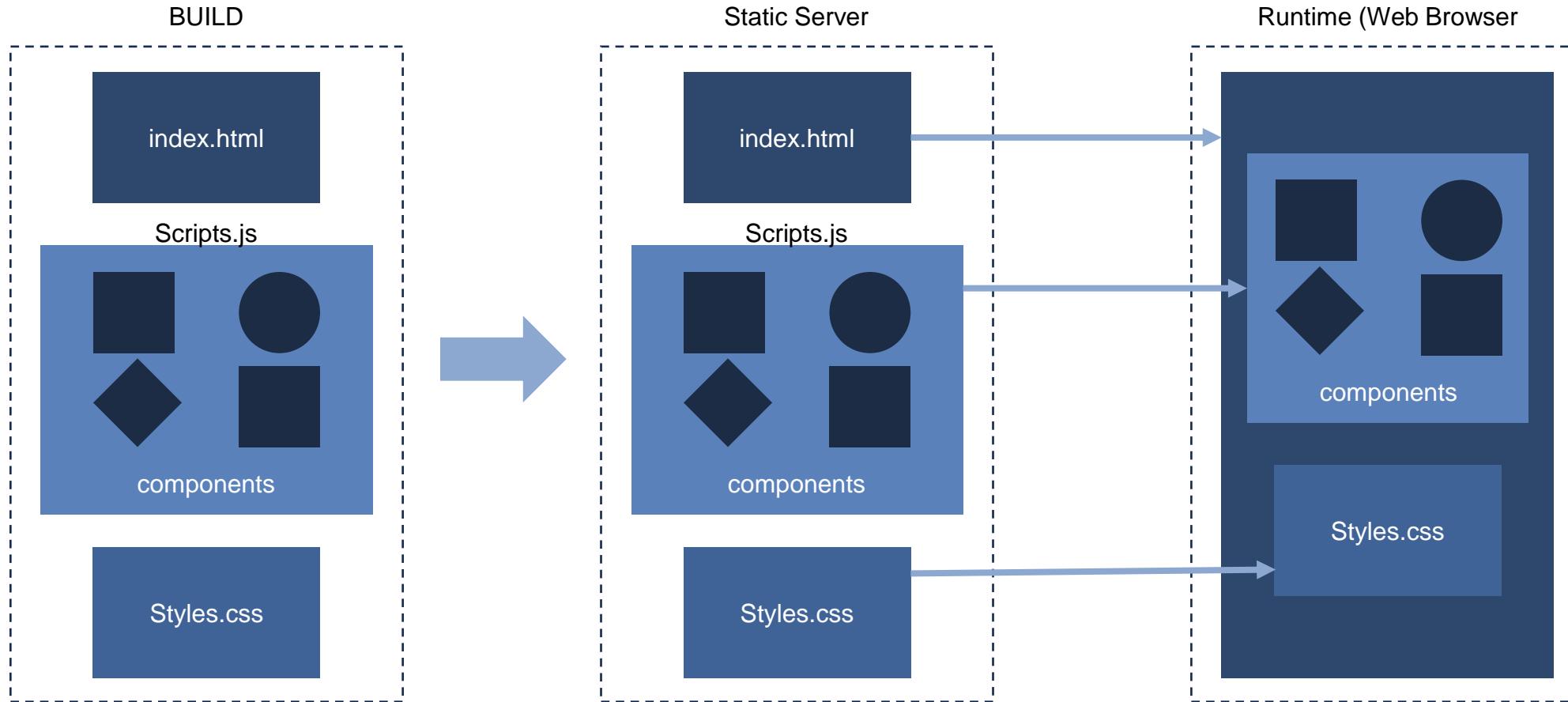
→ ¿Qué es Angular?

Renderizado del lado del cliente (CSR)



→ ¿Qué es Angular?

Renderizado del lado del cliente (CSR)



→ ¿Qué es Angular?

Renderizado del lado del servidor (SSR)

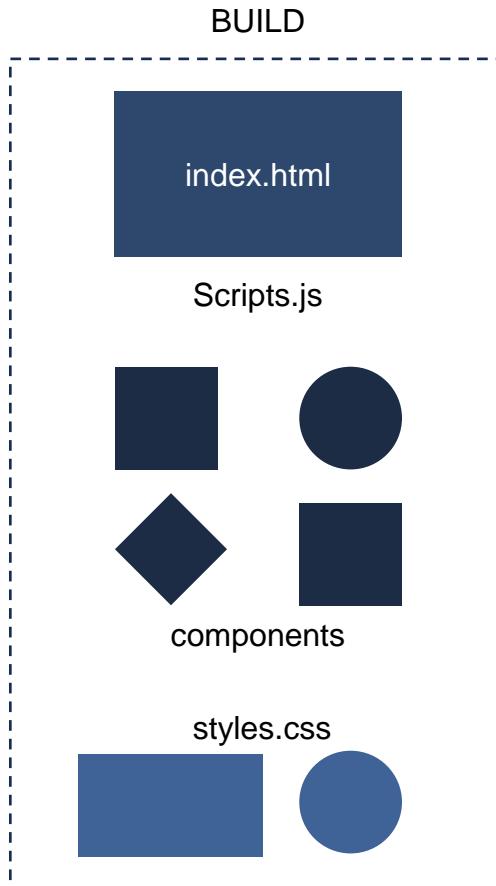
La principal desventaja de la SSR es que no está optimizada para motores de búsqueda. Por lo tanto, la mayoría de los frameworks web también brindan la capacidad de representar las páginas en el servidor.

A diferencia de CSR, cada página iniciará una solicitud al servidor de aplicaciones, donde se representa dinámicamente y se sirve HTML completo para esa página. Dado que solicita y representa la página cada vez que el usuario lo solicita, el tiempo de servicio de la página es más que una aplicación CSR.



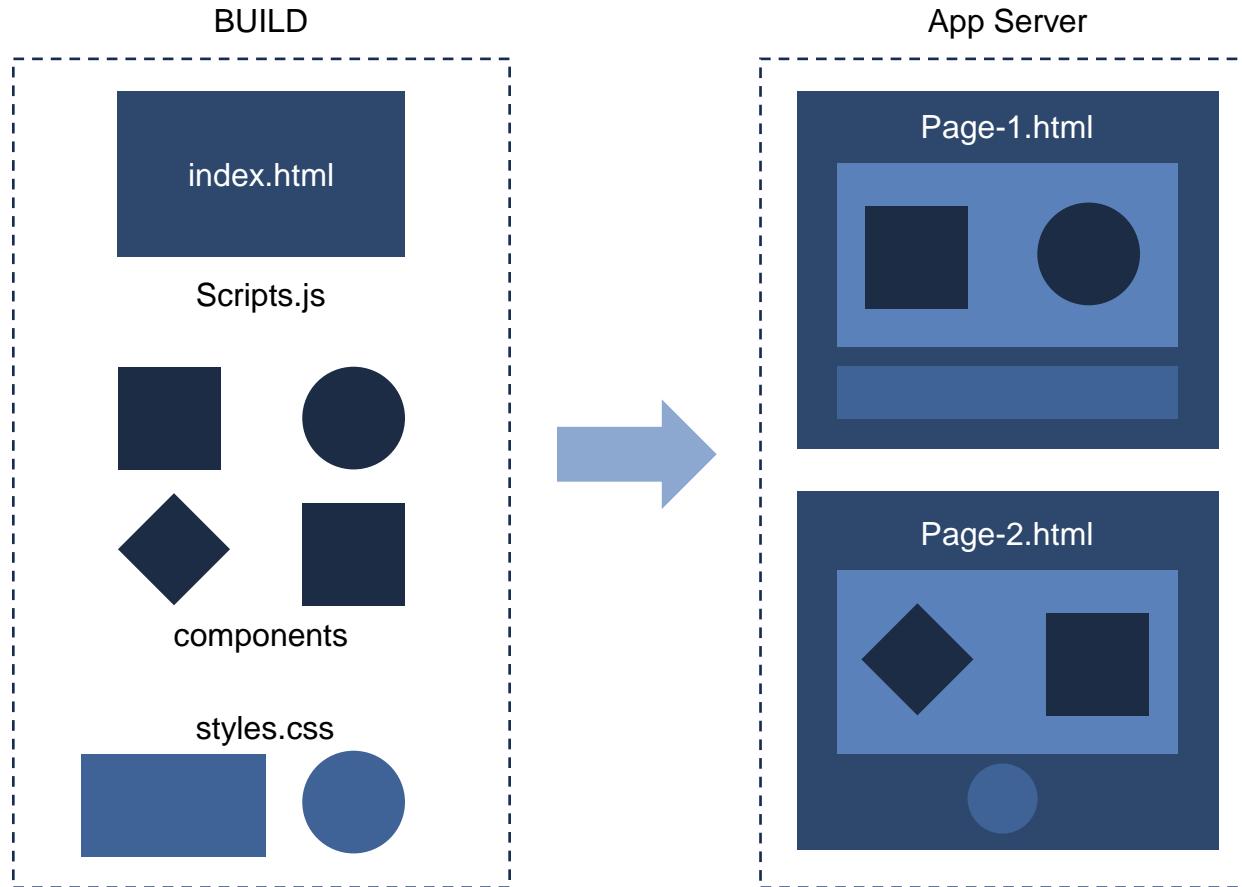
→ ¿Qué es Angular?

Renderizado del lado del servidor (SSR)



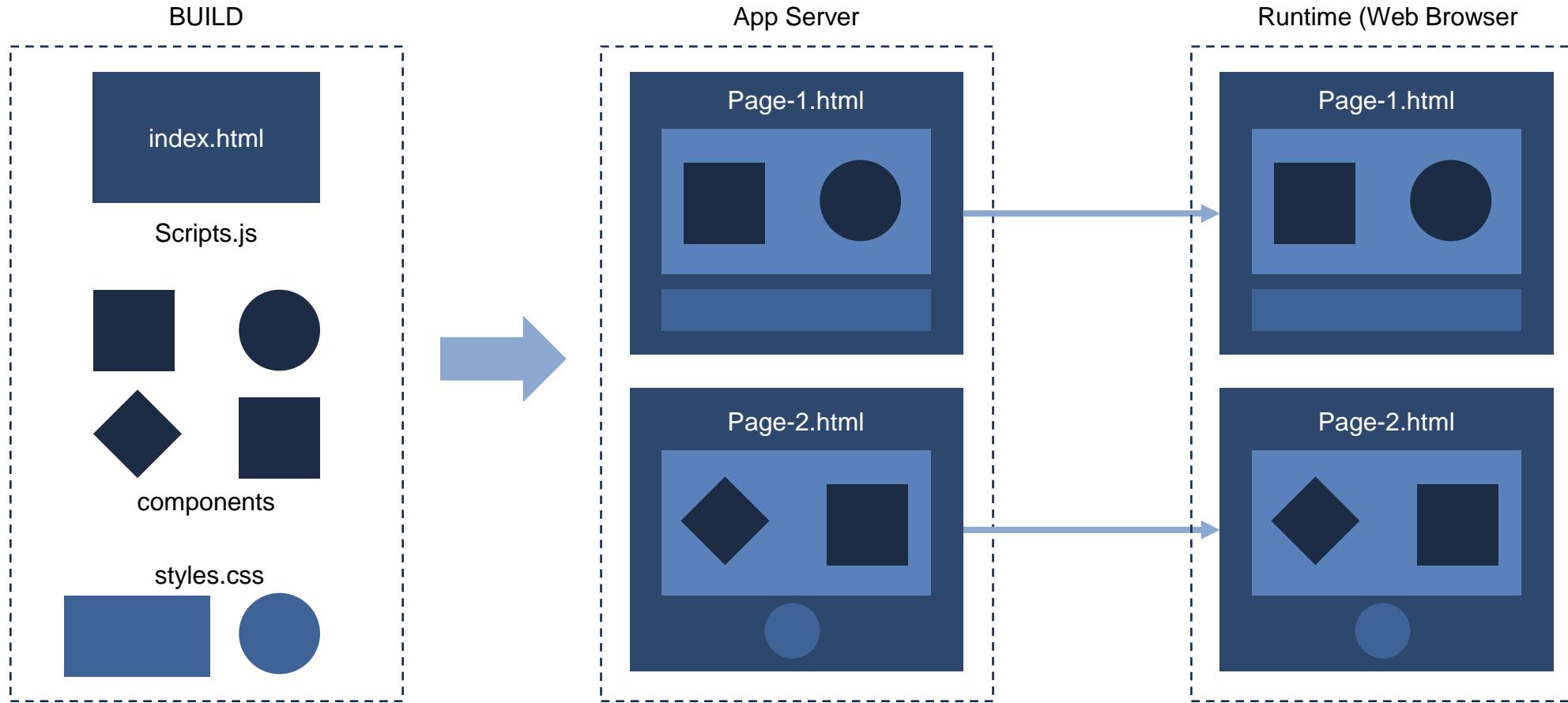
→ ¿Qué es Angular?

Renderizado del lado del servidor (SSR)



→ ¿Qué es Angular?

Renderizado del lado del servidor (SSR)



→ ¿Qué es Angular?

Generación de sitios estáticos (SSG)

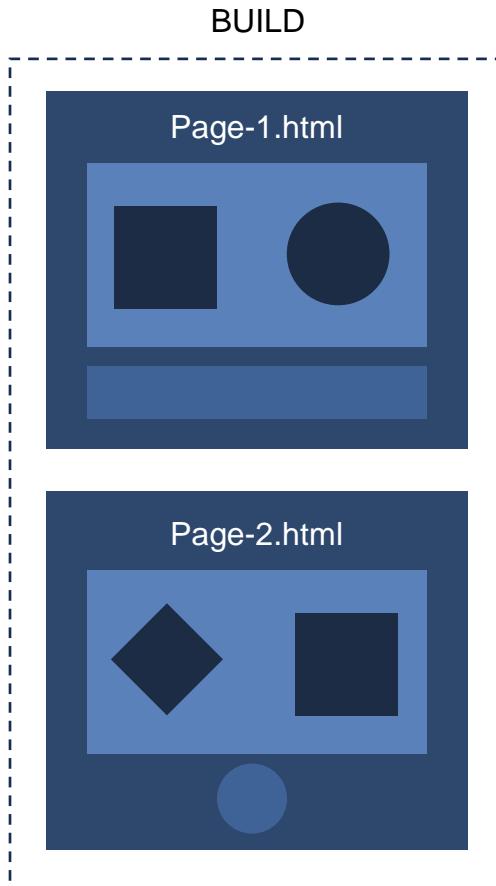
Para evitar la compilación en cada solicitud, ¿por qué no generamos esos archivos en el tiempo de compilación, de modo que podamos servir instantáneamente las páginas cuando el usuario lo solicite?

Esta técnica te será útil si quieras crear una aplicación web llena de contenido estático como un blog. Un inconveniente es que el contenido puede estar desactualizado y es necesario compilar e implementar la aplicación cada vez que se cambia el contenido.



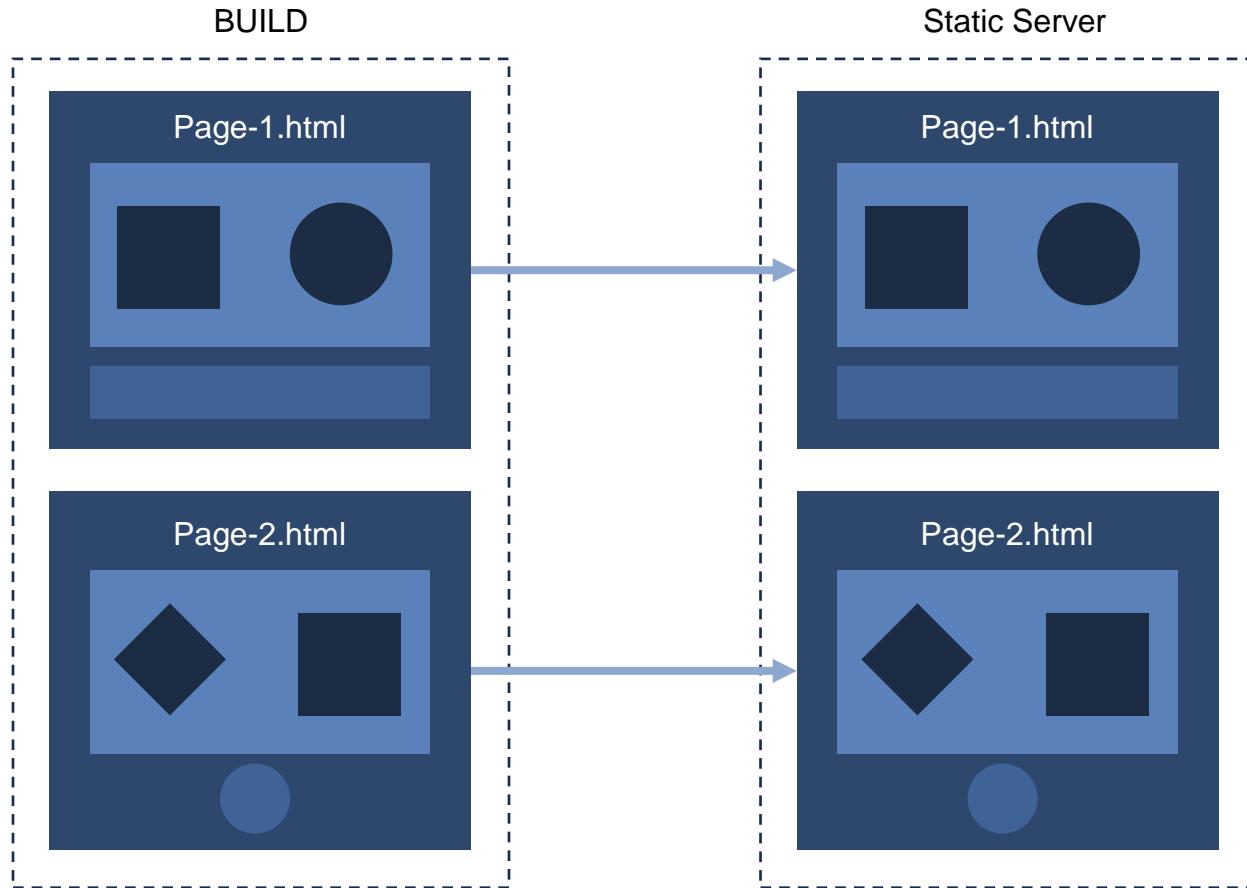
→ ¿Qué es Angular?

Generación de sitios estáticos (SSG)



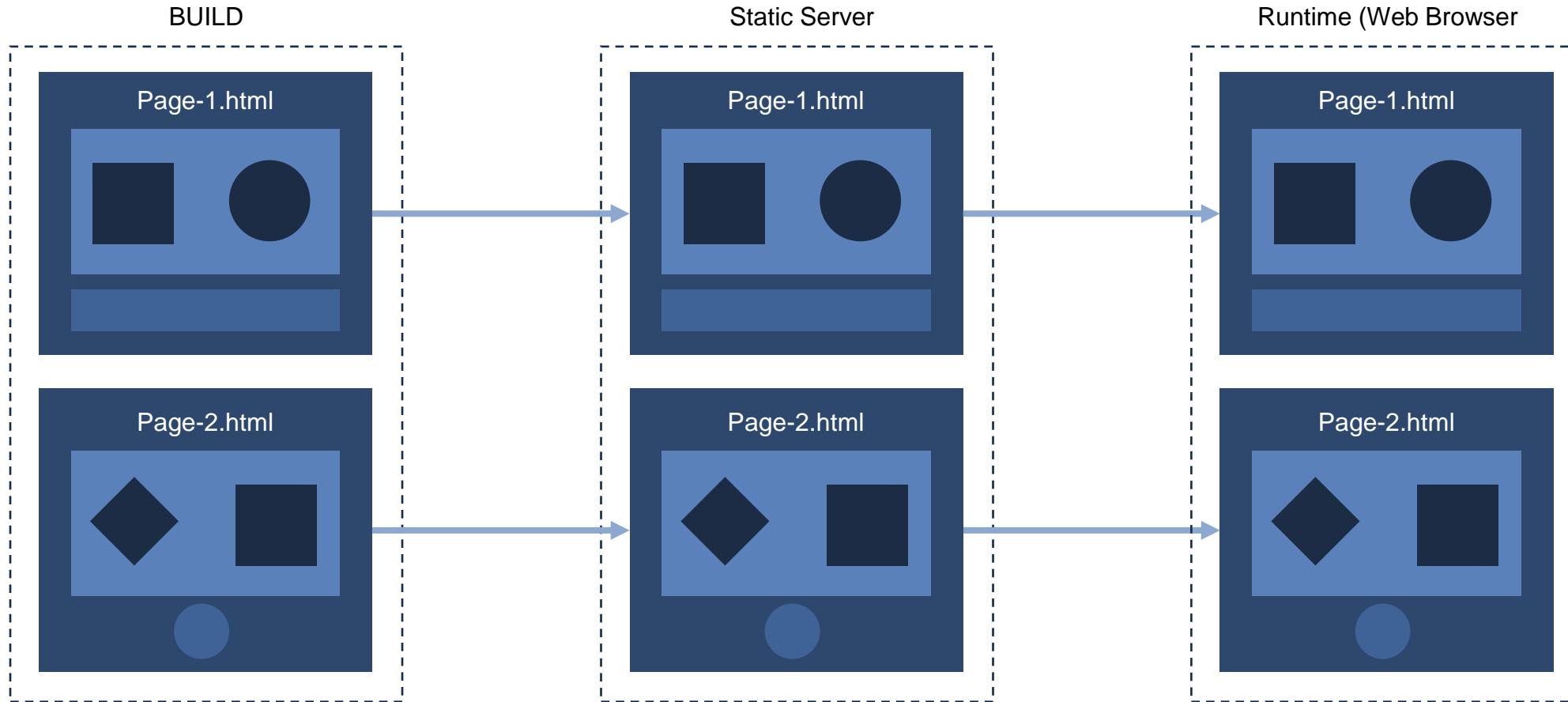
→ ¿Qué es Angular?

Generación de sitios estáticos (SSG)



→ ¿Qué es Angular?

Generación de sitios estáticos (SSG)



→ ¿Qué es Angular?

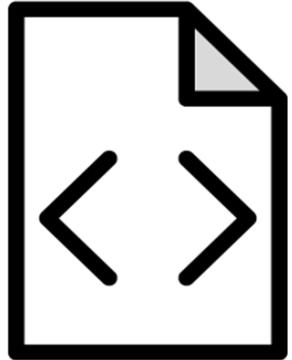


Angular

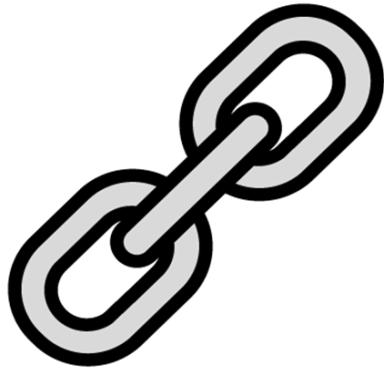
- Sucesor de AngularJS
(Angular 1.x)
- **Framework Javascript Open Source**
- **Creado por Google para sus propios proyectos**
- **Ofrece una estructura base para aplicaciones.**
- Aplicaciones Enriquecidas del lado del cliente
(Aplicaciones SPA)
- Utiliza TypeScript
(Google construyó Angular con TypeScript)

→ ¿Qué es Angular?

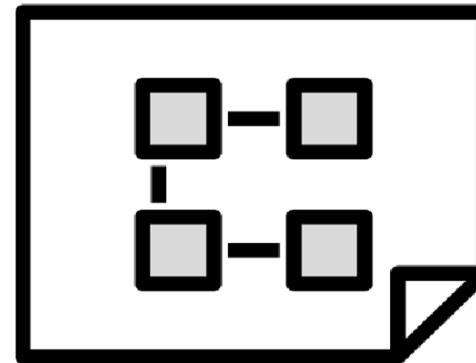
¿Por qué Angular?



HTML
Expresivo



Poderoso
enlace de datos



Diseño modular



Fácil integración
con Back-End

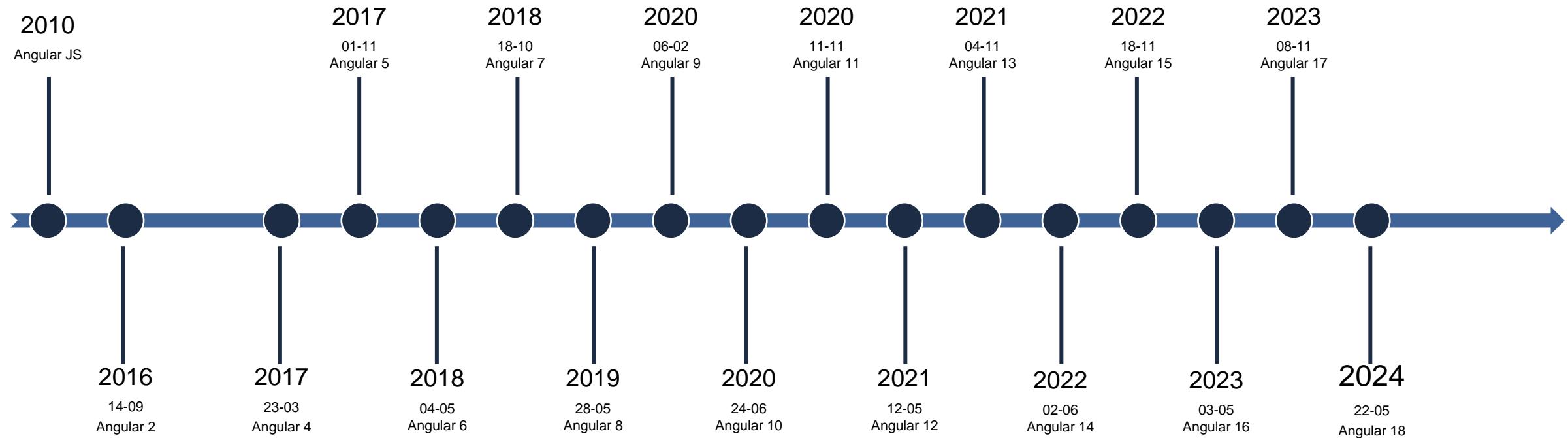
02



Evolución y ventajas de Angular.

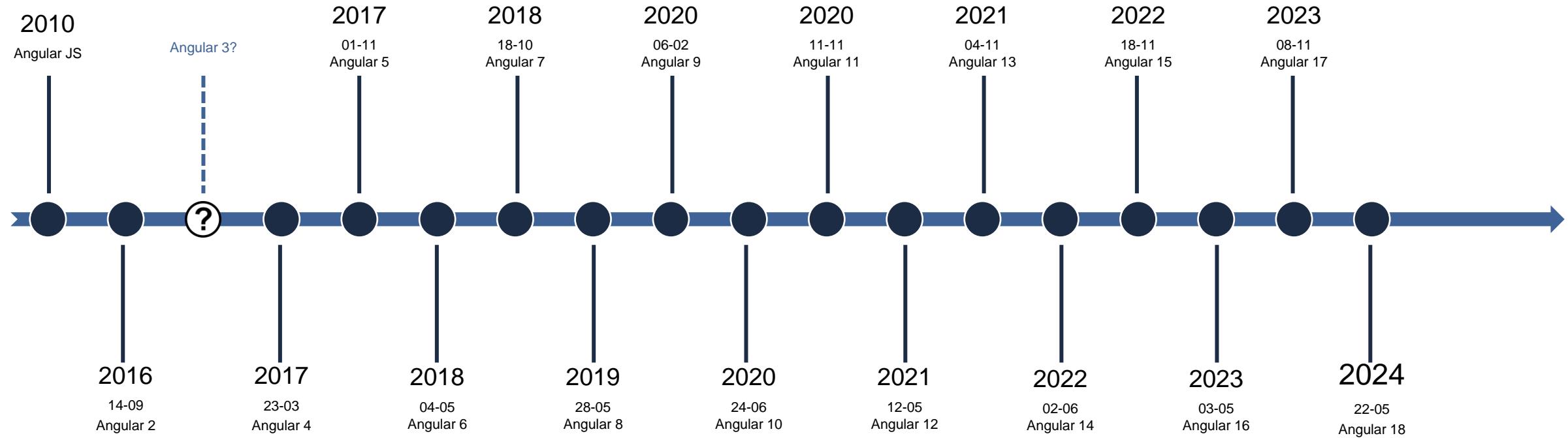
→ Evolución y ventajas de Angular

Versiones de Angular



→ Evolución y ventajas de Angular

Versiones de Angular



→ Evolución y ventajas de Angular

Versiones de Angular

ANGULARJS 1.X .- Es un marco de desarrollo de aplicaciones front-end web de código abierto basado en JavaScript. Es **apoyado principalmente por Google y una comunidad** de individuos/organizaciones. El objetivo principal de AngularJS 1.X es facilitar el desarrollo y las pruebas de aplicaciones al **proporcionar un marco para las arquitecturas modelo-vista-controlador (MVC) y modelo-vista-vista (MVVM)** del lado del cliente, así como componentes que se encuentran comúnmente en aplicaciones en línea enriquecidas. AngularJS está ahora en modo de soporte a largo plazo (LTS), siendo la versión 1.8.x la versión principal.

ANGULAR 2 .- Es una **versión renovada de AngularJS**, que es **reescrita desde cero** por el equipo de Angular **usando TypeScript**. La razón principal de su reconstrucción fue acomodar dispositivos móviles sin esfuerzo. Aparte de esto, Angular 2 mejora el número de idiomas disponibles al agregar características y funcionalidades adicionales. Es posible que se requieran lenguajes como ES5, ES6, TypeScript y Dart para escribir código Angular 2. A pesar de ser un rediseño completo, **Angular 2 conserva muchas de las características y prácticas de AngularJS 1.X**, como una versión optimizada y "nativa JS" de la DI.

→ Evolución y ventajas de Angular

Versiones de Angular

ANGULAR 3 .- Es posible que haya tenido problemas para encontrar información sobre la fecha de lanzamiento de Angular 3. Esto se debe a que **Angular 3 se dejó de lado** principalmente debido a una **discrepancia de versión entre las bibliotecas @angular/core, @angular/compiler y @angular/router**.

Para la versión de Angular 2, el core y el router se ven así:

- v2.3.0 of @angular/core
- v2.3.0 of the @angular/compiler
- v2.3.0 of @angular/compiler-cli
- @angular/http v2.3.0 @angular/http v2.3.0 @angular/http
- @angular/router v3.3.0 is now available.

Sin embargo, **el problema** al que se enfrentó fue con el **@angular/router**, que ya estaba en una versión 3.X. Esto se debe a ciertos avances continuos y significativos en el sector de los enrutadores, como la precarga de rutas. Es probable que el lanzamiento de **Angular 3 con una hoja de ruta hacia la versión 4 cause mucha incertidumbre**. Para evitar esta incertidumbre, los desarrolladores han **decidido omitir la versión 3 y en su lugar lanzar la versión 4.0.0**, lo que garantiza que todas las dependencias principales estén actualizadas.



→ Evolución y ventajas de Angular

Versiones de Angular

ANGULAR 4 .- Es un marco de desarrollo de aplicaciones web que es compatible con angular 2.x.x que le permite crear aplicaciones JavaScript. Como resultado, no hay mucha diferencia entre Angular 4 y Angular 2. Podemos, sin embargo, afirmar categóricamente que Angular 4 no es una reconstrucción de Angular 2. Hay nuevas modificaciones bajo el capó que tienden a reducir el tamaño del código generado para sus componentes en aproximadamente un 60% en la mayoría de las circunstancias, así como **alertas mejoradas de corrección de errores**. Esto dio como resultado una **mayor compatibilidad para los navegadores más nuevos**, así como mejoras significativas en el marco.

Angular 4 también incluyó un **nuevo entorno de prueba que redujo el tiempo de prueba a la mitad** y admitió la compilación JIT (Just-In-Time) para minimizar aún más los tamaños de los paquetes.

En general, fue una versión exitosa con muy pocos defectos que causaron problemas para los desarrolladores de Angular 4. Sin embargo, migrar de una versión a otra puede ser difícil cuando se utilizan ciertas herramientas o bibliotecas de terceros, por lo que es crucial planificar con anticipación cualquier cambio que pueda surgir antes de actualizar su proyecto o aplicación.

→ Evolución y ventajas de Angular

Versiones de Angular

Algunos cambios con respecto al angular 2:

- Las **animaciones se extraen de @angular/core** para reducir la cantidad de código cargado en nuestro paquete de producción.
- Simplemente puede agregar animación a NgModule importando **BrowserAnimationsModule desde @angular/platform-browser/animations**.
- **Renderer 2 se utiliza en lugar de Renderer** desde el mismo '@angular/core'.
- ***ngIf/then** Ahora también puede utilizar la cláusula **else**.

- En Angular 4, no es necesario crear un patrón para la validación de correo electrónico.
- Compatibilidad con TypeScript 2.1 y 2.2:
- Podemos usar typescript 2.1 o superior; anteriormente, solo se admitía typescript 1.8.

→ Evolución y ventajas de Angular

Versiones de Angular

ANGULAR 5 .- Es un hito importante en su evolución porque marcó un cambio importante en la forma en que estaban posicionando. **En sus primeras etapas, se promocionó principalmente como una plataforma de aplicaciones móviles que podría usarse para construir aplicaciones complejas de una sola página.** Sin embargo, Angular 5 ha cambiado su enfoque para incluir aplicaciones web de escritorio.

De forma predeterminada, las compilaciones de producción que se realizaron con **la CLI ahora usa el optimizador de compilación**. Se introdujeron **nuevas características, como el soporte de DOM** y Angular Universal StateTransfer API, mejoras en el compilador, pipes para números internacionalizados, fechas y moneda. **Los formularios se han actualizado para Blur/Submit** y también proporcionan soporte para RxJS 5.5.

→ Evolución y ventajas de Angular

Versiones de Angular

ANGULAR 6 .- El CLI viene con dos **nuevos comandos: ng update y ng add**. Proporciona funcionalidad incorporada para servicios HTTP, animación y **Angular Material** que a su vez proporcionan características de autocompletado, barra de herramientas, menús, navegación, etc.

- **Kit de desarrollo de componentes para elementos de angular (CDK)**
- Componentes de arranque para **Angular Material**
- Proveedores de Tree Shakable
- Mejoras en el rendimiento
- RxJS v6

ANGULAR 7 .- Ahora está disponible con mejoras de velocidad y nuevas características como CLI Prompts, **Virtual Scrolling** y **Drag & Drop**. Aparte de esto. Algunas de las características importantes de Angular 7, Angular CLI 7 y Material 7. Se usan comandos simples como **ng new o ng add @angular/material**, la CLI ahora solicitará a los usuarios que exploren las capacidades integradas, como el enrutamiento y el soporte de SCSS.

Angular tiene una mayor flexibilidad para proporcionar servicios de desarrollo de aplicaciones web impecables para empresas de diversa magnitud.



→ Evolución y ventajas de Angular

Versiones de Angular

ANGULAR 8 .- Es extremadamente similar a las versiones anteriores. **Carga diferencial de forma predeterminada**: un procedimiento en el que el navegador decide si cargar JavaScript actual o heredado en función de sus propias capacidades, **importaciones para configuraciones de ruta que son dinámicas, API de Builder en CLI, soporte de Web Worker**, etc.

- Tiempo de rebuild más rápido.
- Tamaño de payload mejorado.
- Comprobación de tipo mejorada.
- Retrocompatibilidad.
- El código que se genera es más fácil de leer.

ANGULAR 9 .- De forma predeterminada, todas y cada una de las aplicaciones de la versión **9 utilizan el compilador y el runtime de Ivy**, y tienen las siguientes ventajas:

- Reducción del tamaño del paquete
- Finalización más rápida de la prueba
- Depuración mejorada
- Enlace de clase y estilo mejorado en CSS
- Comprobación de tipos mejorada
- Reducción de los errores de compilación
- Default AOTs (Angular ahead-of-time).
- Internacionalización mejorada

→ Evolución y ventajas de Angular

Versiones de Angular

Angular Material incluye:

- Componente del reproductor de YouTube
- Componente para Google Maps

Angular puede admitir TypeScript 3.6 y 3.7 con facilidad. La **base subyacente de todo el marco ha cambiado en Angular 9**. Las aplicaciones más pequeñas son las que más se benefician de la **capacidad del tree shaking** de IVY porque los métodos de interacción de componentes angulares requieren menos código. Uno de los principales objetivos de **Angular 9 es hacer que las aplicaciones sean más pequeñas**

ANGULAR 10 .- Algunas de las características que vale la pena mencionar son advertencias sobre las **importaciones de CommonJS**, configuraciones opcionales más estrictas, TypeScript 3.9, TSLib actualizado a v2.0, TSLint actualizado a v6 y una nueva configuración predeterminada del navegador.

Cuando se utiliza una dependencia que está repleta de **CommonJS**, **Se obtendrían resultados en forma de aplicaciones más grandes y lentas**. Con Angular 10, le dará un mensaje de advertencia. Como resultado, el paquete de módulos ECMAScript (ESM) será una buena opción para su próximo proyecto.



→ Evolución y ventajas de Angular

Versiones de Angular

ANGULAR 11

- Soporte actualizado de Hot Module Replacement (HMR) **ng serve -hmr**
- Automatic Inlining of Fonts
- Component Test Harness
- Compilaciones más rápidas
- Soporte experimental de Webpack 5
- Migración a ESLint
- Mejoras en Routers, Forms y Reporting & Logging
- Compatibilidad con TypeScript 4.0
- Soporte de Lazy Loading con Named Outlets

ANGULAR 12

- Migración a Ivy
- Mejora del diseño de identificación de mensajes i18n
- Independización de Protractor.
- Operador coalescente nulo (??)
- Mejoras de estilo
- Obsolescencia para IE11.
- Mejoras en la API de Ng
- Soporte para Typescript 4.2

Compilación de producción predeterminada (ng build ahora producirá compilaciones de producción)



→ Evolución y ventajas de Angular

Versiones de Angular

ANGULAR 13

- View Engine ya no está disponible (Ahora solo admite Ivy como motor de vista)
- Cambios en el Angular Package Format
- Se ha eliminado la compatibilidad con IE 11.
- Compatibilidad con TypeScript 4.4
- Las versiones de Node.js anteriores a la 12.20 ya no son compatibles
- RxJS Versión 7.4
- Cambios en el router (los parámetros de consulta sean compatibles con los signos de interrogación)
- Se ha mejorado el tiempo de depuración y pruebas

ANGULAR 14 .- Viene con una serie de capacidades integradas para facilitar la creación de aplicaciones de alta calidad.

- Una menor necesidad de código complejo
- Diagnóstico de desarrollador extendido
- **No hay necesidad de módulos ng**
- **Componentes independientes**
- Mejoras de CLI angular
- Captura de fusión nula
- Banana-in-a-box Error

Una mejora notable en Angular 14 es la introducción de **standalone components**.



→ Evolución y ventajas de Angular

Versiones de Angular

ANGULAR 15

Una adición notable en Angular 15 es la introducción de API independientes estables, que permiten a los desarrolladores crear aplicaciones sin depender de los módulos Ng.

- API independientes estables
- Introducción de un nuevo sistema API
- Depuración simplificada con seguimientos de pila mejorados
- Soporte ESBUILD experimental mejorado
- Single File Components
- Directivas de imagen más estables
- Componentes estabilizados basados en MDC

ANGULAR 16 .- El equipo de Angular de Google ha presentado recientemente el muy esperado Angular 16, marcando un hito importante desde su creación. Con su lanzamiento el 3 de mayo de 2023, Angular 16 es aclamado como la actualización más grande y notable hasta la fecha, superando al ya impresionante Angular 15.

La última versión eleva el listón aún más. El equipo de Angular ha estado a la altura del desafío y ha entregado un lanzamiento potente en 2023, repleto de características y actualizaciones impresionantes que satisfacen las necesidades de desarrolladores.

→ Evolución y ventajas de Angular

Versiones de Angular

v16 se ha unido a las filas del revolucionario marco Angular, trayendo consigo una serie de mejoras y actualizaciones que elevan sus capacidades y eficiencia para desarrolladores y entusiastas de la tecnología. Esta última versión de Angular es un testimonio de los comentarios y demandas de la comunidad, ya que aborda numerosas mejoras en la calidad de vida e **incorpora más de 2500 características altamente solicitadas en GitHub.**

- Angular Signals
- Vista previa mejorada de Hydration Developer
- Compilaciones más rápidas con esbuild Developer

- Documentación y esquemas mejorados para componentes independientes
- Opciones para mejorar los paquetes de JavaScript creados por Angular CLI
- Vincular la información del router a las entradas de los componentes
- Soporte para Tailwind CSS
- Compatibilidad con el aislamiento CSS
- Soporte de Native Trusted Types
- Función de importación dinámica de datos del router
- Nuevo componente del selector de intervalo de fechas



→ Evolución y ventajas de Angular

Versiones de Angular

ANGULAR 17

Angular 17, presentado el 8 de noviembre de 2023, presenta una serie de actualizaciones y funciones diseñadas para mejorar la experiencia del desarrollador. Entre estas mejoras, una adición destacada es la incorporación de **sintaxis incorporada para el flujo de control**. Esto permite a los desarrolladores una sintaxis de plantilla más expresiva y eficiente, enriqueciendo aún más las capacidades de Angular.

Las nuevas características incluyen:

- El nuevo e innovador sitio de Angular
- Standalone Components predeterminados

- Soporte SSR y SSG
- Flujo de control incorporado
 - **@if**
 - **@if, @else si, @else**
 - **@for**
- Migración automática al flujo de control built-in
- Deferrable Views
- Mejor rendimiento en la compilación con ES-build

→ Evolución y ventajas de Angular

Versiones de Angular

ANGULAR 18

Angular 18, presentado el 22 de mayo del 2024, introduce las siguientes características.

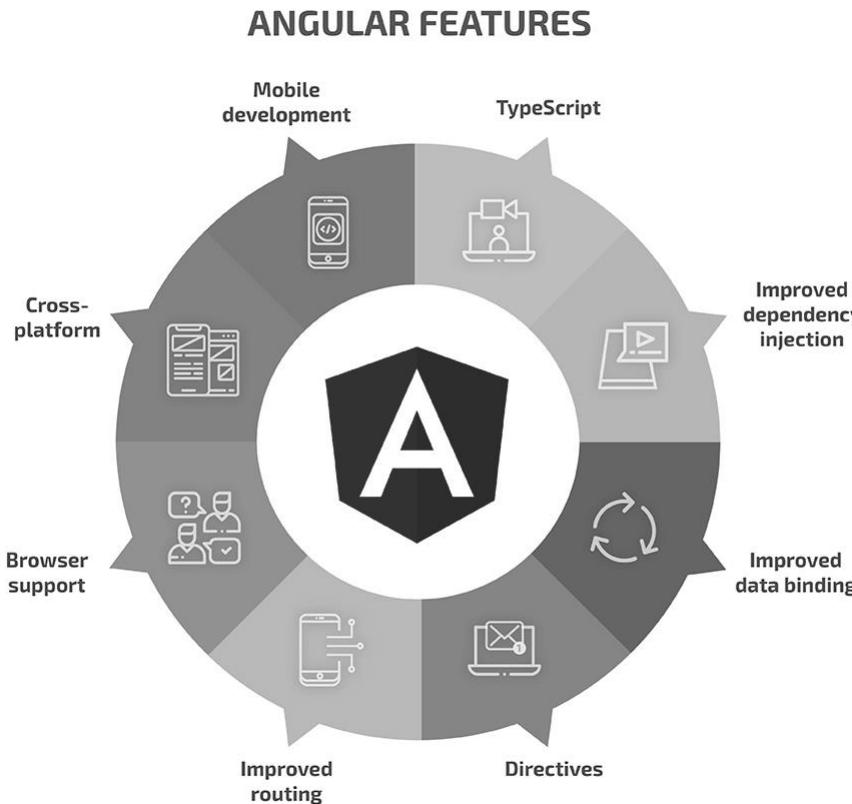
- **Sintaxis de control de flujo estable:**
 - @if y @for permiten un control de flujo más claro y limpio en las plantillas.
 - @defer introduce la capacidad de cargar contenido de forma diferida
- **Nuevo programador de detección de cambios:**
 - incorpora un nuevo programador de detección de cambios que reduce la dependencia de Zone.js.
- **Deprecación de HttpClientModule:**
 - Se reemplaza HttpClientModule por provideHttpClient, ofreciendo una configuración más moderna y flexible
- **Mejoras en ng-content y formularios:**
 - Se han realizado mejoras en el comportamiento de fallback de ng-content
 - La agrupación de eventos en formularios ha sido optimizada.

- **Renderización del lado del servidor:**
 - Las mejoras en la renderización del lado del servidor (SSR) incluyen la repetición de eventos y la internacionalización
- **Nuevo paquete de construcción:**
 - @angular/build es un nuevo paquete que facilita el proceso de construcción.
- **Eliminación del downleveling de Zone.js:**
 - La eliminación del downleveling de Zone.js para async/await
- **Alias de comando ng dev:**
 - Se introduce el alias ng dev como una forma más intuitiva y rápida de iniciar el desarrollo de aplicaciones Angular-
- **Route Redirects as Functions:**
 - Ahora se pueden definir direcciones de rutas como funciones, permitiendo mayor flexibilidad en la lógica de redirecccionamiento.
- **Introducción de @let:**
 - Permite crear variables directamente en HTML, facilitando operaciones simples dentro de las plantillas.



→ Evolución y ventajas de Angular

Ventajas



- **Creación de clientes enriquecidos de forma modular**
(Componentes web y reusabilidad)
- **Uso del patrón MVC pero del lado del cliente**
(Separación de lógica de componente y UI)
- **Data Binding en vistas**, en lugar de código
- **Angular-CLI** ayuda a automatizar el proceso de construcción de componentes.
- **Uso de TypeScript** (respaldado por Microsoft)
(Infierno js, variables tipadas y POO)

03



Arquitectura de Angular

→ Arquitectura de Angular

Consideraciones principales

Overview de la aplicación

Características de la aplicación

Seguridad

Reglas de Negocio

Log / Auditoria

Servicios / Comunicación

Modelo de datos

Componentes por característica

Funcionalidades compartidas

→ Arquitectura de Angular

Consideraciones adicionales

Accesibilidad

i18n

Entornos

CI/CD

CDN, Containers, Server

Pruebas unitarias

Pruebas End-To-End

APIs

Y demás...

Angular Style Guide



- Convenciones de codificación
- Reglas de nomenclatura
- Estructura de aplicación
- Módulo organizador
- Creando y usando componentes
- Crear y usar servicios
- Lifecycle hooks

<https://angular.dev/style-guide>

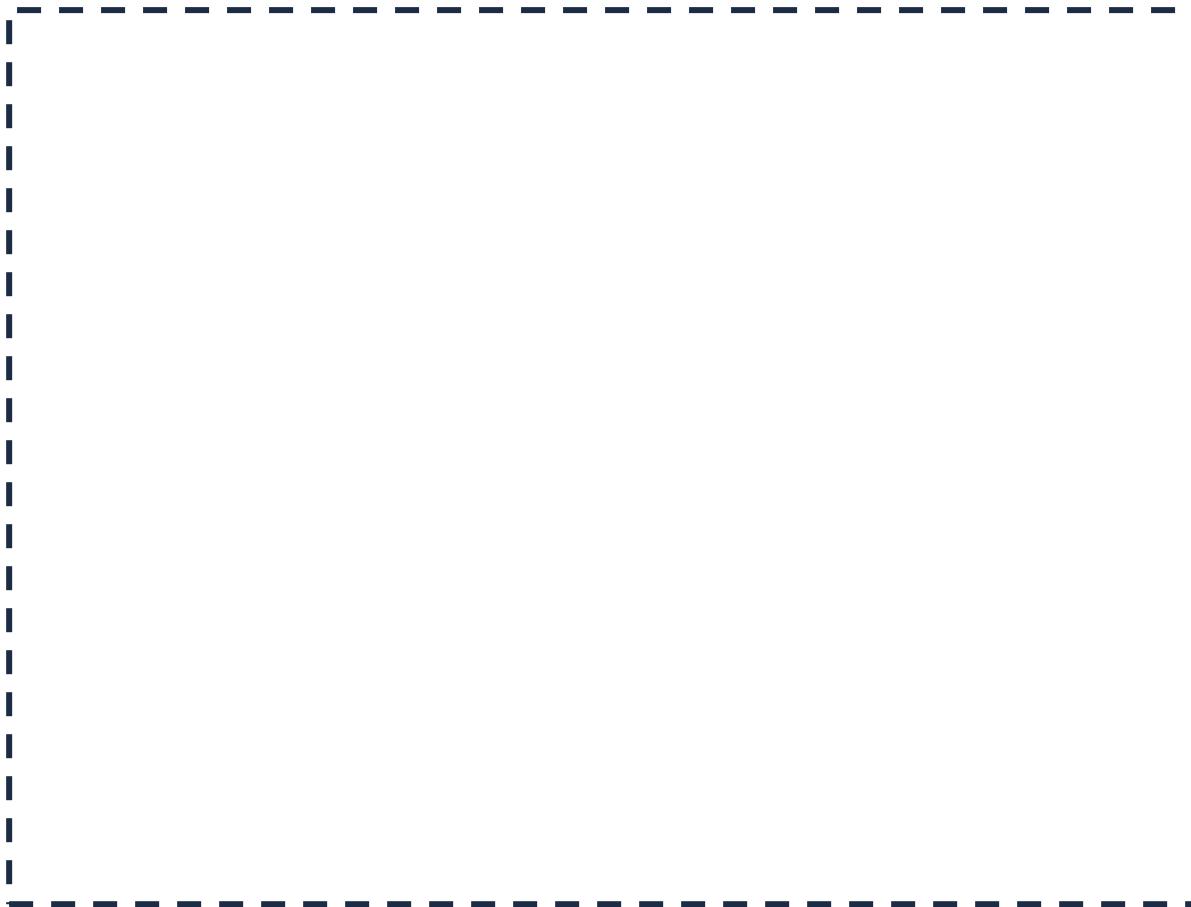
→ Arquitectura de Angular

Arquitectura de aplicación por módulos

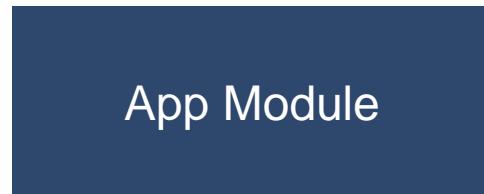


→ Arquitectura de Angular

Arquitectura de aplicación por módulos



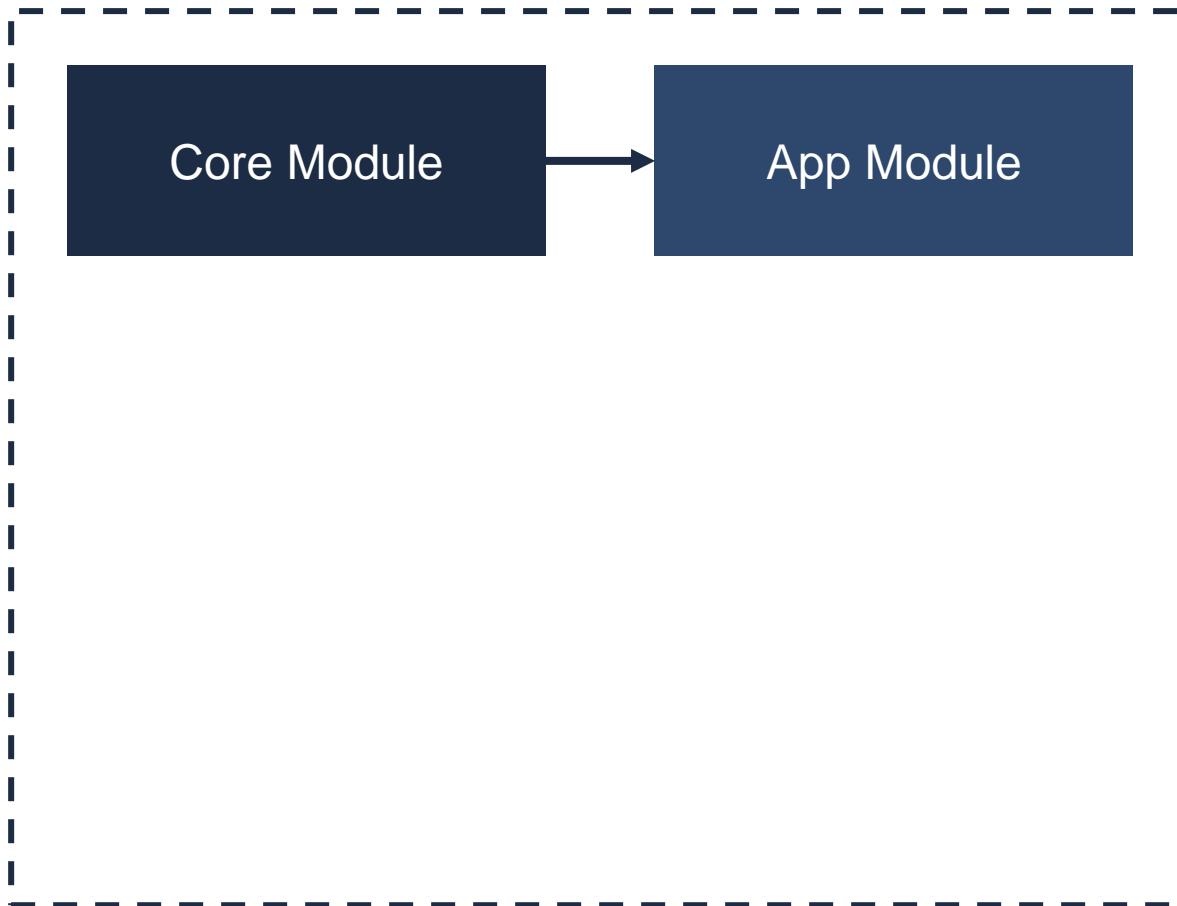
Arquitectura de aplicación por módulos



App Module

Es el **punto de inicio de la aplicación angular**. Este es el primer módulo que se llama cuando se carga la aplicación. Este módulo se crea automáticamente cuando la aplicación se crea con la ayuda de Angular-cli

Arquitectura de aplicación por módulos

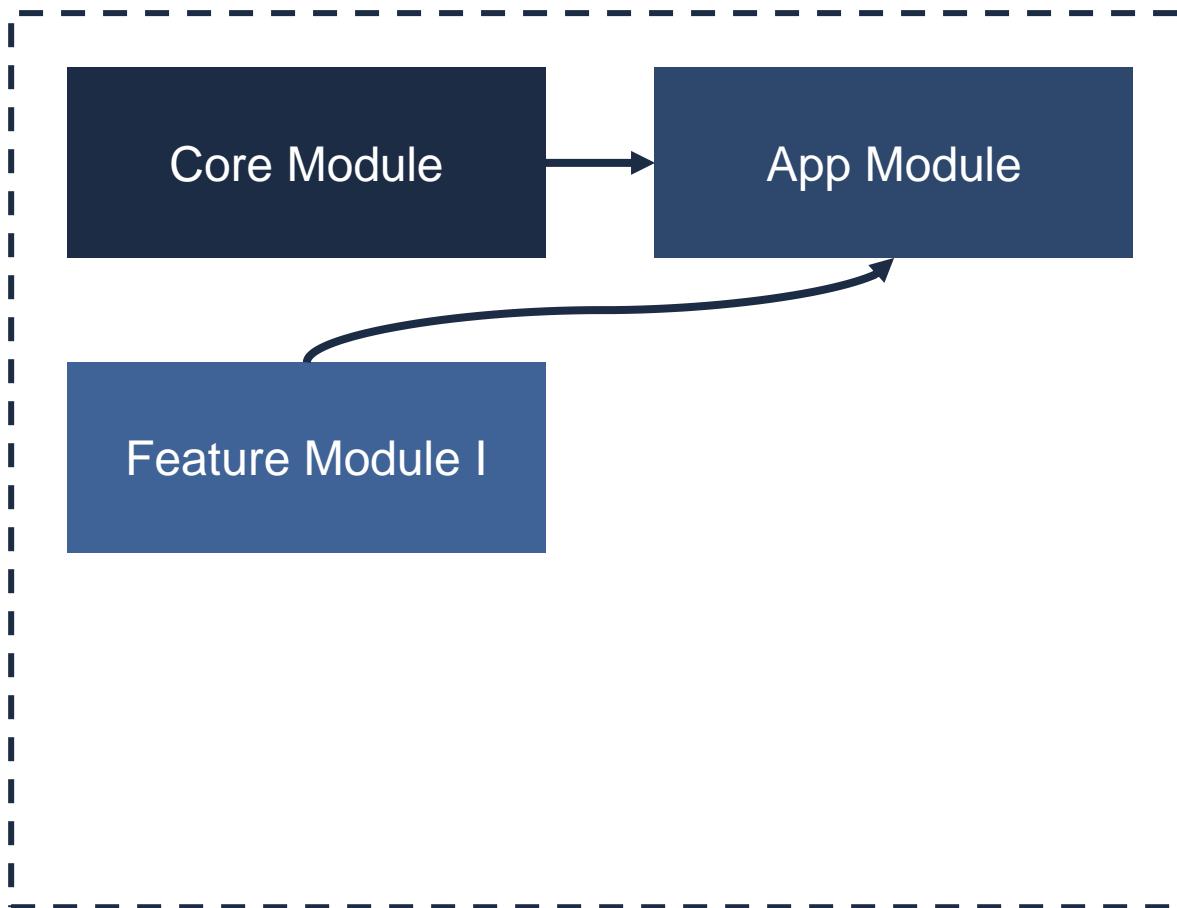


Core Module

En la aplicación angular, su módulo tendrá **servicios singleton a nivel de aplicación** (por ejemplo, HttpInterceptor utilizado para interceptar la solicitud http, LoggerService, LoggedInUserService) o el componente a nivel de aplicación (por ejemplo, Barra de navegación).

→ Arquitectura de Angular

Arquitectura de aplicación por módulos



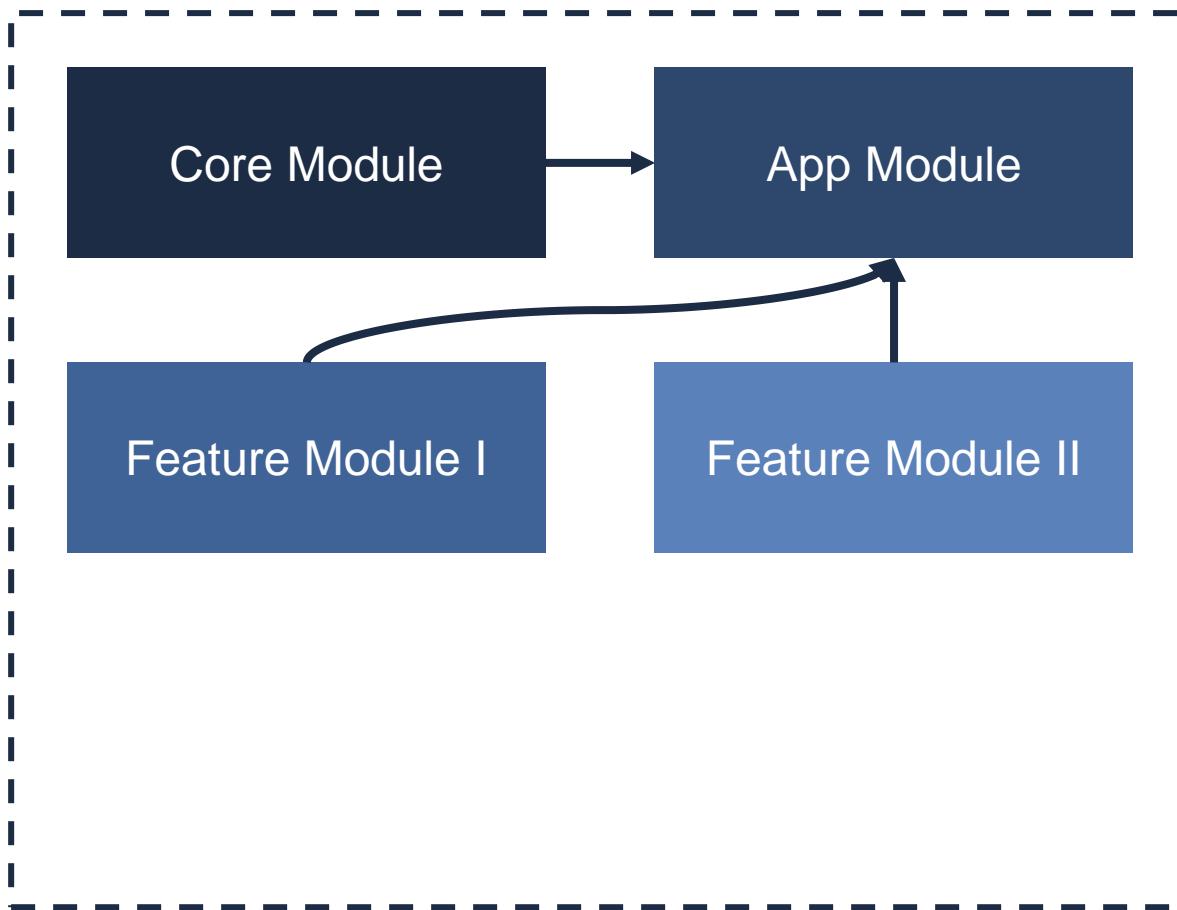
Feature Module

Tendrán características que la aplicación proporcionará.

(Ej. CustomerModule, OrdersModule, ProductsModule, ShippingModule).

→ Arquitectura de Angular

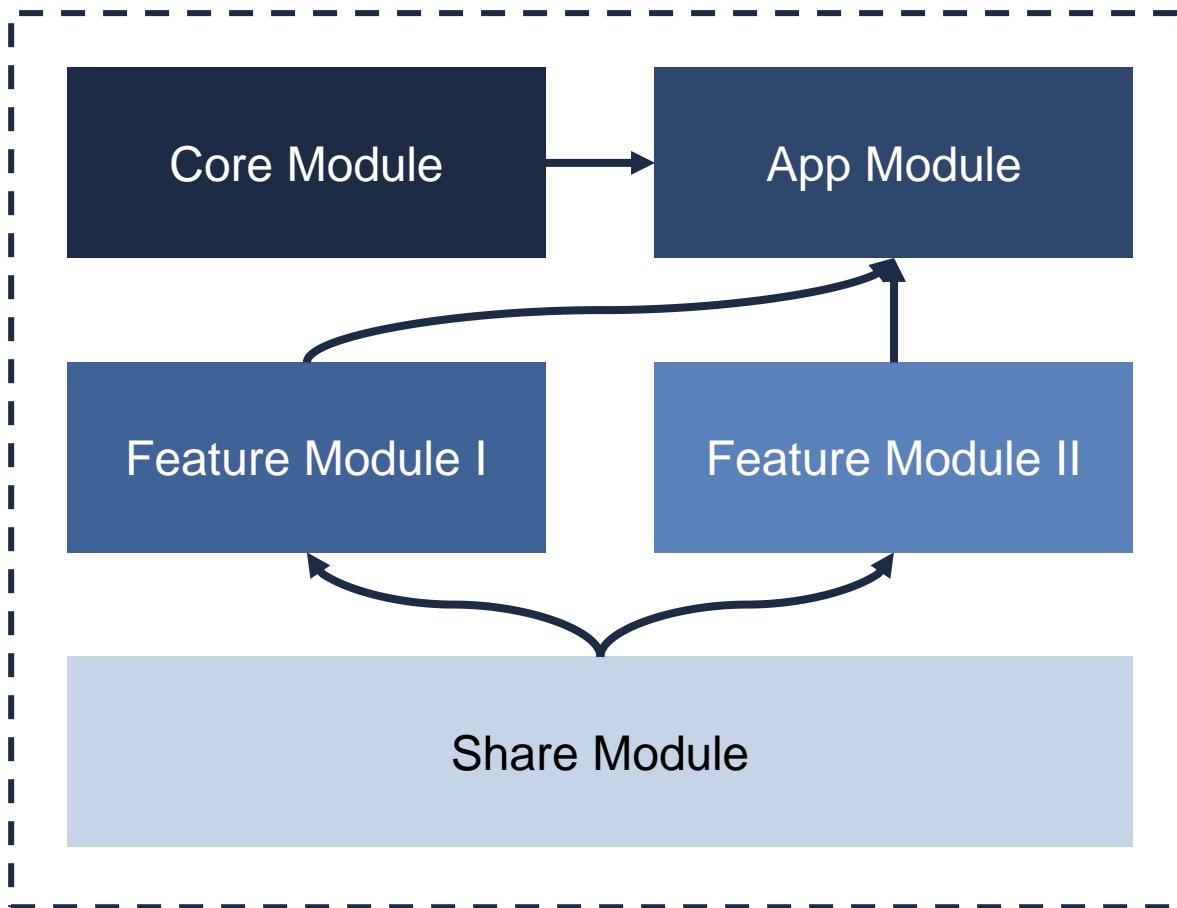
Arquitectura de aplicación por módulos



Feature Module

Tendrán características que la aplicación proporcionará.
(Ej. CustomerModule, OrdersModule, ProductsModule, ShippingModule).

Arquitectura de aplicación por módulos

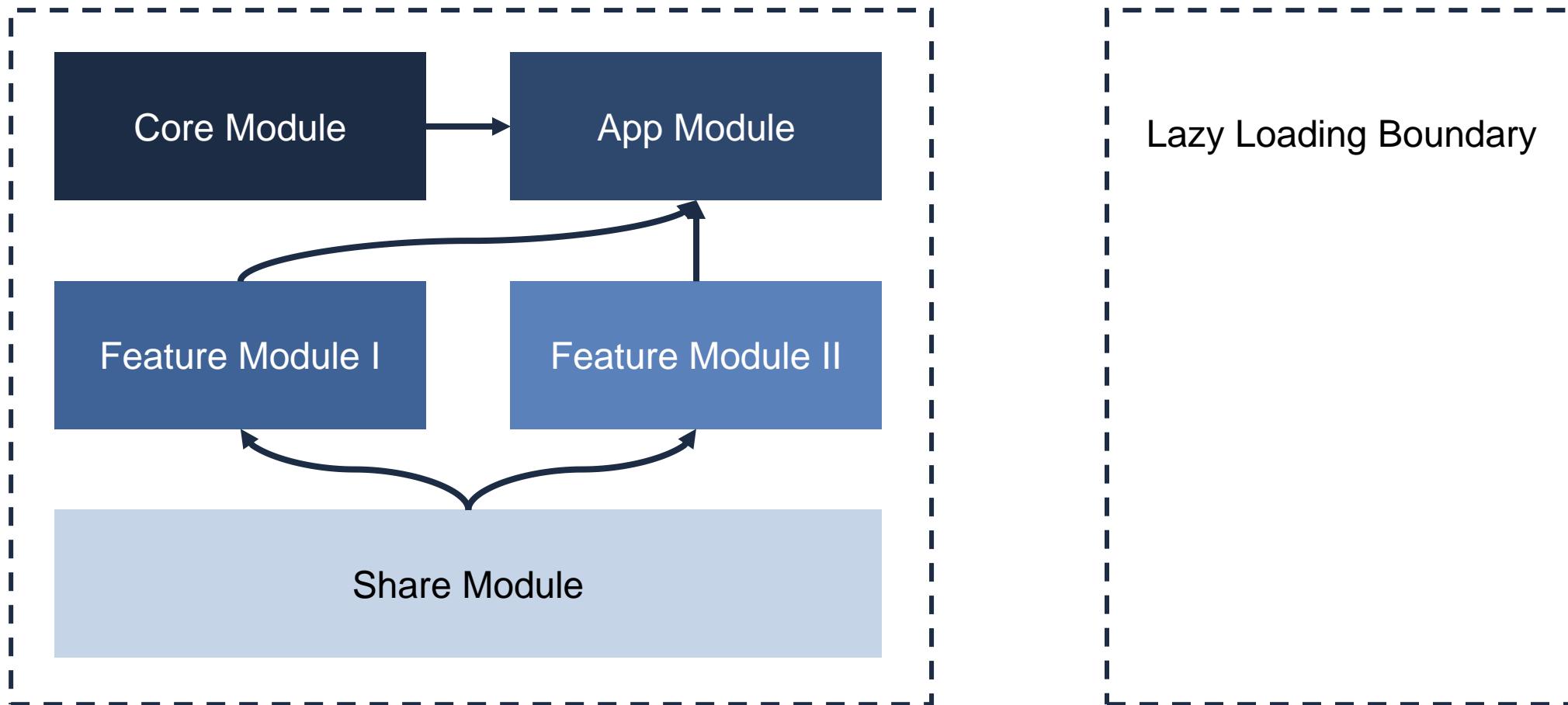


Share Module

Contiene **componentes altamente reusables**. Funciones comunes utilizadas en toda la aplicación, Directiva utilizada por controles, etc., utilidades similares utilizadas en la aplicación.

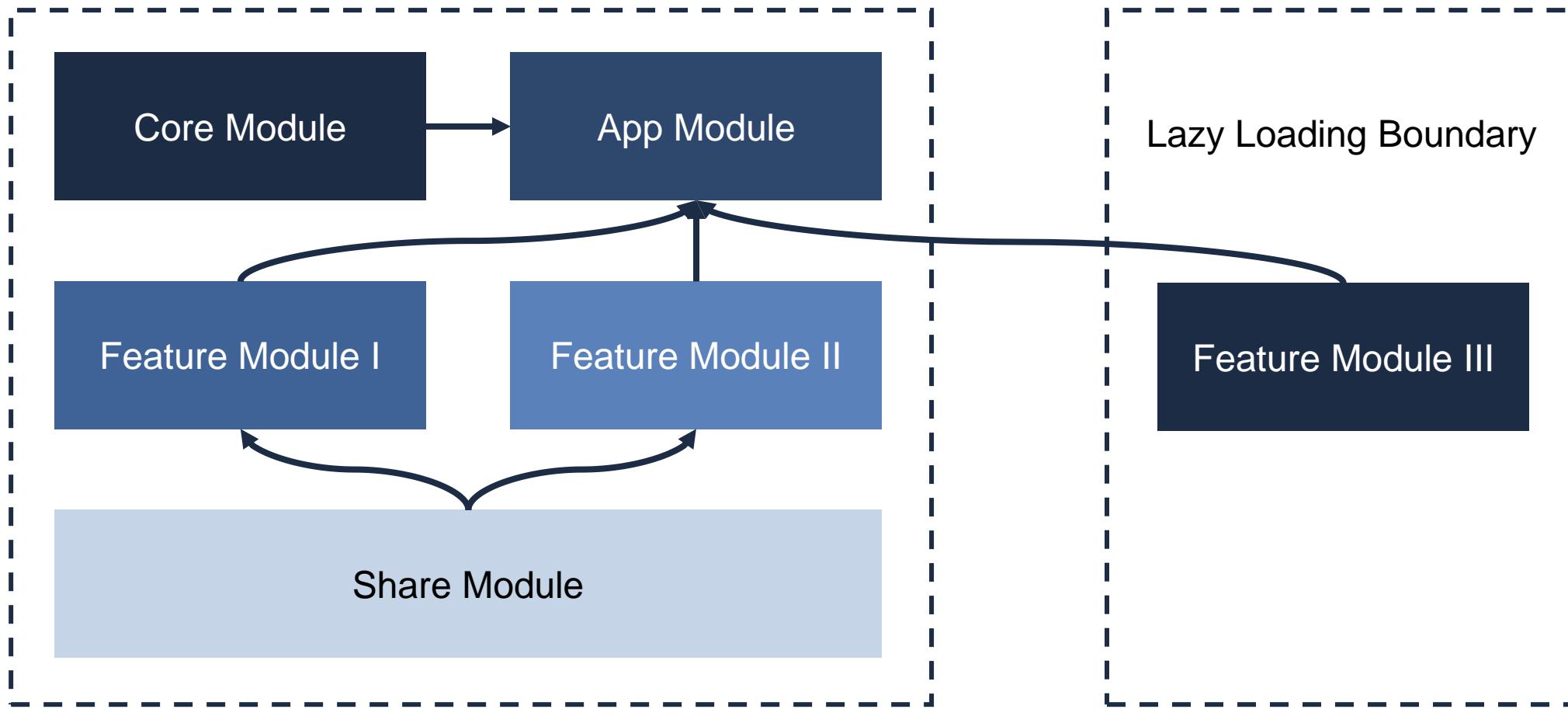
→ Arquitectura de Angular

Arquitectura de aplicación por módulos



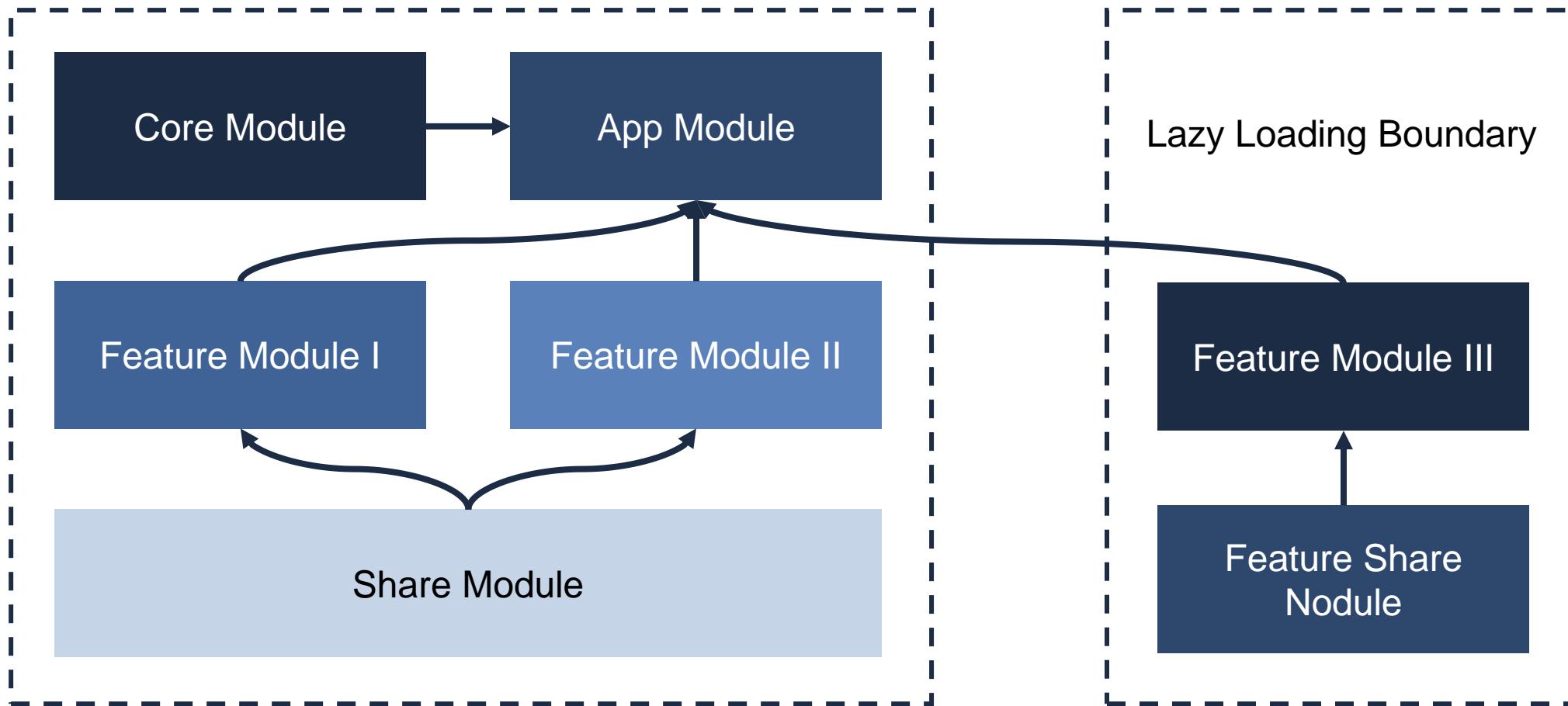
→ Arquitectura de Angular

Arquitectura de aplicación por módulos



→ Arquitectura de Angular

Arquitectura de aplicación por módulos



→ Arquitectura de Angular

Arquitectura de aplicación por módulos

```
project root
├── src
│   ├── app
│   │   ├── core
│   │   │   └── exception.service.ts|spec.ts
│   │   │   └── user-profile.service.ts|spec.ts
│   │   ├── heroes
│   │   │   ├── hero
│   │   │   │   └── hero.component.ts|html|css|spec.ts
│   │   │   └── hero-list
│   │   │       └── hero-list.component.ts|html|css|spec.ts
│   │   └── shared
│   │       └── hero-button.component.ts|html|css|spec.ts
│   │       └── hero.model.ts
│   │       └── hero.service.ts|spec.ts
│   │       └── heroes.component.ts|html|css|spec.ts
│   │       └── heroes.routes.ts
│   └── shared
│       └── init-caps.pipe.ts|spec.ts
│       └── filter-text.component.ts|spec.ts
│       └── filter-text.service.ts|spec.ts
└── villains
    ├── villain
    │   ...
    └── villain-list
        ...
    └── shared
        ...
    └── ...
    └── villains.component.ts|html|css|spec.ts
    └── villains.module.ts
    └── villains-routing.module.ts
    └── app.component.ts|html|css|spec.ts
    └── app.routes.ts
    └── main.ts
    └── index.html
    ...
    └── node_modules/...
```

<https://angular.dev/style-guide#application-structure-and-nmodules>

→ Arquitectura de Angular

Arquitectura de aplicación con “clean arquitecture”



→ Arquitectura de Angular

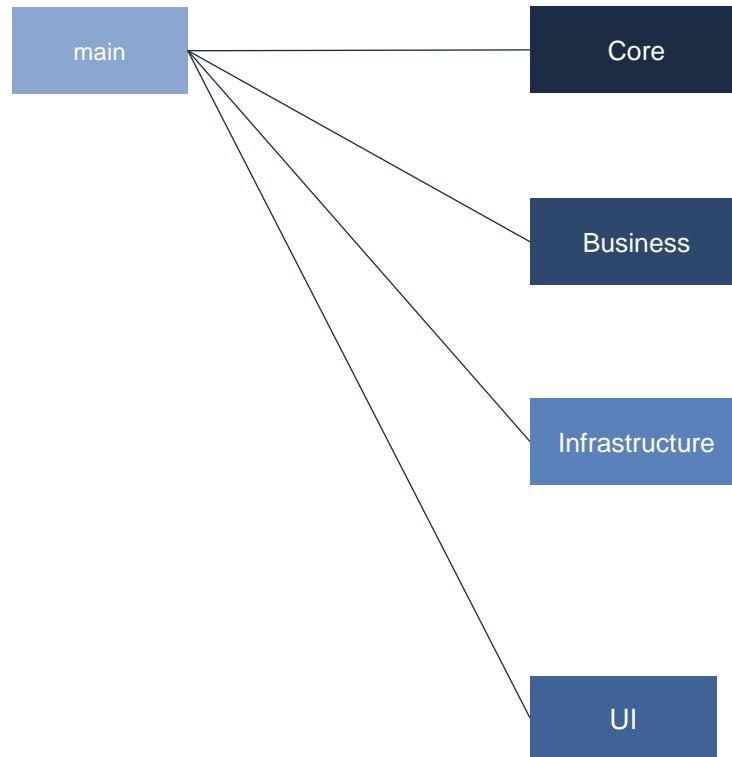
Arquitectura de aplicación con “clean arquitecture”

main



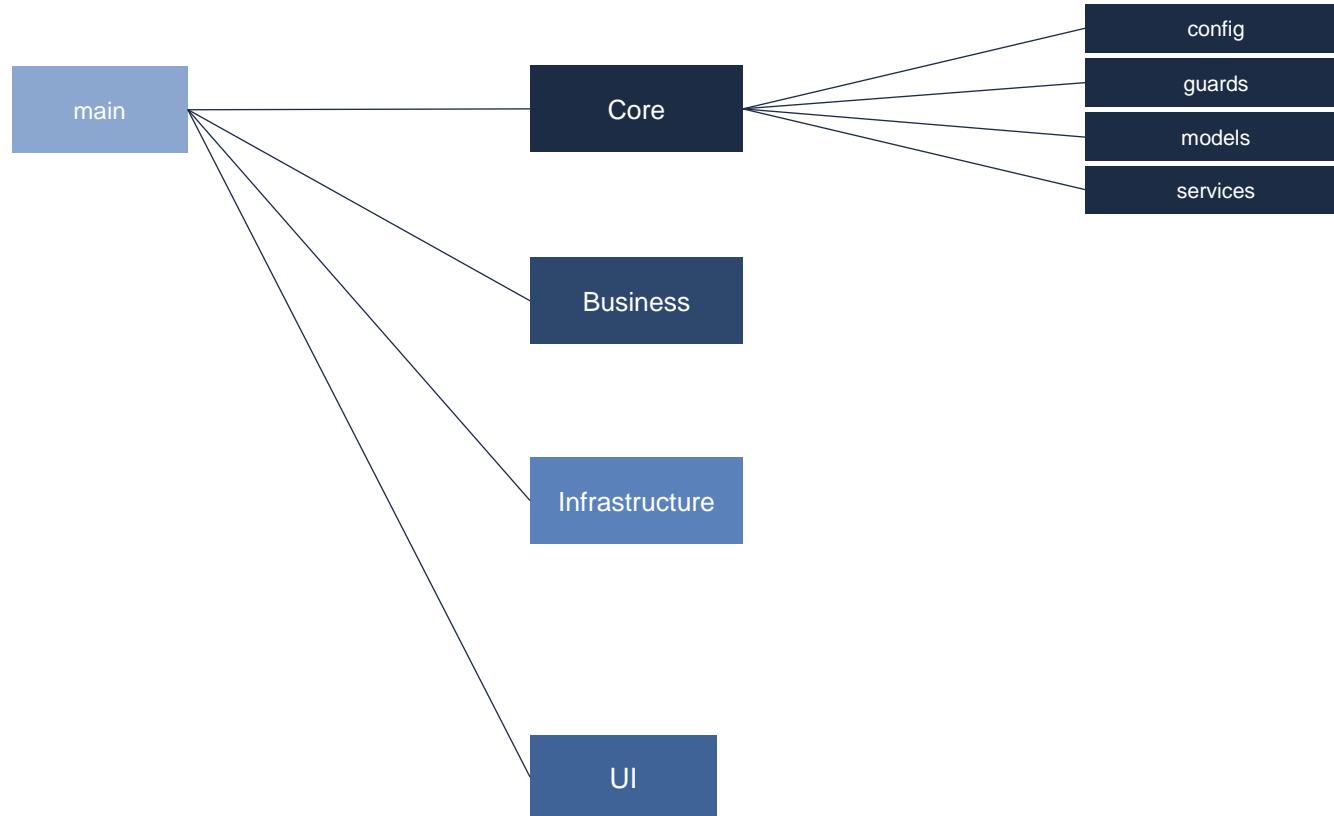
→ Arquitectura de Angular

Arquitectura de aplicación con “clean arquitecture”



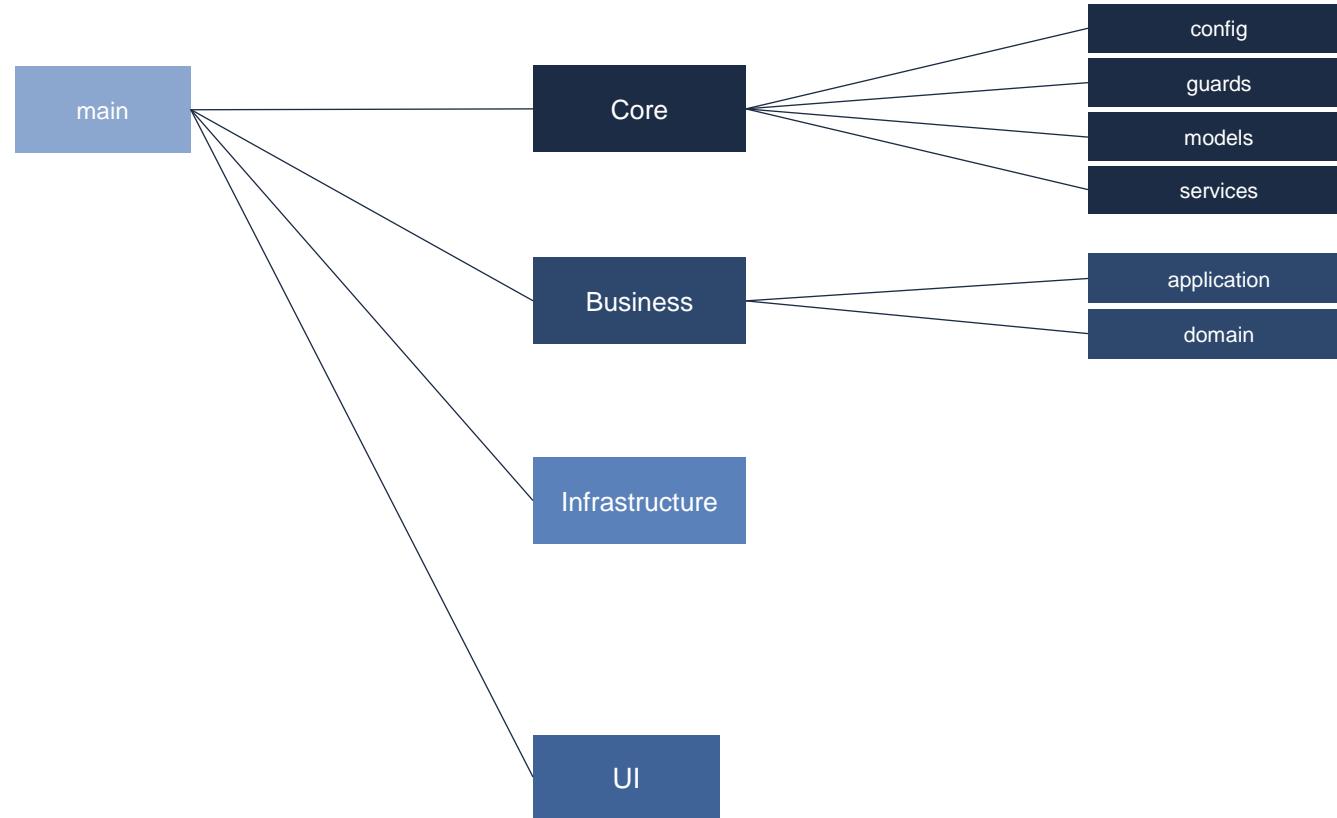
→ Arquitectura de Angular

Arquitectura de aplicación con “clean arquitecture”



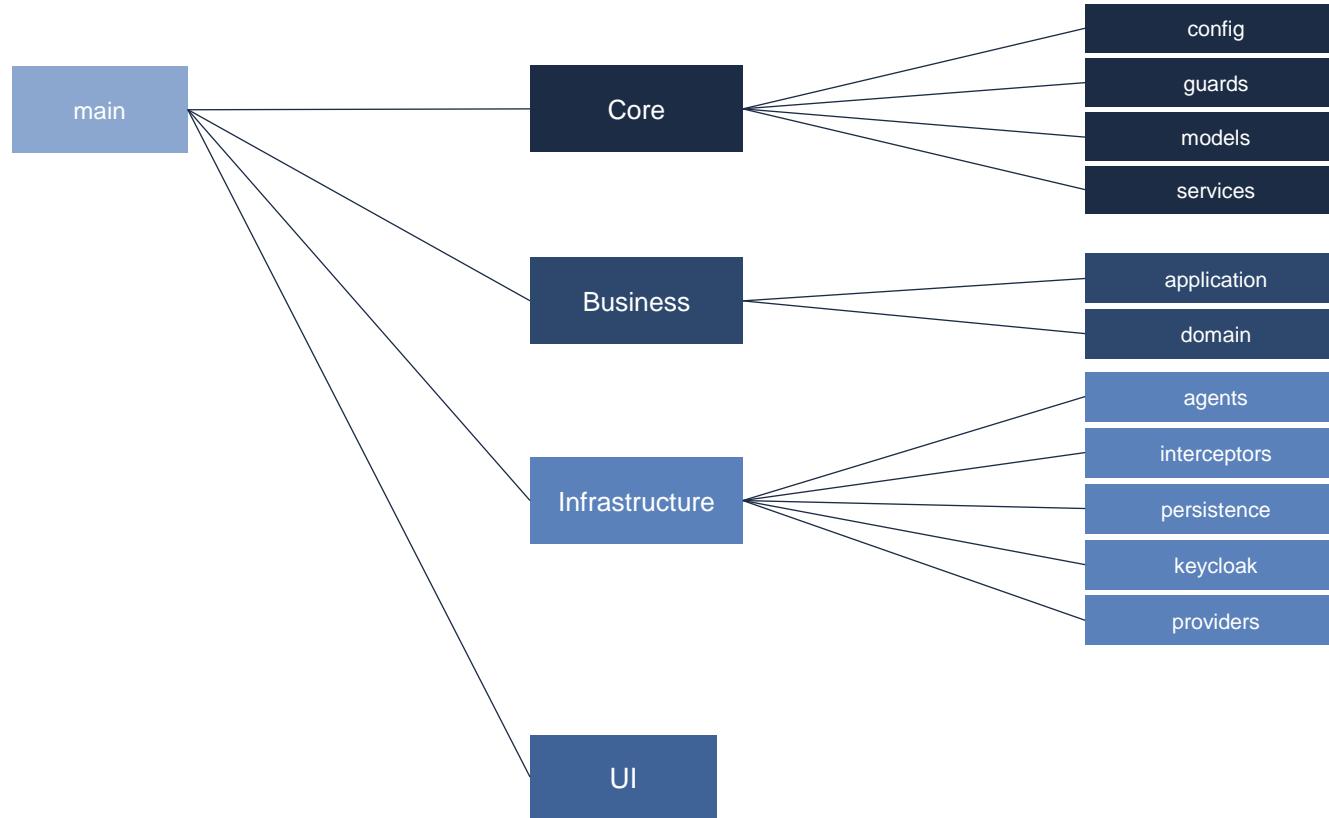
→ Arquitectura de Angular

Arquitectura de aplicación con “clean arquitecture”



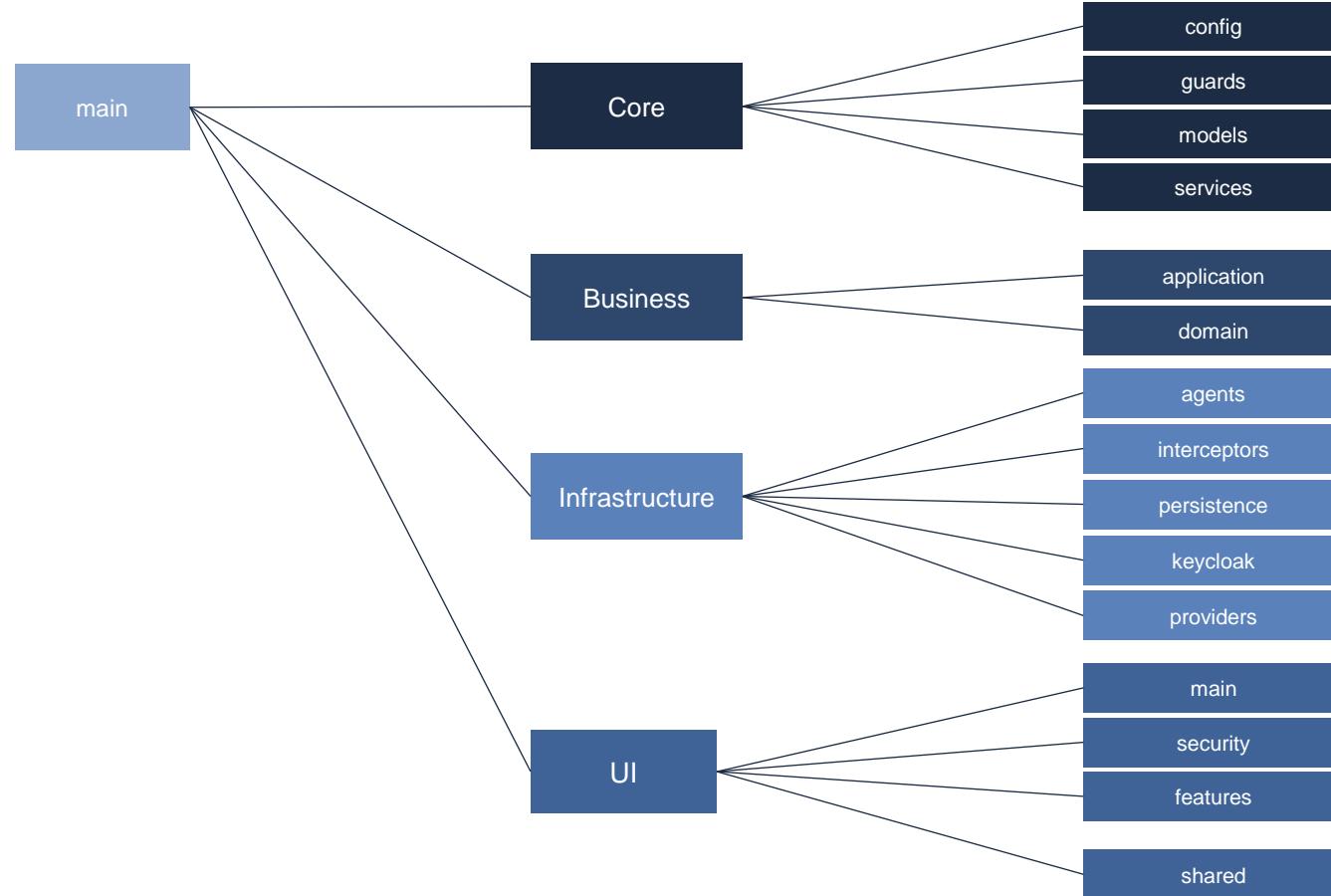
→ Arquitectura de Angular

Arquitectura de aplicación con “clean arquitecture”



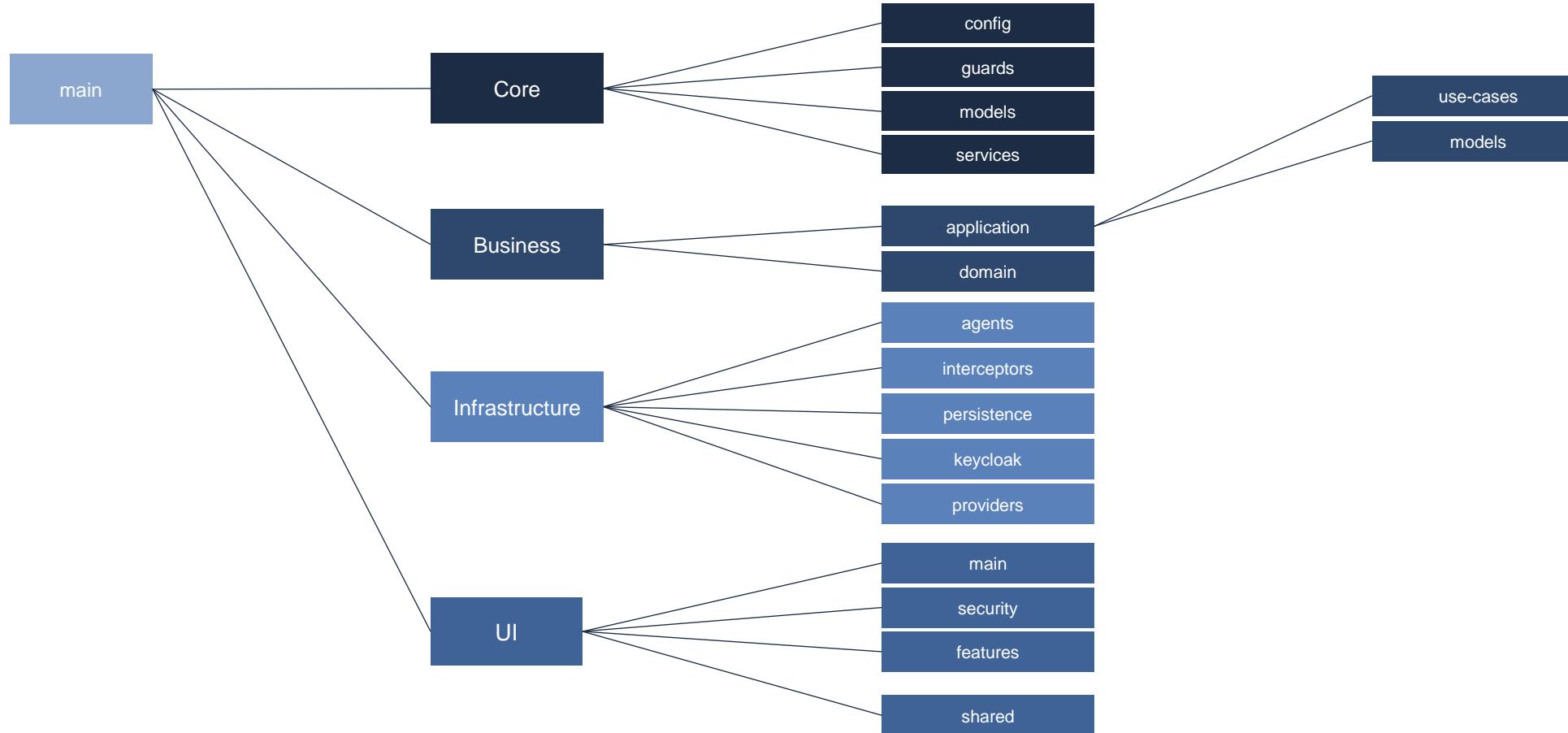
→ Arquitectura de Angular

Arquitectura de aplicación con “clean arquitecture”



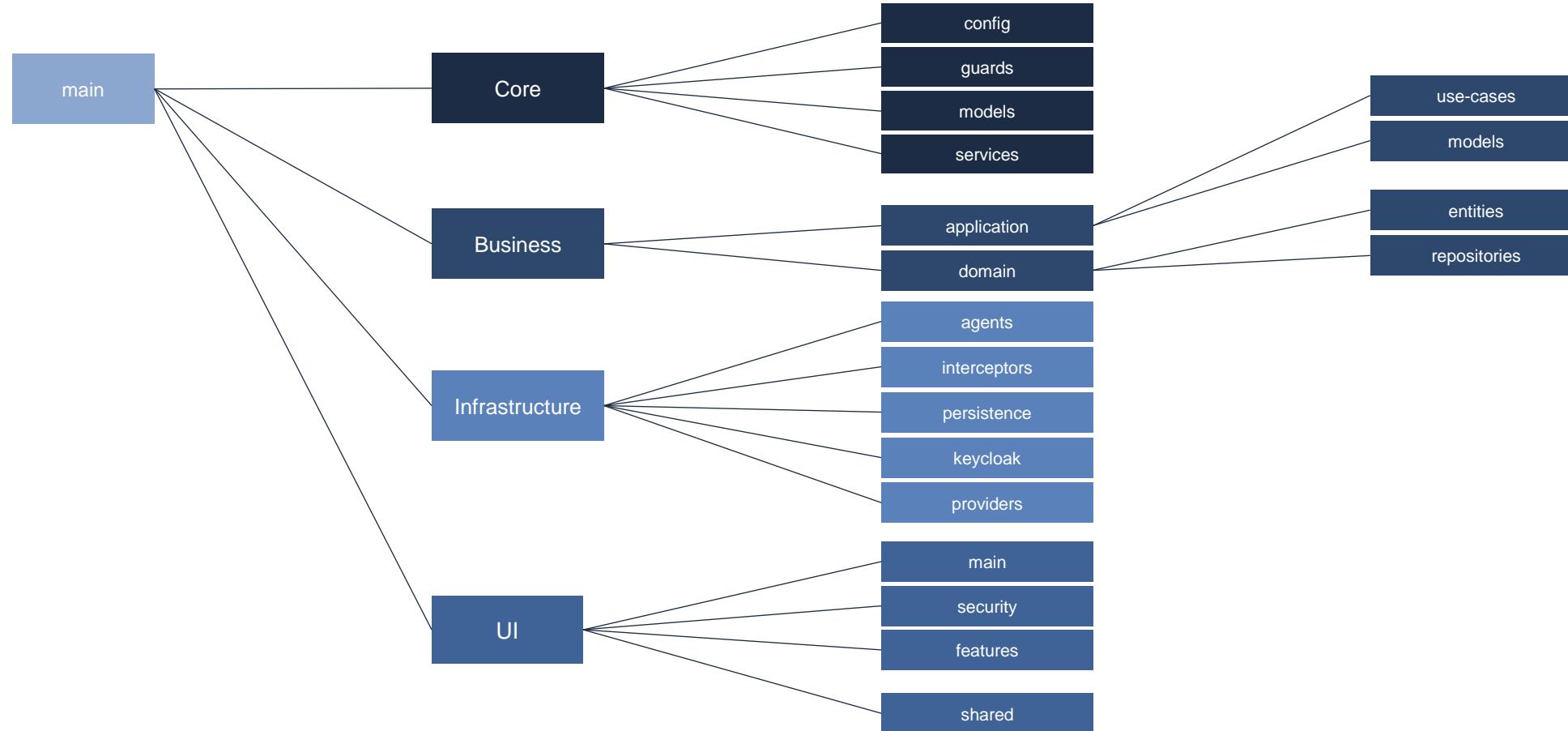
→ Arquitectura de Angular

Arquitectura de aplicación con “clean arquitecture”



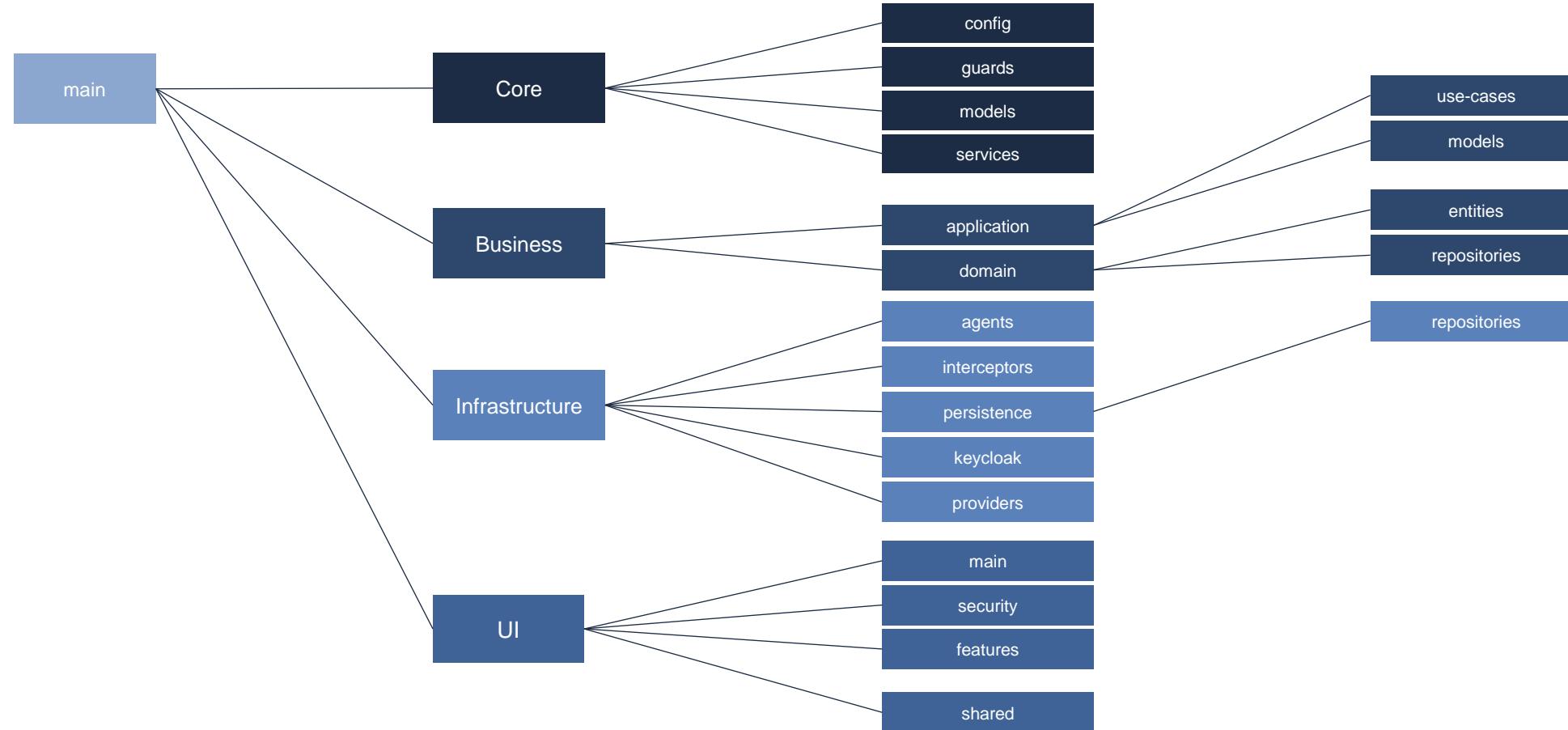
→ Arquitectura de Angular

Arquitectura de aplicación con “clean arquitecture”



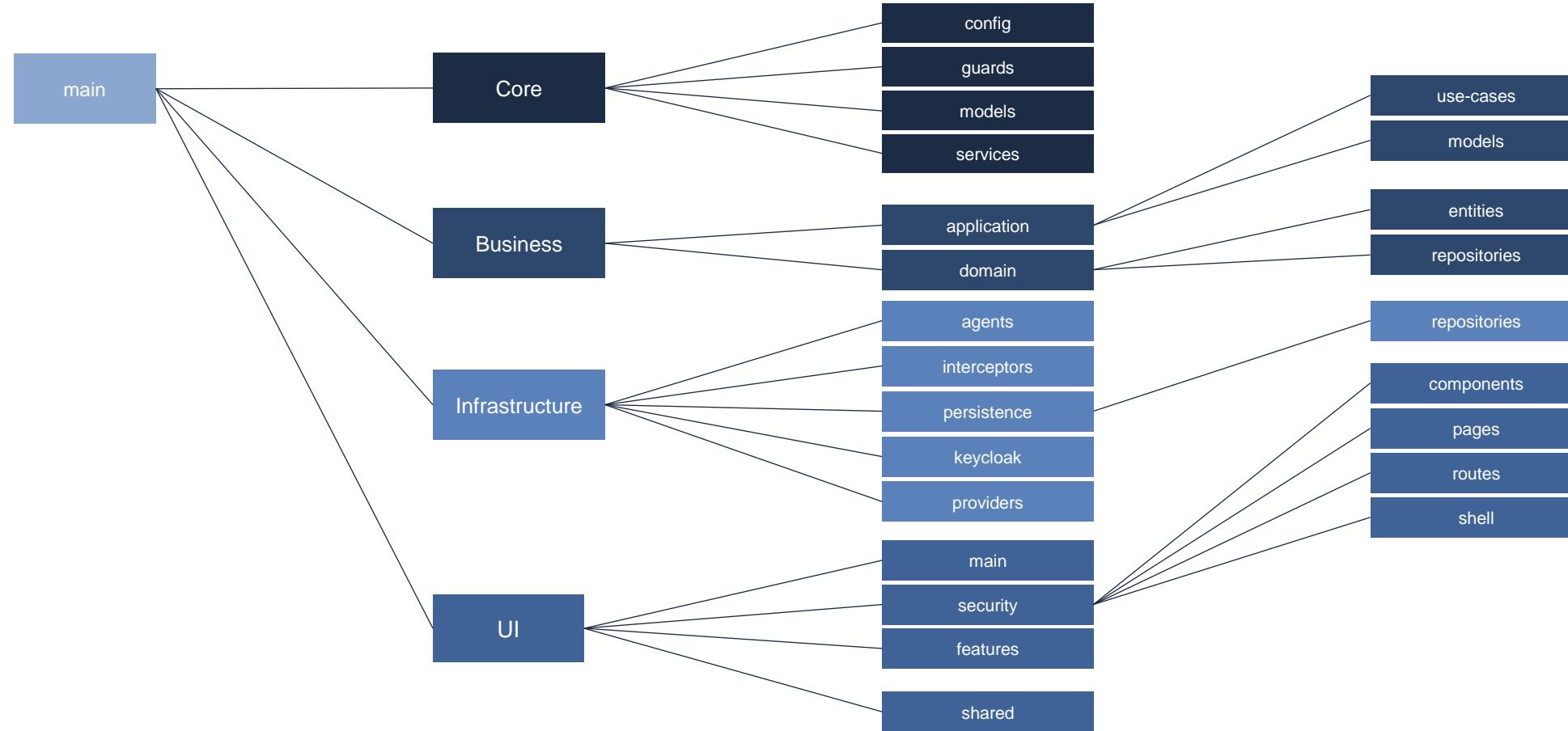
→ Arquitectura de Angular

Arquitectura de aplicación con “clean arquitecture”



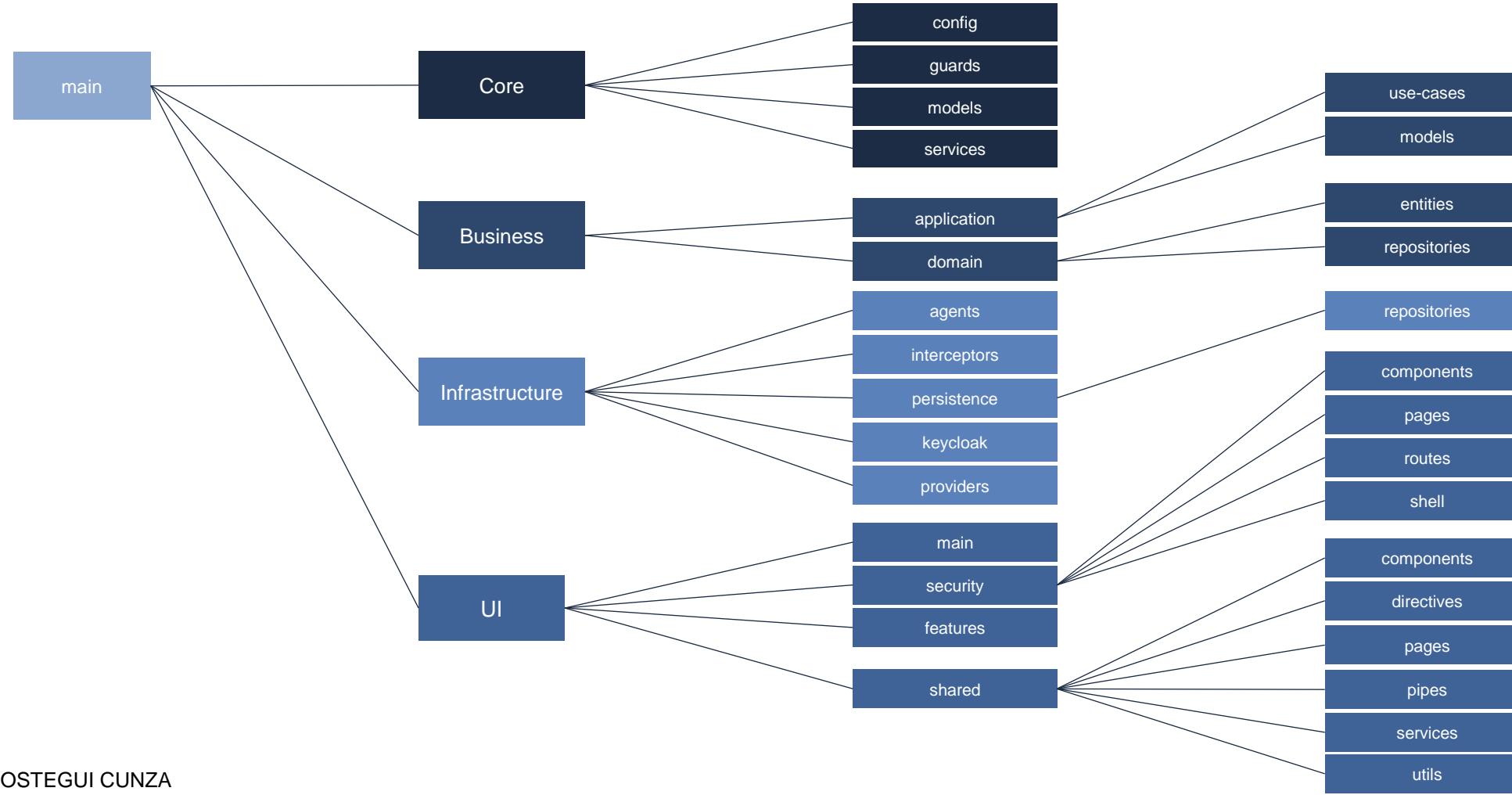
→ Arquitectura de Angular

Arquitectura de aplicación con “clean arquitecture”



→ Arquitectura de Angular

Arquitectura de aplicación con “clean arquitecture”



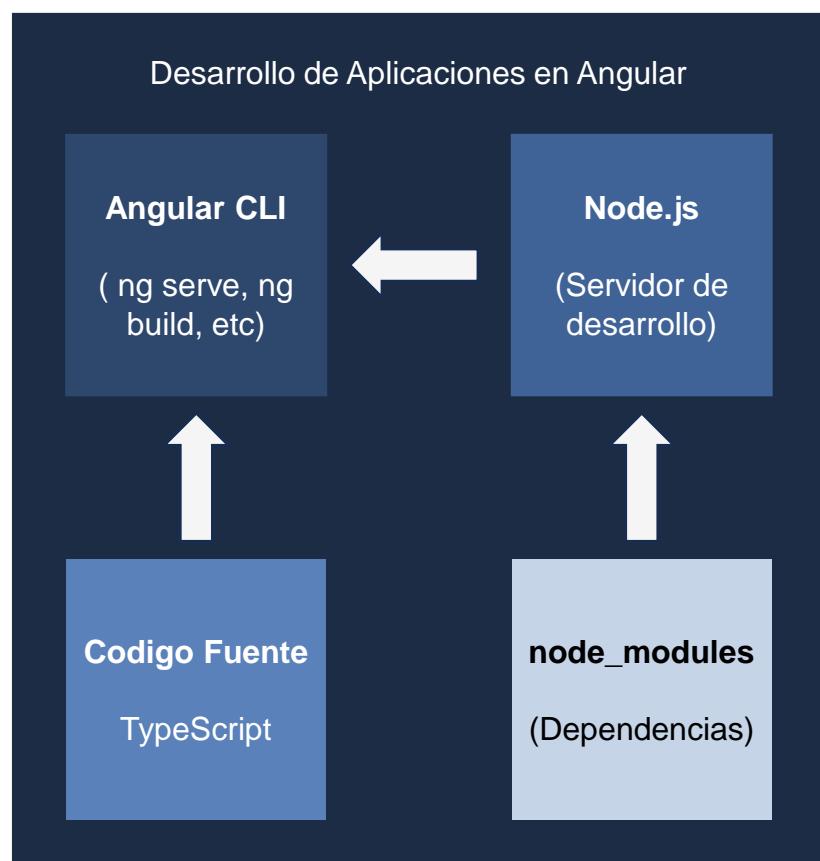
04



Herramientas de desarrollo (Node.js, npm y VSC).

→ Herramientas de desarrollo

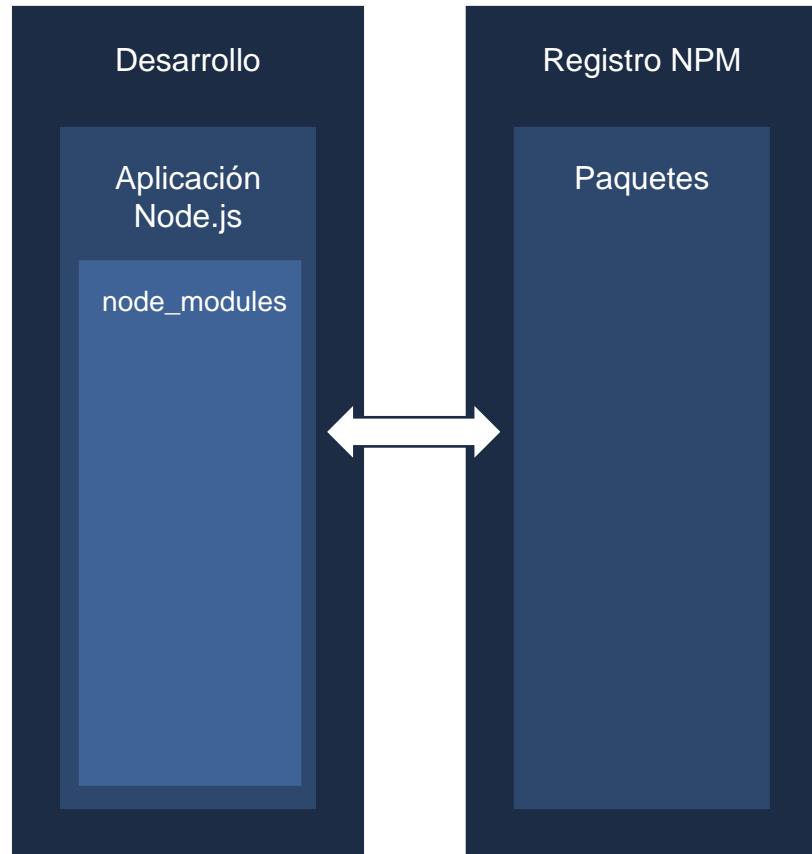
Node.js



- "**Angular CLI**" es la herramienta que utilizas para generar, desarrollar, probar y construir la aplicación. Está **construido con Node.js** y utiliza el código fuente TypeScript para generar la aplicación.
- "**Node.js**" se utiliza para **ejecutar el servidor de desarrollo** cuando usas el comando ng serve. También puede estar involucrado en otras tareas de desarrollo, como **ejecutar pruebas**.
- "**Código Fuente TypeScript**" código Angular. Escribe la aplicación en TypeScript, y luego con **Angular CLI lo compila a JavaScript** para que pueda ser ejecutado en el navegador.
- "**node_modules**" es donde se **almacenan todas las dependencias** del proyecto. Estas son las bibliotecas de terceros que la aplicación necesita para funcionar correctamente. **Se gestionan con npm**, que es el **gestor de paquetes de Node.js**.

→ Herramientas de desarrollo

NPM

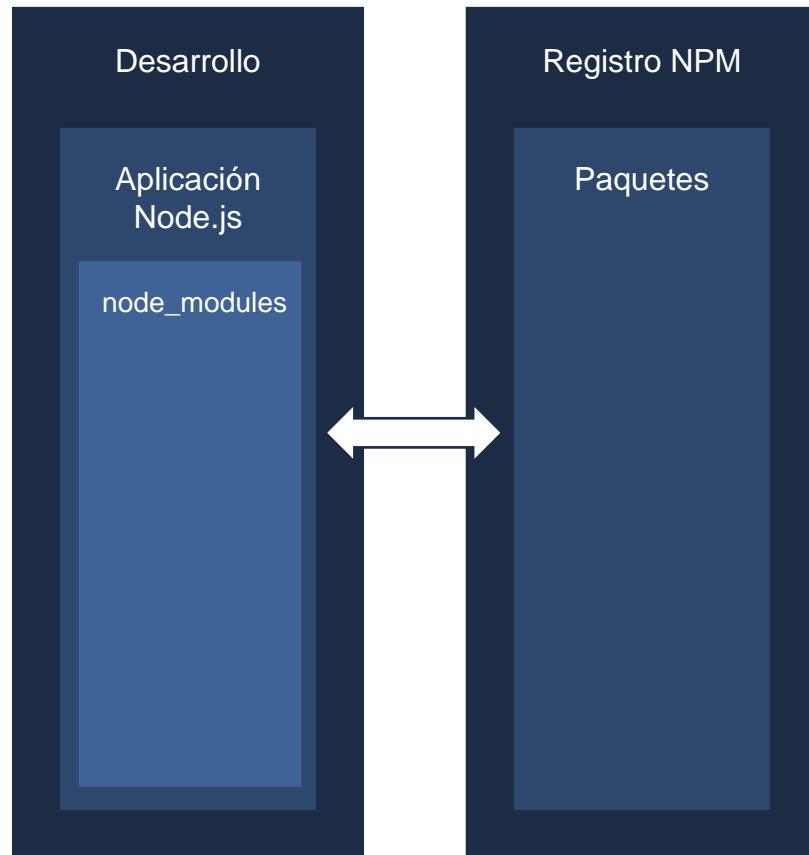


NPM, que significa **Node Package Manager**, es un **administrador de paquetes para JavaScript** y es el sistema de administración de paquetes predeterminado para Node.js, un entorno de ejecución para JavaScript.

NPM facilita a los desarrolladores **compartir y reutilizar código**. Proporciona un catálogo de más de 800,000 paquetes de código que se pueden **descargar e instalar en un proyecto de Node.js**. Estos paquetes de código, o módulos, pueden ser cualquier cosa, desde simples funciones de ayuda hasta complejas bibliotecas de marcos de trabajo.

→ Herramientas de desarrollo

NPM

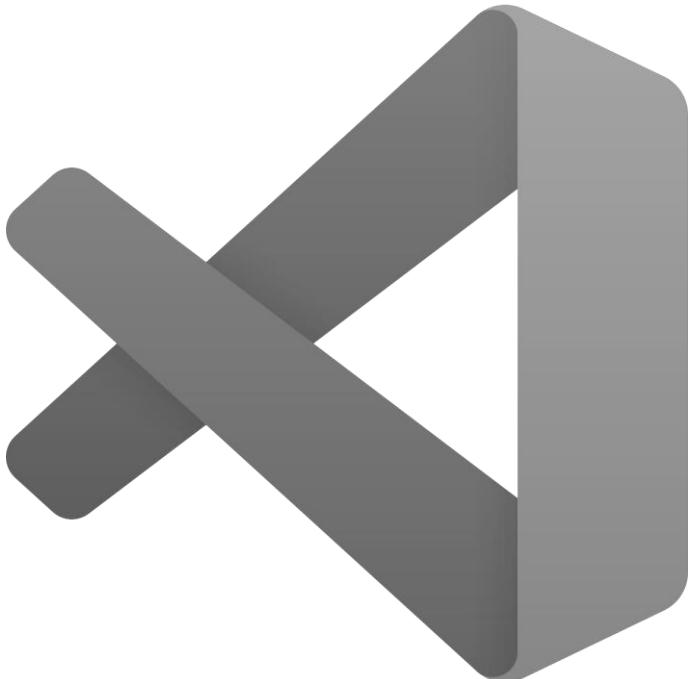


NPM, que significa **Node Package Manager**, es un **administrador de paquetes para JavaScript** y es el sistema de administración de paquetes predeterminado para Node.js, un entorno de ejecución para JavaScript.

NPM facilita a los desarrolladores **compartir y reutilizar código**. Proporciona un catálogo de más de 800,000 paquetes de código que se pueden **descargar e instalar en un proyecto de Node.js**. Estos paquetes de código, o módulos, pueden ser cualquier cosa, desde simples funciones de ayuda hasta complejas bibliotecas de marcos de trabajo.

→ Herramientas de desarrollo

VS Code



Visual Studio Code (VS Code) es un editor de código fuente desarrollado por Microsoft que se ejecuta en sistemas operativos como Windows, Linux y macOS. Gratuito y de código abierto.

Algunas de las características más destacadas de VS Code:

- **Edición de código sofisticada:** VS Code ofrece resaltado de sintaxis para más de 30 idiomas, emparejamiento de corchetes, autocompletado inteligente (IntelliSense), sangría y recorte automáticos, y más.
- **Control de versiones integrado:** VS Code se integra con Git de forma predeterminada, lo que permite a los desarrolladores realizar operaciones de Git desde la interfaz de usuario y ver cambios en el código en tiempo real.
- **Depuración:** Puede configurar puntos de interrupción, inspeccionar variables, ver la pila de llamadas y tener control interactivo sobre la ejecución del programa.
- **Extensiones:** Los usuarios pueden agregar nuevas funciones (como soporte para lenguajes adicionales, temas, formateadores de código, fragmentos de código, etc.) y personalizar su entorno de desarrollo a través de extensiones.
- **Terminal integrado:** VS Code tiene una terminal integrada que puede cambiar entre diferentes shells y también admite múltiples instancias.
- **Soporte de múltiples lenguajes de programación:** VS Code admite una amplia variedad de lenguajes de programación, desde JavaScript, **TypeScript**, Python, PHP, C++, HTML, CSS, Markdown, JSON, hasta Java, y muchos más.
- **Soporte para desarrollo en la nube y contenedores:** VS Code se puede utilizar con plataformas como Microsoft Azure para desarrollar y desplegar aplicaciones en la nube, y también tiene soporte para trabajar con contenedores Docker.

→ Herramientas de desarrollo

Visual Studio

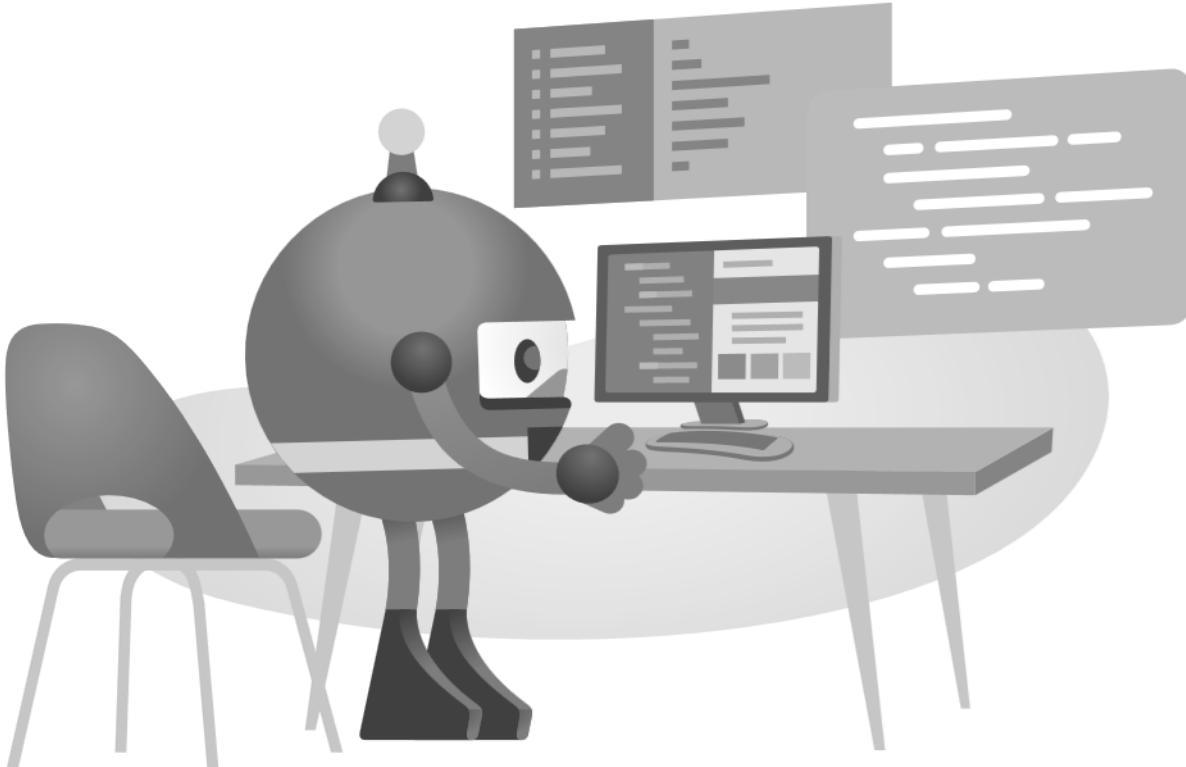


Visual Studio es un entorno de desarrollo integrado (IDE) completo desarrollado por Microsoft. Se utiliza para desarrollar aplicaciones de computadoras, aplicaciones web, aplicaciones móviles, servicios en la nube y sitios web. Visual Studio admite diferentes lenguajes de programación, como C#, C++, Visual Basic, JavaScript, Python y muchos otros. También admite una variedad de frameworks como .NET, ASP.NET, Xamarin, Unity, y más.

Algunas características clave de Visual Studio:

- **Edición avanzada de código:** Visual Studio proporciona resaltado de sintaxis, autocompletado de código (IntelliSense), refactorización de código, y funcionalidades de navegación en el código.
- **Depuración:** Visual Studio tiene un potente depurador que permite a los desarrolladores diagnosticar y probar su código. Los desarrolladores pueden establecer puntos de interrupción, paso a paso a través del código, y examinar variables y memoria.
- **Control de versiones:** Visual Studio se integra con varias herramientas de control de versiones como Git y Team Foundation Version Control.
- **Desarrollo de la interfaz de usuario:** Visual Studio tiene herramientas para diseñar y desarrollar interfaces de usuario. Por ejemplo, los desarrolladores pueden diseñar la interfaz de usuario de su aplicación arrastrando y soltando elementos en un diseñador visual.
- **Pruebas unitarias:** Visual Studio proporciona un marco para escribir pruebas unitarias y realizar pruebas automatizadas en su código.
- **Integración de bases de datos:** Visual Studio puede conectarse a varias bases de datos, lo que permite a los desarrolladores gestionar y manipular datos directamente desde el IDE.
- **Personalización y extensiones:** Los desarrolladores pueden personalizar Visual Studio y agregar nuevas funcionalidades mediante extensiones.

Mi primera aplicación



A black and white photograph of a man in a suit and tie, sitting at a desk and working on a laptop. He is looking down at the screen with a thoughtful expression, his hand resting near his chin. A smartphone lies next to the laptop.

**GRACIAS
POR SU PREFERENCIA**

