Desarrollo avanzado de aplicaciones II

Tema Nº9:Angular Material, directivas, pipes y Ruteo.

Indicador de logro Nº9:

Diseña la interfaz gráfica del usuario, utiliza directivas y añade las rutas a la aplicación teniendo en cuenta las buenas prácticas del desarrollo front end.

**TEMA 01 Teoría de los**

Imagen que contiene Icono

Descripción generada automáticamente

**TEMA Nº10:**

**Angular Material, directivas, pipes y Ruteo**

**Subtema 10.1:**

Instalar Angular Material y Angular Flex Layout

**Ejemplos:**

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

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Comando para instalar angular material.



Angular Flex-Layout

<https://github.com/angular/flex-layout>

Una captura de pantalla de una computadora

Descripción generada automáticamente

Comando para instalar angular flex-layout

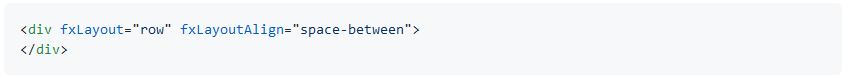


Agregar lo siguiente en el archivo app.module.ts

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Ejemplo:



**Subtema 10.2:**

Principales directivas de Angular.

**Ejemplos:**

Las directivas estructurales son responsables del diseño HTML. Dan forma o remodelan la estructura del DOM, generalmente agregando, eliminando y manipulando los elementos del host a los que están adjuntos.

Esta sección presenta las directivas estructurales integradas más comunes:

* NgIf: crea o elimina condicionalmente subvistas de la plantilla.



* NgFor: repite un nodo para cada elemento de una lista.



* NgSwitch: un conjunto de directivas que cambian entre vistas alternativas.

Tabla

Descripción generada automáticamente con confianza media

**Subtema 10.3:**

Uso de pipes de Angular.

**Ejemplos:**

Para aplicar una tubería, use el operador de tubería (|) dentro de una expresión de plantilla como se muestra en el siguiente ejemplo de código, junto con el nombre de la tubería, que es la fecha para el DatePipe integrado. Las pestañas del ejemplo muestran lo siguiente:

* app.component.html usa la fecha en una plantilla separada para mostrar un cumpleaños.
* hero-birthday1.component.ts usa la misma tubería como parte de una plantilla en línea en un componente que también establece el valor de cumpleaños.

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente





**Subtema 10.4:**

Inyección de dependencias

**Ejemplos:**

Las dependencias son servicios u objetos que una clase necesita para realizar su función. La inyección de dependencia, o DI, es un patrón de diseño en el que una clase solicita dependencias de fuentes externas en lugar de crearlas.

El marco DI de Angular proporciona dependencias a una clase en la instanciación. Puede utilizar Angular DI para aumentar la flexibilidad y modularidad en sus aplicaciones.

Para generar una nueva clase HeroService en la carpeta src / app / heroes, use el siguiente comando CLI de Angular.



Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

La inyección de servicios hace que sean visibles para un componente.

Para inyectar una dependencia en el constructor de un componente (), proporcione un argumento de constructor con el tipo de dependencia. El siguiente ejemplo especifica HeroService en el constructor HeroListComponent. El tipo de heroService es HeroService.

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

Descripción generada automáticamente

**Subtema 10.5:**

Configurar redireccionamientos.

En una aplicación de una sola página, cambia lo que ve el usuario mostrando u ocultando partes de la pantalla que corresponden a componentes particulares, en lugar de ir al servidor para obtener una nueva página. A medida que los usuarios realizan tareas de la aplicación, necesitan moverse entre las diferentes vistas que ha definido.

Para manejar la navegación de una vista a la siguiente, usa el enrutador angular. El enrutador permite la navegación interpretando la URL de un navegador como una instrucción para cambiar la vista

**Ejemplos:**

Texto

Descripción generada automáticamente

**Subtema 10.6:**

Evitar el acceso no autorizado.

**Ejemplos:**

Interfaz que una clase puede implementar para ser un guardia que decide si se puede activar una ruta. Si todos los guardias vuelven verdaderos, la navegación continúa. Si algún guardia devuelve falso, se cancela la navegación. Si algún guardia devuelve un UrlTree, la navegación actual se cancela y comienza una nueva navegación al UrlTree devuelto por el guardia.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

El siguiente ejemplo implementa una función CanActivate que comprueba si el usuario actual tiene permiso para activar la ruta solicitada.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

**Actividad:**

a) CUESTIONARIO TÉCNICO

* ¿Qué es Angular Material?
* ¿Qué es Angular Flex-Layout?
* Describa 2 directivas de angular.
* ¿Para qué sirve PIPES?
* Cree un nuevo proyecto en Angular utilice directivas, pipes e inyección de dependencias.

b) CONCLUSIONES DE LA EXPERIENCIA

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