PREPÁRATE
PARA SER EL
MEJOR

+ ENTREMIENTO
EXPERIENCIA

BIENVENIDOS.

ENTRENAMIENTO
EXPERIENCIA

GALAXY
TRAINING

APLICACIONES WEB ANGULAR 8 (FRONT-END)

Sesión 02

Ing. Erick Arostegui Cunza
Instructor
earostegui@galaxy.edu.pe

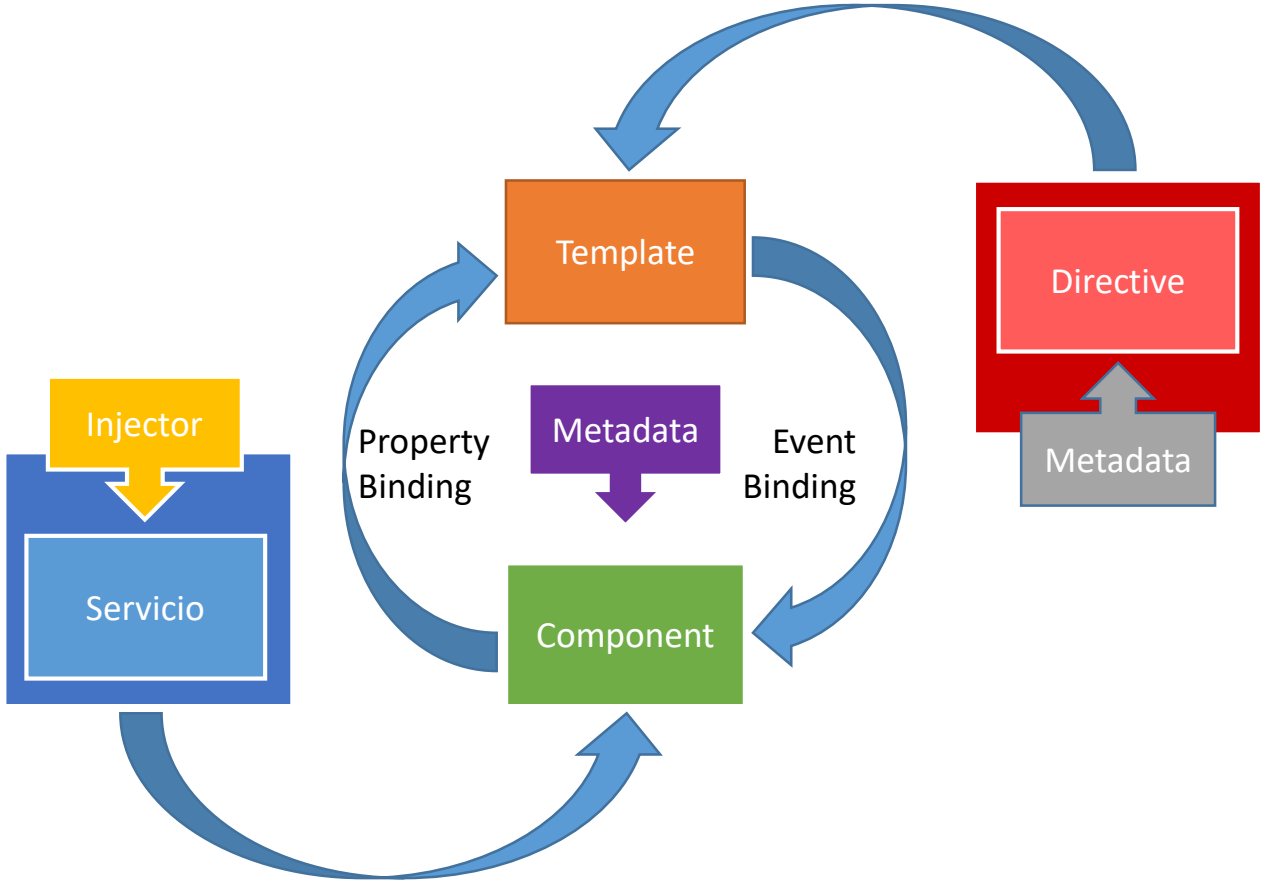
AGENDA

ARQUITECTURA Y MODELOS

- Diseño de la estructura del proyecto.
- Creación del Proyecto.
- Configuración del proyecto.
- Componentes y servicios core.
- Modelos usando TypeScript (clases, interfaces y herencia).

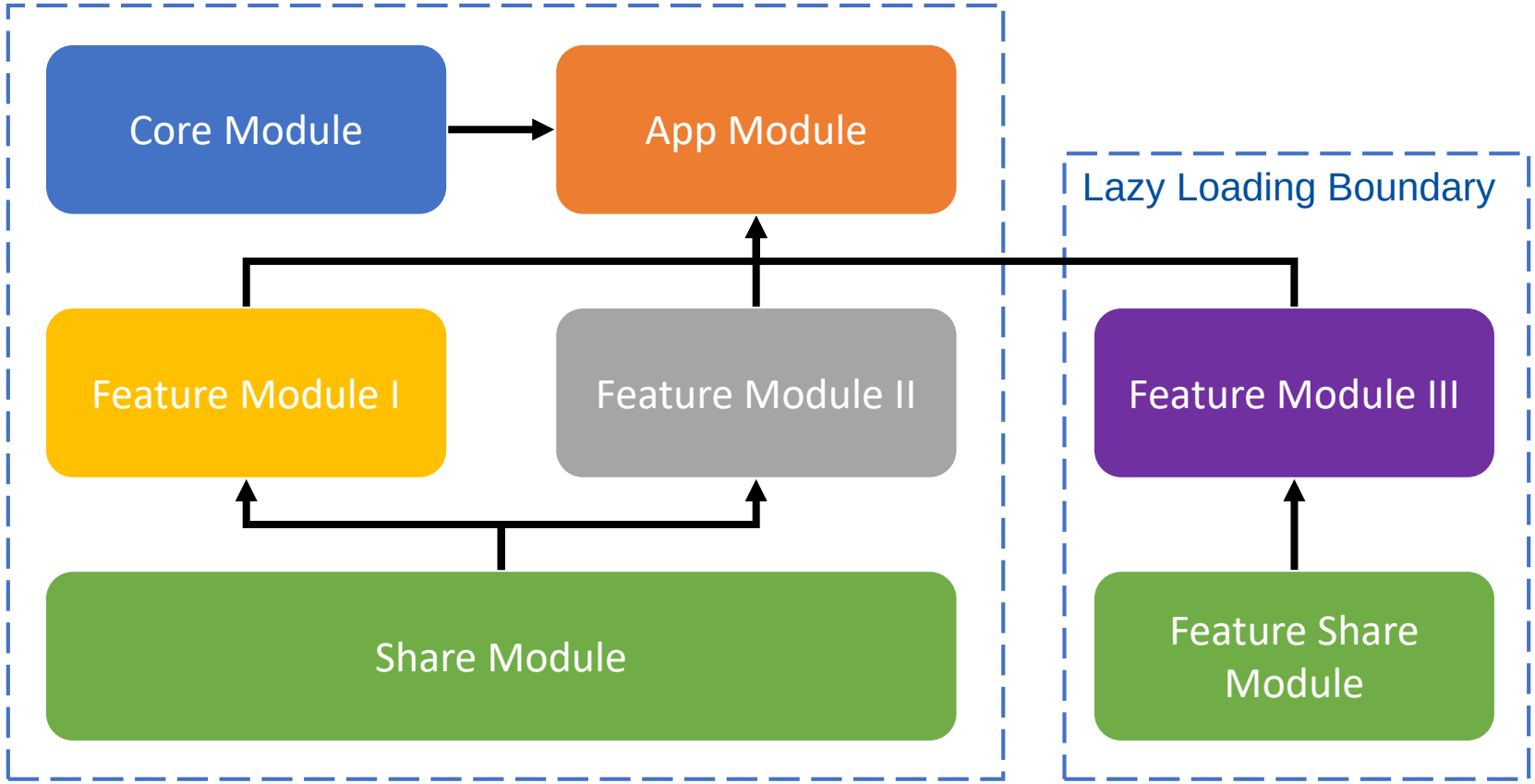
Diseño de la estructura del proyecto.

Esquema de ejecución de componentes



Diseño de la estructura del proyecto.

Arquitectura de aplicación



Diseño de la estructura del proyecto.

Basico

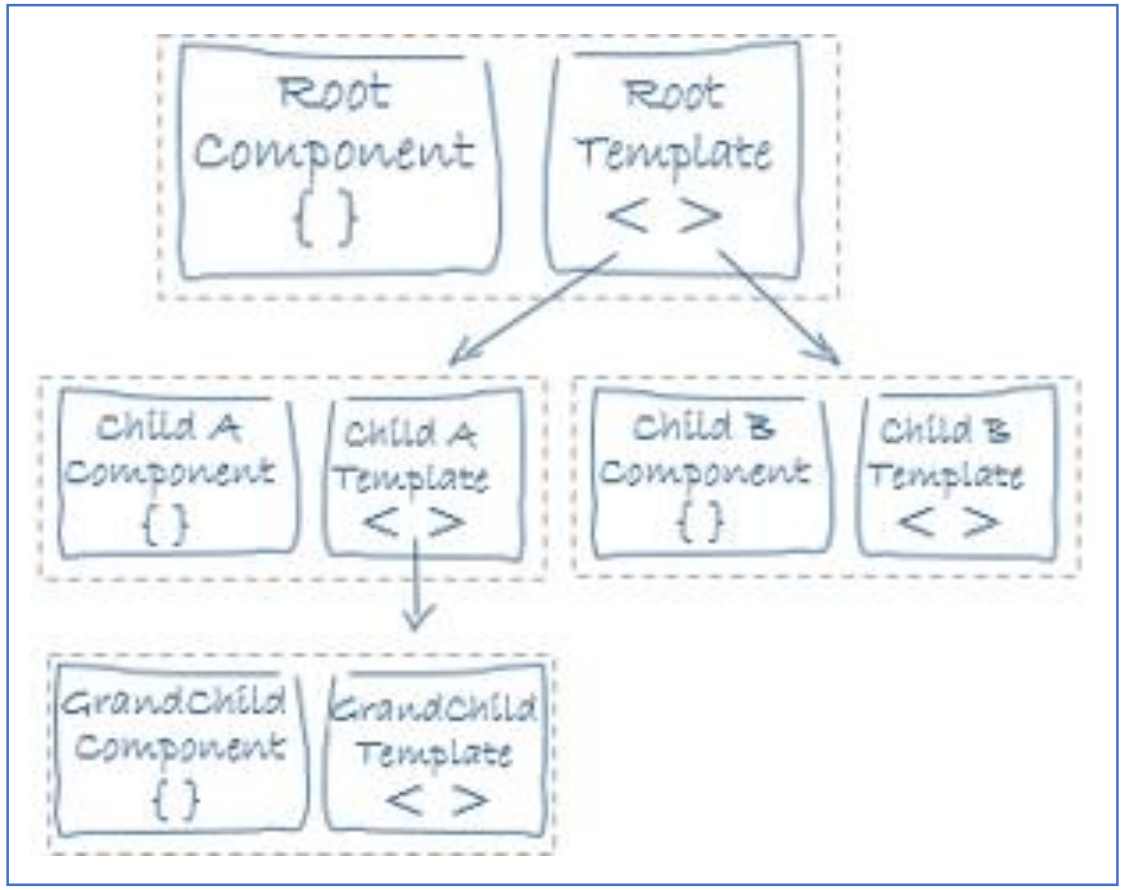
App Module : Es el punto de inicio de la aplicación angular. Este es el primer módulo que se llama cuando se carga la aplicación. Este módulo se crea automáticamente cuando la aplicación se crea con la ayuda de Angular-cli.

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).

Share Module : Contiene componente altamente reusables. Funciones comunes utilizadas en toda la aplicación, Directiva utilizada por controles, etc., utilidades similares utilizadas en la aplicación.

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

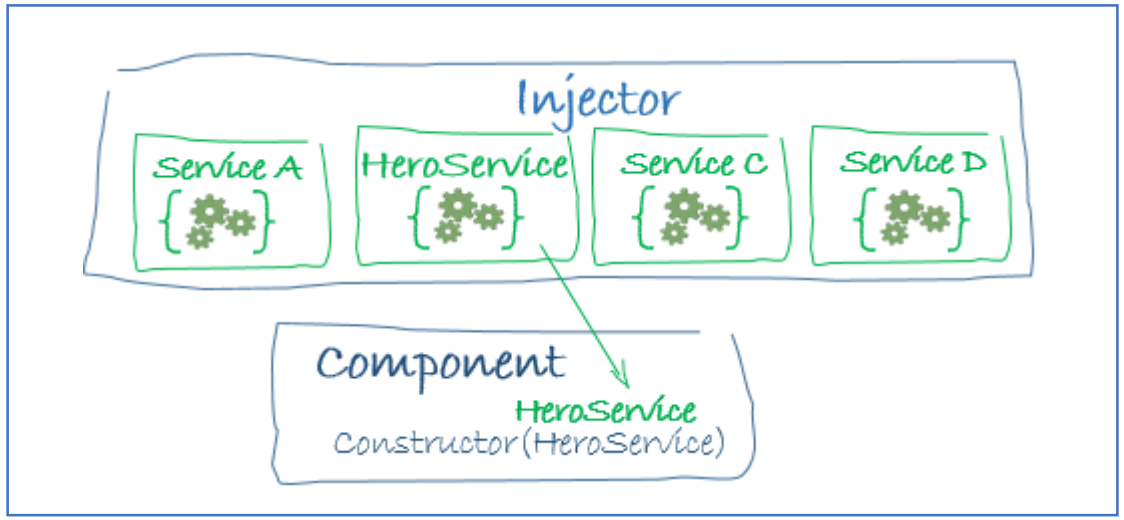
Componentes



- Un Template es un HTML que indica la estructura visual
- Una clase, formada por por propiedades, métodos, atributos y funciones que definen el comportamiento del componente.
- Metadata, que sirve para indicar un nombre mediante el cual referenciar al componente desde nuestro HTML.

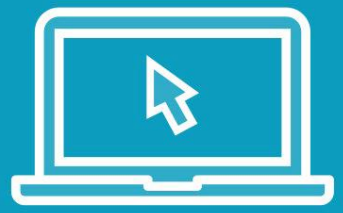
Component = Template + Propeties, Methods, fuctions + Metadata

Servicios



- Para los datos o la lógica que no están asociados a una vista específica y que desea compartir entre componentes, cree una clase de servicio.
- Una definición de clase de servicio es inmediatamente precedida por el decorador. El decorador proporciona los metadatos que permiten que otros proveedores se inyecten como dependencias en la clase. @Injectable()
- La inyección de dependencia (DI) le permite mantener sus clases de componentes eficientes. No recuperan datos del servidor, validan la entrada del usuario ni registran directamente en la consola; delegan tales tareas a los servicios.

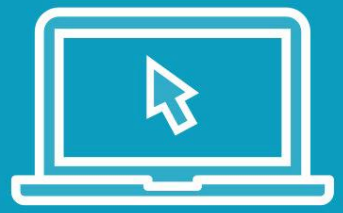
Demo



Creación del Proyecto.

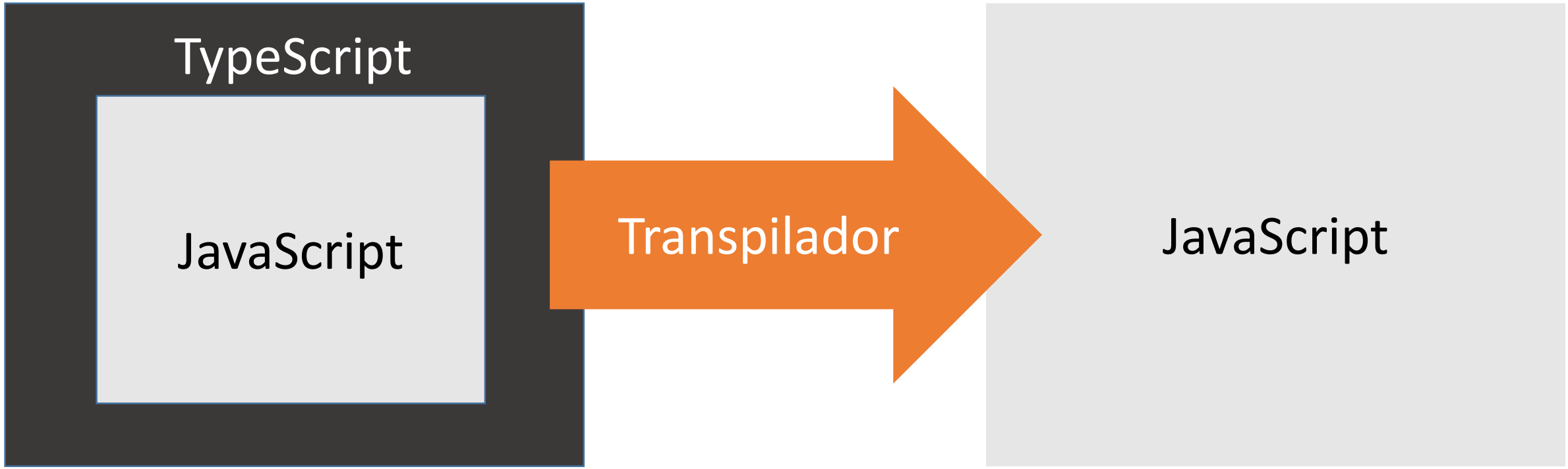
Configuración del proyecto.

Demo



Configuración del proyecto.

TypeScript, Definición



TypeScript, Definición

Static Typing

Interface

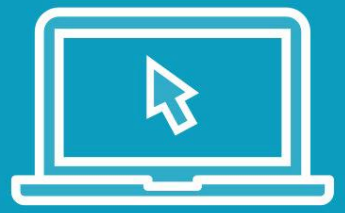
Class Properties

Accesibilidad Public / Private

Modelos usando TypeScript (clases, interfaces y herencia).

Basico

Demo



Clases, interfaces y herencia.


GALAXY
TRAINING