Desarrollo avanzado de aplicaciones II

Tema Nº9:Fundamentos de Angular.

Indicador de logro Nº9:

Aplica los conceptos fundamentales de un proyecto front-end utilizando como referencia a Angular.

**TEMA 01 Teoría de los**

Imagen que contiene Icono

Descripción generada automáticamente

**TEMA Nº9:**

**Fundamentos de Angular.**

**Subtema 9.1:**

Herramientas por utilizar (Postman, Visual Studio Code, NodeJS, Angular CLI).

**Ejemplos:**

Angular es un framework opensource desarrollado por Google para facilitar la creación y programación de aplicaciones web de una sola página, las webs SPA (Single Page Application).

Angular separa completamente el frontend y el backend en la aplicación, evita escribir código repetitivo y mantiene todo más ordenado gracias a su patrón MVC (Modelo-Vista-Controlador) asegurando los desarrollos con rapidez, a la vez que posibilita modificaciones y actualizaciones.

En una web SPA aunque la velocidad de carga puede resultar un poco lenta la primera vez que se abre, navegar después es totalmente instantáneo, ya que se ha cargado toda la página de golpe.

Solamente es una ruta la que se tiene que enviar el servidor, y Angular lo que hace ‘por debajo’ es cambiar la vista al navegar para que dé la apariencia de una web normal, pero de forma más dinámica.

Entre otras ventajas, este framework es modular y escalable adaptándose a nuestras necesidades y al estar basado en el estándar de componentes web, y con un conjunto de interfaz de programación de aplicaciones (API) permite crear nuevas etiquetas HTML personalizadas que pueden reutilizarse.

El lenguaje principal de programación de Angular es Typescript, y así toda la sintaxis y el modo de hacer las cosas en el código es el mismo, lo que añade coherencia y consistencia a la información, permitiendo por ejemplo, la incorporación de nuevos programadores, en caso de ser necesarios, ya que pueden continuar su trabajo sin excesiva dificultad.

Como ya se ha indicado, las plantillas de Angular almacenan por separado el código de la interfaz del usuario (front-end) y el de la lógica de negocio (back-end), que entre otros beneficios permite utilizar mejor otras herramientas anteriormente existentes.

Y por si fuera poco, los principales editores y entornos de desarrollo integrado (IDEs) ofrecen ya extensiones para poder trabajar con este framework con mayor comodidad.

Por su programación reactiva, la vista se actualiza automáticamente tras realizar los cambios. Además Angular dispone de asistente por línea de comandos para poder crear proyectos base y también se integra bien con herramientas de testing y con Ionic, lo que facilita la creación de web-responsive, es decir, adaptadas a móviles.

Este aspecto cada día adquiere mayor importancia tanto por el creciente uso de estos dispositivos para acceder a internet como por la penalización que Google realiza de aquellas páginas que no facilitan su visita en cualquier dispositivo.

Las herramientas por utilizar son los siguientes:

<https://code.visualstudio.com/download>

Imagen que contiene Sitio web

Descripción generada automáticamente

<https://www.postman.com/downloads/>

Interfaz de usuario gráfica, Sitio web

Descripción generada automáticamente

<https://nodejs.org/es/download/>

Interfaz de usuario gráfica, Sitio web, Escala de tiempo

Descripción generada automáticamente

**Subtema 9.2:**

Introducción a Angular y Angular CLI.

**Ejemplos:**

Este tema puede ayudarlo a comprender Angular: qué es Angular, qué ventajas ofrece y qué puede esperar al comenzar a construir sus aplicaciones.

Angular es una plataforma de desarrollo, construida sobre TypeScript. Como plataforma, Angular incluye:

* Un marco basado en componentes para crear aplicaciones web escalables
* Una colección de bibliotecas bien integradas que cubren una amplia variedad de características, que incluyen enrutamiento, administración de formularios, comunicación cliente-servidor y más.
* Un conjunto de herramientas para desarrolladores que lo ayudarán a desarrollar, compilar, probar y actualizar su código

Con Angular, está aprovechando una plataforma que puede escalar desde proyectos de un solo desarrollador hasta aplicaciones de nivel empresarial. Angular está diseñado para que la actualización sea lo más sencilla posible, así que aproveche los últimos desarrollos con un mínimo de esfuerzo. Lo mejor de todo es que el ecosistema Angular consta de un grupo diverso de más de 1,7 millones de desarrolladores, autores de bibliotecas y creadores de contenido.

**¿Qué es Angular CLI?**

Dentro del ecosistema de Angular encontramos una herramienta fundamental llamada "Angular CLI" (Command Line Interface). Es un producto que en el momento de escribir este artículo todavía se encuentra en fase beta, pero que ya resulta fundamental para el trabajo con el framework.

Angular CLI no es una herramienta de terceros, sino que nos la ofrece el propio equipo de Angular. En resumen, nos facilita mucho el proceso de inicio de cualquier aplicación con Angular, ya que en pocos minutos te ofrece el esqueleto de archivos y carpetas que vas a necesitar, junto con una cantidad de herramientas ya configuradas. Además, durante la etapa de desarrollo nos ofrecerá muchas ayudas, generando el "scaffolding" de muchos de los componentes de una aplicación. Durante la etapa de producción o testing también nos ayudará, permitiendo preparar los archivos que deben ser subidos al servidor, transpilar las fuentes, etc.

**Node y npm**

Angular CLI es una herramienta NodeJS, es decir, para poder instalarla necesitaremos contar con NodeJS instalado en nuestro sistema operativo.

Además, se instala vía "npm". Por npm generalmente no te tienes que preocupar, pues se instala al instalar NodeJS. No obstante, es importante que ambas versiones, tanto la de la plataforma Node como el gestor de paquetes npm, se encuentren convenientemente actualizados. En estos momentos como requisito nos piden tener Node 4 o superior

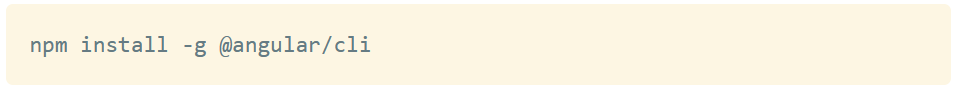
Para saber la versión de node instalada podemos utilizar los siguientes comandos

Patrón de fondo

Descripción generada automáticamente

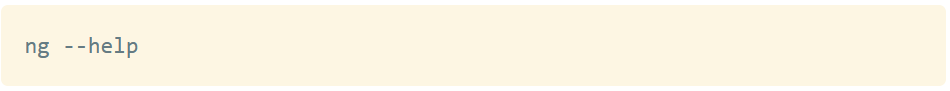
**Instalar Angular CLI**

Esto lo conseguimos desde el terminal, lanzando el comando:



Durante el proceso de instalación se instalará el propio Angular CLI junto con todas sus dependencias. La instalación puede tardar varios minutos dependiendo de la velocidad de tu conexión a Internet.

Una vez instalado dispondrás del comando "ng" a partir del cual lanzarás cualquiera de las acciones que se pueden hacer mediante la interfaz de comandos de Angular. Puedes comenzar lanzando el comando de ayuda:



Angular CLI es la forma más rápida, sencilla y recomendada de desarrollar aplicaciones en Angular. Angular CLI hace que una serie de tareas no presenten problemas. Aquí hay unos ejemplos:Texto

Descripción generada automáticamente

**Subtema 9.4:**

Crea tu primera aplicación en Angular.

**Ejemplos:**

Uno de los comandos que puedes lanzar con Angular CLI es el de creación de un nuevo proyecto Angular. Este comando se ejecuta mediante "new", seguido del nombre del proyecto que queramos crear.

Imagen que contiene Logotipo

Descripción generada automáticamente

Lanzado este comando se creará una carpeta igual que el nombre del proyecto indicado y dentro de ella se generarán una serie de subcarpetas y archivos que quizás por su número despisten a un desarrollador que se inicia en Angular 2. Si es así no te preocupes porque poco a poco nos iremos familiarizando con el código generado.

Además, como hemos dicho, se instalarán y se configurarán en el proyecto una gran cantidad de herramientas útiles para la etapa del desarrollo front-end. De hecho, gran cantidad de los directorios y archivos generados al crear un nuevo proyecto son necesarios para que estas herramientas funcionen. Entre otras cosas tendremos:

* Un servidor para servir el proyecto por HTTP
* Un sistema de live-reload, para que cuando cambiamos archivos de la aplicación se refresque el navegador
* Herramientas para testing
* Herramientas para despliegue del proyecto
* Etc.

Una vez creado el proyecto inicial podemos entrar en la carpeta con el comando cd.

Logotipo

Descripción generada automáticamente con confianza media

**Subtema 9.3:**

Estructura de un proyecto Angular

**Ejemplos:**

Una vez dentro de esa carpeta encontrarás un listado de archivos y carpetas similar a este:

Aplicación

Descripción generada automáticamente con confianza baja

Angular CLI lleva integrado un servidor web, lo que quiere decir que podemos visualizar y usar el proyecto sin necesidad de cualquier otro software. Para servir la aplicación lanzamos el comando "serve".

Imagen que contiene Logotipo

Descripción generada automáticamente

**Subtema 9.4:**

¿Qué son los componentes en Angular?

**Ejemplos:**

Los componentes son el bloque de construcción principal para las aplicaciones angulares. Cada componente consta de:

* Una plantilla HTML que declara lo que se representa en la página.
* Una clase de TypeScript que define el comportamiento.
* Un selector de CSS que define cómo se usa el componente en una plantilla
* Opcionalmente, estilos CSS aplicados a la plantilla.

Para crear un componente usando la CLI de Angular:

* Desde una ventana de terminal, navegue hasta el directorio que contiene su aplicación.
* Ejecute el comando **ng generate** component <component-name>, donde <component-name> es el nombre de su nuevo componente.

De forma predeterminada, este comando crea lo siguiente:

* Una carpeta con el nombre del componente.
* Un archivo de componente, <component-name> .component.ts
* Un archivo de plantilla, <component-name> .component.html
* Un archivo CSS, <component-name> .component.css
* Un archivo de especificación de prueba, <component-name> .component.spec.ts

Donde <component-name> es el nombre de su componente.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

**Subtema 9.5:**

Data Binding, String Interpolation, Property binding.

**Ejemplos:**

**String Interpolation:**

La interpolación se refiere a incrustar expresiones en texto marcado. De forma predeterminada, la interpolación utiliza las llaves dobles {{y}} como delimitadores.

Para ilustrar cómo funciona la interpolación, considere un componente angular que contiene una variable currentCustomer:

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

**Property Binding:**

Para enlazar a la propiedad de un elemento, enciérrelo entre corchetes, [], que identifica la propiedad como una propiedad de destino. Una propiedad de destino es la propiedad DOM a la que desea asignar un valor. Por ejemplo, la propiedad de destino en el siguiente código es la propiedad src del elemento de imagen.

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

**Event Binding:**

Para enlazar a un evento, usa la sintaxis de enlace de eventos Angular. Esta sintaxis consta de un nombre de evento de destino entre paréntesis a la izquierda de un signo igual y una declaración de plantilla entre comillas a la derecha. En el siguiente ejemplo, el nombre del evento de destino es click y la declaración de plantilla es onSave ().

Texto

Descripción generada automáticamente con confianza media

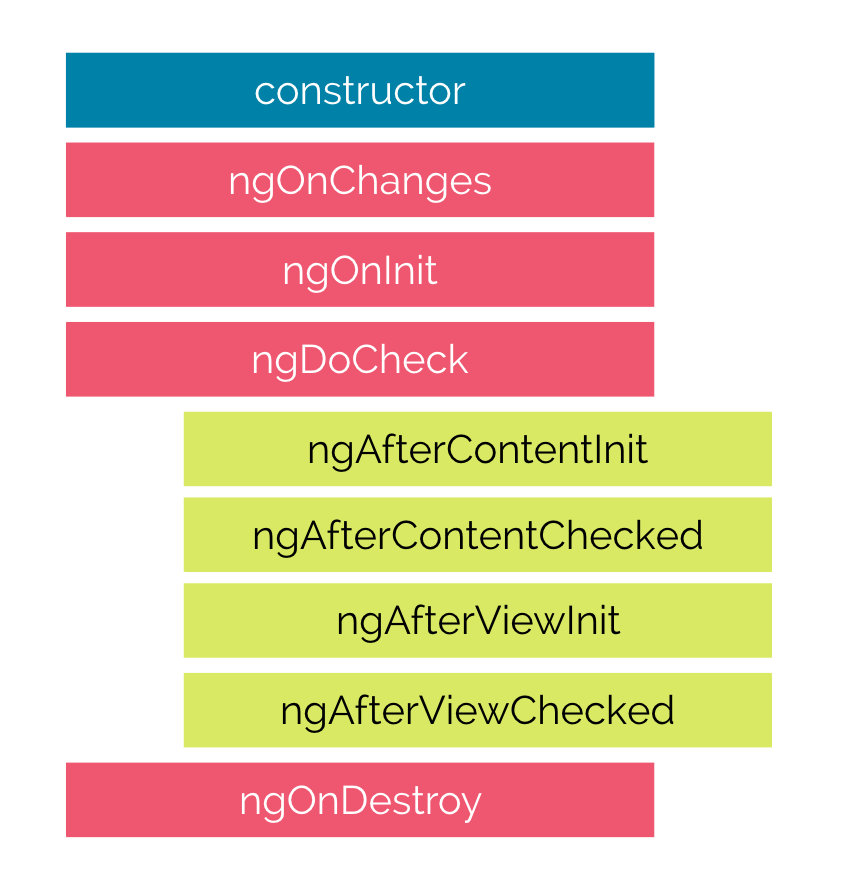
**Subtema 9.6:**

Ciclo de vida de los componentes.

**Ejemplos:**

Una instancia de componente tiene un ciclo de vida que comienza cuando Angular crea una instancia de la clase de componente y representa la vista del componente junto con sus vistas secundarias. El ciclo de vida continúa con la detección de cambios, ya que Angular verifica cuándo cambian las propiedades vinculadas a los datos y actualiza tanto la vista como la instancia del componente según sea necesario. El ciclo de vida termina cuando Angular destruye la instancia del componente y elimina su plantilla renderizada del DOM. Las directivas tienen un ciclo de vida similar, ya que Angular crea, actualiza y destruye instancias en el curso de la ejecución.

Su aplicación puede usar métodos de enlace de ciclo de vida para aprovechar eventos clave en el ciclo de vida de un componente o directiva con el fin de inicializar nuevas instancias, iniciar la detección de cambios cuando sea necesario, responder a las actualizaciones durante la detección de cambios y limpiar antes de eliminar las instancias.



Revisemos cada uno de los eventos:

* ngOnChanges: Este evento se ejecuta cada vez que se cambia un valor de un input control dentro de un componente. Se activa primero cuando se cambia el valor de una propiedad vinculada. Siempre recibe un change data map o mapa de datos de cambio, que contiene el valor actual y anterior de la propiedad vinculada envuelta en un SimpleChange
* ngOnInit: Se ejecuta una vez que Angular ha desplegado los data-bound properties(variables vinculadas a datos) o cuando el componente ha sido inicializado, una vez que ngOnChanges se haya ejecutado. Este evento es utilizado principalmente para inicializar la data en el componente.
* ngDoCheck: Se activa cada vez que se verifican las propiedades de entrada de un componente. Este método nos permite implementar nuestra propia lógica o algoritmo de detección de cambios personalizado para cualquier componente.
* ngAfterContentInit: Se ejecuta cuando Angular realiza cualquier muestra de contenido dentro de las vistas de componentes y justo después de ngDoCheck. Actuando una vez que todas las vinculaciones del componente deban verificarse por primera vez. Está vinculado con las inicializaciones del componente hijo.
* ngAfterContentChecked: Se ejecuta cada vez que el contenido del componente ha sido verificado por el mecanismo de detección de cambios de Angular; se llama después del método ngAfterContentInit. Este también se invoca en cada ejecución posterior de ngDoCheck y está relacionado principalmente con las inicializaciones del componente hijo.
* ngAfterViewInit: Se ejecuta cuando la vista del componente se ha inicializado por completo. Este método se inicializa después de que Angular ha inicializado la vista del componente y las vistas secundarias. Se llama después de ngAfterContentChecked. Solo se aplica a los componentes.
* ngAfterViewChecked: Se ejecuta después del método ngAfterViewInit y cada vez que la vista del componente verifique cambios. También se ejecuta cuando se ha modificado cualquier enlace de las directivas secundarias. Por lo tanto, es muy útil cuando el componente espera algún valor que proviene de sus componentes secundarios.
* ngOnDestroy: Este método se ejecutará justo antes de que Angular destruya los componentes. Es muy útil para darse de baja de los observables y desconectar los event handlers para evitar memory leaks o fugas de memoria.

**Actividad:**

a) CUESTIONARIO TÉCNICO

* ¿Qué es Angular?
* ¿Para qué sirve Node JS?
* ¿Qué comando se utiliza para instalar Angular CLI?
* Indique 3 comandos de Angular CLI.
* Cree un nuevo proyecto en Angular con 3 componentes.

b) CONCLUSIONES DE LA EXPERIENCIA

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_