

TEST 1 - PREGUNTAS ANGULAR

¿Qué es Angular?

Angular es un framework mantenido por [Google](#) que nos permite hacer Single Page Applications (SPA) en [frontend](#), es decir, aplicaciones web en entorno cliente. La principal característica de una SPA es que el usuario no observa recargas de la página web, de forma que obtiene una mejor experiencia de usuario y puede acceder a la información de forma más rápida.

¿Qué ventajas me da un framework como Angular respecto a hacer la misma aplicación web en Vanilla JavaScript?

Para empezar, Angular usa TypeScript, lo que otorga más robustez a mi aplicación web al tener yo un mejor control de los datos gracias al tipado que me ofrece TypeScript. Por otro lado, al ser Angular un framework, puedo utilizar sus funcionalidades para ahorrarme escribir código (por ejemplo, puedo validar campos de un formulario de forma muy sencilla con una línea de código, mientras que con Vanilla JavaScript escribiría este código de forma manual, consumiendo más tiempo y líneas de código)-lo cual ayuda a que el código sea más mantenible y limpio. Para finalizar, una de las grandes ventajas que yo aprecio de Angular es su estructura: al obligarme a programar con la estructura estándar del framework, yo ya sé qué me puede ofrecer un servicio, qué me puede ofrecer un módulo... de modo que **obtengo código más eficiente, limpio y mantenible que programando la misma aplicación web en Vanilla [JavaScript](#).**

```
constructor(private gifsService:GifsService){}

search(){

    const value=this.txtSearch.nativeElement.value;

    if(value.trim().length===0){
        return ;
    }
    this.gifsService.searchGifs(value);
    this.txtSearch.nativeElement.value='';
}

}
```

¿Qué es una inyección de dependencias en Angular?

Mediante una inyección de dependencias yo puedo inyectar a mi componente una dependencia para utilizar una funcionalidad en el mismo, de un servicio, por ejemplo. Imaginemos que tenemos un servicio que llama a una API que me ofrece información meteorológica. Yo tendré la llamada a la API en el servicio, ahí recojo los datos, y en mi componente tengo el HTML y el archivo de TS dónde voy a manejar la lógica de dicho componente. Digamos que dicho componente simplemente va a mostrar los datos que reciba de la API. Al inyectar la dependencia del servicio tendré acceso a sus métodos, de forma que en el componente de TS puedo llamar al método, guardar la información recibida de la API en una variable y mostrarla en el HTML de la forma que considere más conveniente.

¿Qué es un componente en Angular?

Una aplicación de Angular es como un puzzle, dónde yo integraré los distintos componentes de mi aplicación, que serían las piezas de dicho puzzle. Por ejemplo, un componente puede ser el menú de mi aplicación web, y en ella tendré como mínimo un fichero de HTML (la vista de mi componente) y un fichero de TypeScript (dónde ejecuto la lógica de mi componente). También es muy normal tener un fichero de estilos (con CSS, por ejemplo) y otro fichero de testing unitario, dónde compruebo que este componente funciona efectivamente como yo lo he programado para que funcione. Según como tenga estructurada mi aplicación web, en dicho componente puedo tener también

otro tipo de ficheros, pero digamos que estos son los ficheros más básicos que van a conformar un componente de Angular.

```
Angular CLI

Angular CLI: 13.0.1
Node: 14.16.1
Package Manager: npm 6.14.12
OS: win32 x64

Angular:
...

Package                                Version
-----
@angular-devkit/architect              0.1300.1 (cli-only)
@angular-devkit/core                   13.0.1 (cli-only)
@angular-devkit/schematics             13.0.1 (cli-only)
@schematics/angular                   13.0.1 (cli-only)
```

¿Qué es Angular CLI?

Es una interfaz de línea de comando que yo puedo utilizar para crear mis proyectos de Angular de forma rápida y sencilla, con el siguiente comando: `ng new nombreProyecto`. Al crear el proyecto me solicitará distintas configuraciones, como si quiero tipado estricto, si utilizaré para los estilos CSS/SASS...

¿Qué es una pipe en Angular? ¿Podrías poner un ejemplo?

Una pipe en Angular es una herramienta que nos permite modificar de forma visual la información que mostramos al usuario. Es importante resaltar que no modifica el valor que tiene una determinada variable, sino simplemente cómo se le presenta al usuario. Supongamos que tenemos información recogida de un usuario mediante un formulario, y dicho usuario se llama “*pedro gómez ramírez*”. El usuario, en un descuido, ha puesto toda su información en minúscula. Pero yo quiero que la persona que tiene que consultar esta información, la consulte cómo debería de mostrarse, es decir, con la primera letra del nombre y del apellido en mayúscula. Para ello puedo utilizar un pipe, en este caso sería el

pipe “TitleCase”. Con este pipe, aunque en mi programa el nombre de dicho usuario seguiría mal escrito, al mostrárselo a otro usuario que consulte los nombres, leerá “Pedro Gómez Ramírez” porque el pipe ha puesto la primera letra de cada palabra en mayúscula y el resto en minúsculas, mejorando la experiencia de usuario.

¿Qué es una directiva en Angular? Pon algunos ejemplos

Las directivas de Angular me permiten modificar el HTML de forma muy sencilla. Existen directivas, las llamadas **Directivas de Atributo**, que simplemente me permiten modificar el estilo de un elemento del DOM (por ejemplo, cuando uso Angular Material y añado *mat-button* como atributo al botón de mi HTML, realmente lo que estoy es aplicándole una directiva, que me aplicará los estilos de Angular Material correspondientes). **Existen otras directivas que me permiten modificar la estructura del DOM y por ello se llaman Directivas Estructurales**. Por ejemplo, con un `*ngIf` aplico renderizado condicional sobre un elemento del DOM, que se mostrará en función de la condición que hayamos aplicado a la directiva.

```
const routes:Routes=[

  {
    path: '',
    component: HomeComponent,
    children: [
      {
        path: 'list',
        component: ListComponent
      },
      {
        path: 'add',
        component: AddComponent
      },
    ]
  }
]
```

¿Qué es Lazy Load en Angular?

Lazy Load es un patrón de diseño que me permite que sólo cargue en un determinado momento lo que el usuario necesita en dicho momento, optimizando por tanto el rendimiento de mi aplicación web. A día de hoy es un

estándar a la hora de desarrollar aplicaciones web de entorno cliente con Angular.

Pon un ejemplo de cómo mostrarías el valor de una variable del componente TS en el HTML de un componente de Angular.

Por ejemplo, yo puedo tener en mi código una variable que sea *name* y el valor de dicha variable sea *Pedro*. Puedo mostrarla en mi HTML mediante interpolación, pongamos que la quiero mostrar con un H1, de la siguiente forma: `<h1> {{name}}</h1>`. **Mediante la interpolación, muestro de forma dinámica el valor de una variable de mi fichero TypeScript de mi componente de Angular en mi fichero HTML y, por tanto, ofrezco esa información al usuario gracias a la interfaz gráfica que proporciona el HTML.**

¿Para qué sirve ngOnInit? Pon un ejemplo

Es un hook o función que forma parte del ciclo de vida de los componentes de Angular, y se ejecuta una vez que el componente está creado. Todo el código que hayamos escrito dentro del `ngOnInit` se ejecutará en ese momento. Podemos usarlo para inicializar variables, llamar a métodos que nos traigan datos mediante una petición HTTP a una API para guardarlos en variables y mostrarlos en el HTML de nuestro componente etc.

TEST 2 - PREGUNTAS ANGULAR

¿Qué es Angular?

Es un marco de aplicación web de código abierto basado en TypeScript. Ha sido desarrollado y mantenido por Google como un lenguaje fácil y poderoso para construir aplicaciones front-end en la web. Este lenguaje que integra características como inyección de dependencia, plantillas declarativas, herramientas integrales y otras herramientas que facilitan el desarrollo de aplicaciones web.

Menciona algunas características de Angular

Al ser un marco JavaScript front-end ideal, Angular es capaz de:

- Crear aplicaciones de accesibilidad: utiliza componentes habilitados para ARIA, una infraestructura de prueba integrada y guías para desarrolladores.
- Proporciona soporte para herramientas de interfaz de línea de comandos: se pueden usar para agregar componentes, implementación instantánea, pruebas, etc.
- Brinda soporte de animación: su API intuitiva le permite crear líneas de tiempo de animación complejas de alto rendimiento con poco código.
- Se utiliza para el desarrollo de aplicaciones multiplataforma: Angular te permite crear aplicaciones web de escritorio, nativas y progresivas eficientes y potentes. Proporciona soporte para Ionic, Cordova o NativeScript, utilizando capacidades modernas de plataforma

web. También se puede usar para crear aplicaciones de escritorio para Windows, Linux o macOS.

- Generación de código: puede convertir plantillas en código altamente optimizado, para ser utilizado en máquinas virtuales JavaScript modernas.
- División de código: gracias a su enrutador de componentes, Angular tiene división automática de código, por lo que las aplicaciones se cargan más rápido.
- Haga sinergia con los editores de código e IDE populares: Angular agrega la finalización del código, los errores instantáneos y más con los editores de código e IDE.
- Hacer plantillas: crea vistas de interfaz de usuario con sintaxis de plantilla.
- Pruebas: Al usar Karma, Angular permite pruebas unitarias frecuentes y Protractor permite ejecutar pruebas de escenarios estables.

**Demuestra tus tech skills en
nuestros challenges y conecta
con empresas TOP**

rviewer.
by GetWith

¿Cuáles son las ventajas y desventajas de usar Angular?

Entre las ventajas de usar Angular están:

- Soporte para inyección de dependencias, validaciones y servicios RESTful
- Soporte para enlace de datos bidireccional
- Admite plantillas estáticas y angulares.
- Utiliza la arquitectura de patrón MVC
- Características sólidas como animación y controladores de eventos
- Buena comunicación con el cliente y el servidor.
- Gran apoyo de la comunidad.
- Posibilidad de agregar directivas personalizadas

Por otro lado, las desventajas de usar Angular son:

- Angular requiere esfuerzo y tiempo para ser aprendido
- Las aplicaciones dinámicas a veces funcionan mal
- Los SPA complejos son inconvenientes y lentos.

Algunas Características de Angular 7

La principal diferencia entre Angular 7 y las versiones anteriores es que esta versión viene con división en @ angular / núcleo para reducir el tamaño. Como no se requieren todos los módulos, en Angular 7 cada división no tendrá más de 418 módulos.

Otras diferencias son la función de arrastrar y soltar y el desplazamiento virtual, así como la descarga de elementos del DOM. Además, Angular 7 viene con una nueva versión del compilador ng.

¿Cuál es la diferencia entre Angular y AngularJS?

- En arquitectura: AngularJS es compatible con el modelo de diseño MVC. Angular se basa en componentes y directivas.
- In Dependency Injection: Angular admite una inyección de dependencia jerárquica con detección de cambios basada en árboles. AngularJS no admite la inyección de dependencias.
- En la sintaxis de expresión: en AngularJS, se requiere una directiva ng específica mientras Angular usa () y [] para cegar un evento y lograr el enlace de propiedad.
- En Soporte móvil: AngularJS no tiene dispositivos móviles.
- En lenguaje recomendado: AngularJS se basa en JavaScript. Angular se basa en TypeScript.
- En Enrutamiento: AngularJS usa \$routeProvider.when (). Mientras tanto, Angular usa @RouteConfig {...}
- En rendimiento: el enlace de datos bidireccional en AngularJS reduce el esfuerzo y el tiempo, pero Angular es más rápido debido a las características mejoradas.
- En Estructura: Angular proporciona un desarrollo y mantenimiento más fácil de grandes aplicaciones.
- En Soporte: No hay soporte oficial o actualizaciones para AngularJS. Por el contrario, Angular tiene soporte activo con actualizaciones.

¿Cuál es la diferencia entre Angular y backbone.js?

Hay varias diferencias principales:

- En tipo: Angular es un marco de aplicación web front-end de código abierto basado en JS. Backbone.js es una biblioteca ligera de JavaScript con una interfaz JSON RESTful y un marco MVP.
- En el Enfoque para la prueba: en Angular, se prefiere la prueba unitaria y el proceso de prueba es más fluido. Backbone.js permite una prueba rápida para una sola página y pequeñas aplicaciones.
- En plantillas: aunque Angular admite plantillas a través de atributos HTML dinámicos, Backbone.js utiliza plantillas Underscore.js que no tienen todas las funciones.
- In Performance: Angular ofrece un buen rendimiento para proyectos pequeños y grandes. Backbone.js puede sobresalir en pequeños conjuntos de datos o páginas web, pero no se recomienda para páginas web o conjuntos de datos más grandes, ya que no tiene un proceso de enlace de datos.

- En DOM: El enfoque de Angular JS está en HTML válido y elementos dinámicos para reconstruir el DOM. Por el contrario, Backbone.js sigue el enfoque directo de manipulación DOM.
- En enlace de datos: mientras Angular usa un proceso de enlace de datos bidireccional, Backbone.js tiene una API simplista.
- En Soporte Comunitario: Dado que Angular está respaldado por Google, tiene un gran soporte comunitario y una extensa documentación. Backbone.js tiene un buen soporte de la comunidad, pero carece de documentación ampliamente accesible.
- En arquitectura: Backbone.js utiliza la arquitectura MVP, mientras que Angular trabaja en la arquitectura MVC. Por lo tanto, utiliza el enlace de datos bidireccional para la actividad de la aplicación.

Principales preguntas técnicas sobre Angular

A continuación, debes enfrentar algunas cuestiones más técnicas. También debes estar preparado para codificar sobre la marcha si es necesario.

¿Cuáles son los bloques de construcción de Angular?

Hay 9 bloques de construcción en una aplicación angular:

- Componentes: controla una o más vistas. Cada aplicación angular tiene al menos un componente raíz dentro del módulo principal, conocido como módulo raíz.
- Enlace de datos: cómo las partes de una plantilla se coordinan con las partes de un componente, agregadas a la plantilla HTML.
- Inyección de dependencias: para proporcionar dependencias requeridas a nuevos componentes. La inyección de dependencia suministra dependencias completamente formadas requeridas por una nueva instancia de una clase.
- Directivas: responsables de instruir a Angular sobre cómo transformar el DOM al representar una plantilla.
- Metadatos: le dicen a Angular cómo procesar una clase.
- Módulos: AKA NgModules, un módulo es un bloque de código organizado con un conjunto específico de capacidades, dentro de un dominio de aplicación específico o un flujo de trabajo.
- Enrutamiento: responsable de interpretar la URL de un navegador como una instrucción para navegar a una vista generada por el cliente.
- Servicios: una amplia categoría de cualquier cosa, desde un valor y función hasta una característica, el Servicio es una clase con un propósito bien definido.
- Plantilla: una forma de etiquetas HTML que le permite a Angular saber cómo se debe representar el componente.

¿Qué es el enlace de datos?

El enlace de datos es la forma de conectar los datos de la aplicación con el DOM (Modelo de objetos de datos). Hay 3 formas de hacerlo:

Enlace de eventos: la aplicación responde a la entrada del usuario en el entorno de destino.

Enlace de propiedad: interpola los valores calculados a partir de los datos de la aplicación en el HTML.

Enlace bidireccional: los cambios realizados en el estado de la aplicación se reflejan en la vista y viceversa. La directiva ngModel se utiliza para lograr esto.

Definir la Directiva de contenido ng

Los elementos HTML comunes tienen contenido entre etiquetas. En Angular, la Directiva de contenido ng se utiliza para construir componentes reutilizables.

¿Qué es un «servicio» en Angular?

Los servicios en Angular son objetos Singleton instanciados solo una vez durante la vida útil de una aplicación. En un servicio Angular hay métodos que mantendrán los datos durante la vida de una aplicación, para organizar y compartir la lógica empresarial, datos y funciones o modelos con varios componentes de una aplicación Angular.

En Angular, ¿qué es la interpolación de cadenas?

También llamada «sintaxis de bigote», la interpolación de cadenas en Angular es una sintaxis especial que utiliza expresiones de plantilla para mostrar los datos del componente, encerrados entre llaves dobles, es decir, `{{}}`.

Dentro de las llaves, podemos ejecutar expresiones de JavaScript e incrustar la salida en el código HTML. Estas expresiones se actualizan y registran como parte del ciclo de resumen.

¿Qué es la autenticación y autorización angular?

En el servidor, hay una API de autenticación que recibe las credenciales de inicio de sesión del usuario. Luego, la validación posterior del lado del servidor de las credenciales se devuelve en un JWT (JSON Web Token). Este token tiene información o atributos relacionados con el usuario actual, que se llama autenticación. Cuando inicie sesión correctamente, cada usuario tendrá un nivel de acceso diferente, que es su autorización.

¿Qué es la jerarquía de alcance en Angular?

La jerarquía de alcance en Angular es cómo se organizan los objetos de alcance en una jerarquía, utilizada por las vistas. Tiene un ámbito raíz que puede contener uno o varios ámbitos, llamados ámbitos secundarios.

Cada vista tiene su propio alcance. Luego, las variables establecidas por el controlador de vista de una vista permanecerán ocultas para los demás.

Definir el concepto de plantillas en Angular.

Las plantillas en Angular se escriben con HTML con atributos y elementos específicos de Angular. Luego, combinado con la información del controlador y el modelo, se procesan para atender al usuario con la vista dinámica.

En Angular, ¿cuál es la diferencia entre una anotación y un decorador?

Las anotaciones se utilizan para crear una matriz de anotaciones, que es un conjunto de metadatos de la clase que utiliza la biblioteca Reflect Metadata. Los decoradores son patrones de diseño utilizados para separar la decoración o modificación de alguna clase, sin cambiar el código fuente original.

¿Qué son las directivas en angular?

Las directivas permiten a un desarrollador Angular escribir una nueva sintaxis HTML específica de la aplicación, como funciones ejecutadas por el compilador Angular. Las directivas pueden ser de atributo, componente o estructural.

¿Qué es el Angular Material?

Angular Material es una biblioteca de componentes de UI que ayuda a crear páginas web y aplicaciones web atractivas, completamente funcionales y consistentes.

¿Qué es la compilación AOT (por adelantado)?

Cada aplicación Angular se compila internamente, cuando el compilador Angular toma el código JS, lo compila y luego produce un nuevo código JS solo una vez por ocasión por usuario.

TEST 3 - PREGUNTAS ANGULAR

P # 1) ¿Qué entiende por AngularJS?

Responder: AngularJS es un marco de JavaScript que se utiliza para crear aplicaciones web ricas y extensibles.

Se ejecuta en JavaScript y HTML sin formato, por lo que no necesita ninguna otra dependencia para que funcione. AngularJS es perfecto para aplicaciones de página única (SPA). Básicamente se utiliza para vincular objetos JavaScript con elementos de la interfaz de usuario HTML.

Q # 2) Defina las características de AngularJS.

Respuesta: Las características incluyen:

- La plantilla (ver)
- El alcance (modelo)
- El controlador (controlador)
- Servicios
- Filtros
- Directivas

P # 3) Definir el enlace de datos.

Responder: La vinculación de datos es una sintonización automática de datos entre la vista y los componentes del modelo.

P # 4) Distinga entre AngularJS y expresiones JavaScript.

Respuesta: Existen varias diferencias entre AngularJS y las expresiones JavaScript:

- Podemos escribir expresiones AngularJS en HTML, pero no podemos escribir expresiones JavaScript en HTML.
- No podemos usar iteraciones condicionales, bucles y excepciones en AngularJS, pero podemos usar todas estas propiedades condicionales en expresiones JavaScript.
- Los filtros son compatibles con AngularJS, mientras que los filtros no son compatibles con JavaScript.

P # 5) Escriba todos los pasos para configurar un nuevo Aplicación angular (en la aplicación).

Respuesta: Para configurar una aplicación angular debemos seguir ciertos pasos como se menciona a continuación:

- angular.module se creará al principio.
- Se asignará un controlador al módulo.
- El módulo se vinculará con la plantilla HTML (es decir, UI o View) con una aplicación angular (ng-app).
- La plantilla HTML se vinculará con el controlador (es decir, JS) con una directiva ng-controller.

P # 6) ¿Qué son los módulos angulares?

Responder: Los módulos angulares definen colectivamente una aplicación angular donde podemos escribir el código angular. Los módulos contienen las diferentes partes de una aplicación angular. Un módulo se crea mediante la función angular.module en angular.

[Es Soporte Técnico Preguntas Y Respuestas De La Entrevista](#)

P # 7) ¿Cuáles son los ámbitos de directiva en AngularJS?

Responder: Hay tres ámbitos de directiva disponibles en AngularJS.

Son:

- **Alcance principal:** Cualquier cambio que realice en su directiva que provenga del ámbito principal, también se reflejará en el ámbito principal y también es un ámbito predeterminado.
- **Alcance del niño:** Es un ámbito anidado que hereda una propiedad del ámbito principal. Además, si las propiedades y funciones del ámbito no están conectadas con la directiva de ámbito principal, se crea una nueva directiva de ámbito secundario.
- **Alcance aislado:** Es reutilizable y se utiliza cuando creamos una directiva autónoma. Solo se utiliza para uso privado e interno, lo que significa que no contiene ninguna propiedad del ámbito principal.

P # 8) ¿Cómo podemos compartir los datos entre controladores en AngularJS?

Responder: Primero, tenemos que crear un servicio. El servicio se utiliza para compartir los datos entre controladores en AngularJS de una manera muy lúcida, fácil y rápida. Usamos eventos, \$ parent, next sibling y controller usando \$rootScope.

P # 9) ¿Cuál es el ciclo de resumen en AngularJs?

Responder: Es parte del proceso de enlace de datos en AngularJS. Compara las versiones antigua y nueva del valor del modelo de alcance en cada ciclo de resumen.

El ciclo de resumen se activa automáticamente. También podemos mejorar la usabilidad usando \$apply () si queremos activar el ciclo de resumen manualmente.

P # 10) Explique las diferencias entre el enlace unidireccional y el enlace bidireccional.

Responder: La vinculación unidireccional se utiliza para vincular los datos del modelo para verlos sin actualizar la plantilla HTML o verlos automáticamente.

Por lo tanto, para actualizar la plantilla HTML, necesitamos escribir un código personalizado que actualice la vista cada vez que se enlacen datos de un modelo a otra.

Considerando que, la vinculación bidireccional se utiliza para vincular los datos del modelo a la vista y viceversa (es decir, vista a modelo) actualizando automáticamente la plantilla HTML sin escribir ningún código personalizado.

P # 11) Diferencia entre sessionStorage, cookies y localStorage.

Respuesta: Las diferencias son las siguientes:

- **SessionStorage** - Los datos se almacenan para una sesión en particular. Los datos se perderán cada vez que se cierre la pestaña del navegador o después de alguna sesión en particular. El tamaño máximo almacenado puede ser de hasta 5 MB.
- **Almacenamiento local** - Los datos se almacenan sin fecha de vencimiento. Los datos solo pueden borrarse mediante JavaScript o borrando la memoria caché del navegador. El límite de almacenamiento es máximo que el sessionStorage y la cookie.
- **Galletas** - Almacena los datos que deben enviarse de vuelta al servidor con algunas solicitudes. La caducidad de la cookie varía según el tipo y la duración configurados del lado del servidor o del lado del cliente. El tamaño máximo almacenado puede ser inferior a 4 KB.

P # 12) ¿Cuál es el rol de \$routeProvider en AngularJS?

Responder: Es el \$routeProvider que ayuda a navegar entre diferentes páginas / enlaces sin cargar por separado la página / enlace cada vez que un usuario hace clic en un enlace.

El método ngRoute config () se utiliza para configurar routeProvider.

P # 13) ¿Cuál es la diferencia entre \$scope y scope?

Responder: En AngularJS, \$scope se usa para lograr la inyección de dependencia y el alcance se usa para vincular entre View (es decir, HTML) y Controller (es decir, JS).

P # 14) ¿Cómo se usan los prefijos \$ y \$\$ de AngularJS?

Responder: La variable \$\$ en AngularJS se usa como una variable privada, ya que se usa para evitar la colisión accidental del código con el código del usuario.

Considerando que, \$ prefix se puede usar para denotar funcionalidades del núcleo angular (como una variable, parámetro, propiedad o método).

P # 15) ¿Dónde podemos implementar la manipulación DOM en AngularJS?

Responder: La manipulación de DOM está en las directivas y, aparte de esto, no debería existir en los servicios del controlador ni en ningún otro lugar.

P # 16) ¿Cómo podemos mostrar que una variable de ámbito debe tener un enlace único?

Responder: Para mostrar la vinculación de una sola vez, debemos usar ' :: "Delante del visor.

P # 17) ¿Qué es SPA (aplicación de página única) en AngularJS?

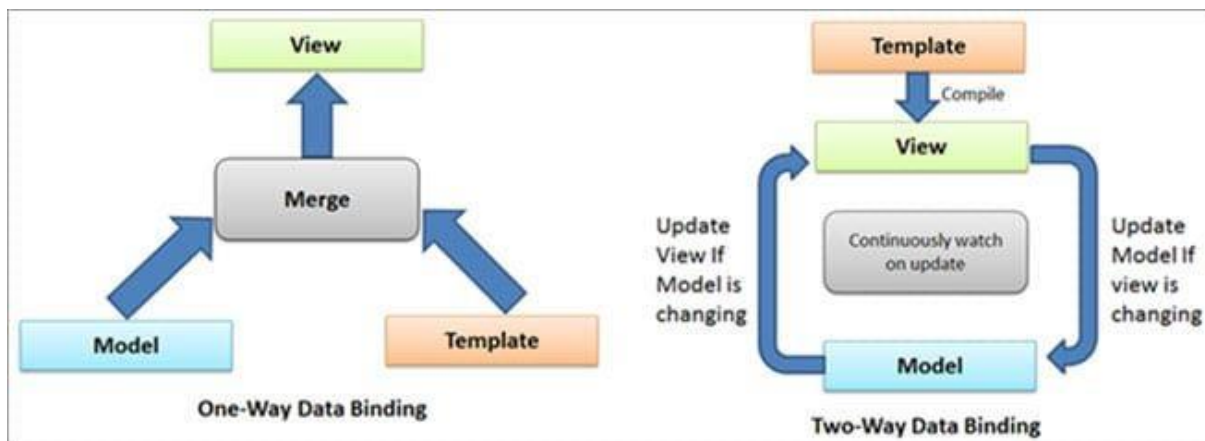
Responder: Es una aplicación web que carga una sola página HTML y actualiza dinámicamente la página a medida que el usuario se conecta con la aplicación.

Mediante el uso de AJAX y HTML, SPA puede crear una aplicación web fluida y receptiva sin recargas de página invariables. A través de esto, podemos hacer una interfaz de usuario receptiva sin parpadeo de página.

P # 18) ¿Cuántos tipos de enlaces de datos hay en AngularJS?

Responder: AngularJS admite enlaces unidireccionales y bidireccionales.

De una forma vinculante si cambiamos el modelo de datos, entonces no habrá ningún cambio dinámico que verá a la vista, pero en la vinculación bidireccional habrá un cambio dinámico siempre que se realice un cambio en el modelo de datos.



P # 19) ¿Cuáles son las directivas vinculantes en AngularJs?

Respuesta: Las directivas vinculantes incluyen:

- ng-bind
- ng-bind-html
- ng-bind-template
- de no enlazable

- ng-modelo

P # 20) Explique las directivas ng-bind y ng-bind-html.

Responder:

ng-bind : Es una directiva que reemplaza el contenido del elemento HTML con el valor de la variable o expresión asignada.

El contenido del elemento HTML cambiará al cambiar el valor de la variable o expresión.

Es como {{{expresión}}} y la sintaxis para esto es,

ng-bind-html : Es una directiva que vincula el contenido al elemento HTML (vista) de forma segura. El servicio \$ sanitize se utiliza para desinfectar el contenido y enlazarlo en un elemento HTML. Para hacer esto, 'angular-sanitize.js' debe estar incluido en nuestra aplicación.

Sintaxis para escribir esto,

P # 21) Explique ng-bind-template y ng-non-bindable.

Responder:

ng-bind-template : Reemplaza el contenido de texto del elemento por interpolación de la plantilla. Puede contener múltiples marcas de doble rizado.

No enlazable : Especifica AngularJS para no compilar el contenido de este elemento HTML y sus nodos secundarios.

Q # 22) Explique la directiva ng-model en AngularJs.

Responder: Esto puede ser un salto con el control de formulario de entrada HTML personalizado (como entrada, área de texto y selección) a los datos de la aplicación. Proporciona un comportamiento de validación de formularios con enlace bidireccional.

|_+_|

Q # 23) Defina el método Factory en AngularJS.

Responder: Es bastante similar al servicio, las fábricas implementan un patrón de módulo en el que usamos un método de fábrica para generar un objeto que se usa para construir modelos.

En una fábrica, un objeto de método se devuelve al final creando un nuevo objeto y agregando funciones como propiedades.

Sintaxis :

|_+_|

P # 24) ¿Qué es la directiva ng-repeat en AngularJS?

Responder: Representa o itera sobre una colección de elementos y crea elementos DOM. Supervisa periódicamente la fuente de datos para volver a representar una plantilla en respuesta a un cambio.

Sintaxis:

|_+_|

P # 25) ¿Qué es un controlador en AngularJS?

Responder: Un controlador es una función de JavaScript que está vinculada al ámbito especificado. Angular crea una instancia del nuevo objeto controlador e inyecta el nuevo alcance como una dependencia.



Se puede usar un controlador para configurar el estado inicial del objeto de alcance y para agregar comportamiento al objeto. No se puede usar un controlador para compartir código o estado entre controladores, pero en lugar de ese servicio Angular se puede usar.

P # 26) ¿Qué son los filtros en AngularJS?

Responder: El trabajo principal de los filtros es modificar los datos, de modo que se puedan fusionar en una expresión o directiva mediante el uso de un carácter de tubería (se utiliza para aplicar filtros en un símbolo angular de una tubería que es `|` o este es el símbolo).

Un filtro formatea el valor de una expresión para mostrar al usuario. Se pueden usar en plantillas de vista, controladores o servicios, y también podemos crear fácilmente nuestro propio filtro. Un filtro es un módulo proporcionado por AngularJS. Hay nueve componentes de un filtro que proporciona.

Ejemplos: moneda, fecha, filtro, JSON, limitTo, etc.

P # 27) ¿Qué es la directiva ng-App en AngularJS?

Responder: Se utiliza para definir la aplicación AngularJs. Designa el elemento raíz de la aplicación y se mantiene cerca de la etiqueta `<html>`.

Podemos definir cualquier número de directivas ng-app dentro del documento HTML, pero solo una aplicación AngularJS puede arrancarse automáticamente (autoarrancarse) y las otras aplicaciones deben arrancarse manualmente.

Ejemplo:

```
|_+_|
```

Q # 28) ¿Qué es ng-switch en AngularJS?

Responder: Se utiliza para intercambiar condicionalmente la estructura de DOM en una plantilla que se basa en una expresión basada en el alcance.

Esta directiva le permite mostrar u ocultar el elemento HTML según la expresión.

P # 29) ¿Cuál es el uso de un evento de doble clic en AngularJs?

Responder: Le permite especificar el comportamiento personalizado en un evento de doble clic del mouse en una página web. Podemos usarlo (ng-dblclick) como un atributo del elemento HTML como,

```
|_+_|
```

P # 30) ¿Qué son las directivas ng-include y ng-click en AngularJs?

Responder:

ng-include ayuda a incluir diferentes archivos en la página principal. La directiva ng-include incluye HTML de un archivo externo.

El contenido incluido se incluirá como nodos secundarios del elemento especificado. El valor del atributo ng-include también puede ser una expresión, que devuelve un nombre de archivo.

De forma predeterminada, el archivo incluido debe estar ubicado en el mismo dominio que el documento.

ng-click se puede utilizar en escenarios como cuando hace clic en un botón o cuando desea realizar cualquier operación. Le dice a AngularJS qué hacer cuando se hace clic en un elemento HTML.

Ejemplo:

[Cómo Encontrar La Clave De Seguridad Para El Enrutador](#)

|_+_|

El código anterior aumentará la variable de recuento en uno cada vez que se haga clic en el botón.

P # 31) ¿Qué es una transferencia de estado representacional (REST) en AngularJs?

Responder: REST es un estilo de API que opera sobre la solicitud HTTP.

La URL solicitada identifica los datos sobre los que se va a operar y el método HTTP identifica la operación que se va a realizar. REST es un estilo de API en lugar de una especificación formal, y hay mucho debate y desacuerdo sobre qué es y qué no es RESTful, que es un término que se usa para indicar una API que sigue el estilo REST.

AngularJS es flexible sobre cómo se consumen los servicios web RESTful.

P # 32) ¿Qué son las API globales de AngularJs?

Responder: Es una combinación de función JavaScript global que se utiliza para realizar tareas como comparar objetos, iterar objetos y convertir datos.

Hay algunas funciones API comunes como:

- **angular. minúscula:** Convierte una cadena en una cadena en minúsculas.
- **angular. mayúscula:** Convierte una cadena en una cadena en mayúsculas.
- **angular. isString:** Devolverá verdadero si la referencia actual es una cadena.
- **angular. isNumber:** Devolverá verdadero si la referencia actual es un número.

P # 33) ¿Qué es un método de proveedor en AngularJs?

Responder: Un proveedor es un objeto que crea un objeto de servicio al permitir tomar más control.

El método \$ get () se usa en el proveedor que devuelve el objeto de servicio. El nombre del servicio y la función de fábrica son los argumentos que se pasan al método del proveedor. AngularJS usa \$ provide para registrar nuevos proveedores.

Sintaxis:

|_+_|

P # 34) ¿Qué es el manejo de eventos?

Responder: El manejo de eventos en AngularJs es muy útil cuando desea crear aplicaciones avanzadas de AngularJs.

Necesitamos manejar eventos DOM como clics del mouse, movimientos, pulsaciones de teclado, eventos de cambio, etc. AngularJs tiene algunas directivas de escucha como ng-click, ng-dbl-click, ng-mousedown, ng-keydown, ng-keyup, etc.

P # 35) ¿Qué es DOM de AngularJs?

Responder: AngularJs tienen algunas directivas que se utilizan para encapsular los datos de la aplicación AngularJs en un atributo deshabilitado de los elementos HTML.

Ejemplo: La directiva ng-disabled encapsula los datos de la aplicación a los atributos deshabilitados del elemento DOM de HTML.

|_+_|

P # 36) ¿Cuáles son los atributos que se pueden usar durante la creación de nuevas directivas AngularJs?

Responder: Hay varios atributos que se pueden utilizar durante la creación de una nueva directiva.

Incluyen:

1. **Plantilla:** Describe una plantilla en línea como una cadena.
2. **URL de la plantilla:** Este atributo especifica el compilador HTML de AngularJs para reemplazar la directiva personalizada dentro de una plantilla con el contenido HTML ubicado dentro de un archivo separado.
3. **Reemplazar:** Reemplaza el elemento actual si la condición es verdadera; si es falso, agregue esta directiva al elemento actual.
4. **Transclude:** Le permite mover los elementos secundarios originales de una directiva a una ubicación dentro de la nueva plantilla.
5. **Alcance:** Crea un nuevo ámbito para esta directiva en lugar de heredar el ámbito principal.
6. **Controlador:** Crea un controlador que publica una API para comunicarse a través de las directivas.
7. **Exigir:** Requiere la presencia de otra directiva para que la directiva actual funcione de manera eficiente.
8. **Enlace:** Modifica las instancias de elementos DOM resultantes, agrega detectores de eventos y configura el enlace de datos.
9. **Compilar:** Modifica la plantilla DOM para características en copias de una directiva, como cuando se usa en otras directivas. Su función de compilación también puede devolver funciones de enlace para modificar las instancias de elementos resultantes.

P # 37) ¿Son posibles los controladores anidados o no en AngularJs?

Responder: Sí, es posible, ya que los controladores anidados están bien definidos de manera clasificada al usar una vista.

P 38) ¿AngularJS se adapta bien a todos los navegadores?

Responder: Sí, es compatible con todos los navegadores como Safari, Chrome, Mozilla, Opera, IE, etc., así como con los navegadores móviles.

Q 39) Defina servicios en AngularJS.

Responder: Los servicios de AngularJS son los objetos o funciones singleton que se utilizan para llevar a cabo tareas definidas. Abarca algunas ideas corporativas y estos propósitos pueden denominarse controladores, directivas, filtros, etc.

P 40) Explique las ventajas de AngularJS.

Respuesta: las ventajas incluyen:

- Es compatible con la forma MVC.
- Organice dos formas de enlace de datos usando AngularJS.
- Admite la comunicación mutua cliente-servidor.
- Soporta simulaciones.

Q # 41) Diferencia entre servicios y fábrica.

Responder: Las fábricas son funciones que devuelven el objeto, mientras que los servicios son funciones constructoras del objeto que utiliza una nueva palabra clave.

Sintaxis:

Fábrica - `module.factory ('factoryName', función);`

Servicio - `module.service ('serviceName', función);`

P # 42) Si tanto la fábrica como el servicio son equivalentes, ¿cuándo debería usarlos?

Responder: Se prefiere al proveedor de fábrica usando un objeto, mientras que se prefiere a un proveedor de servicios usar con clase.

Q # 43) Diferencia entre AngularJS y React.JS.

Responder: AngularJS es un marco JS basado en lenguaje TypeScript lanzado en octubre de 2010 por Google. Es un marco completamente gratuito y de código abierto que se utiliza en proyectos de SPA (es decir, proyectos de aplicación de una sola página).

React.JS es una biblioteca de JavaScript desarrollada por Facebook en marzo de 2013 para crear UI. Los componentes de React se pueden usar en varias páginas pero no como un SPA (es decir, una aplicación de una sola página).

Q # 44) Diferencia entre la directiva ng-bind y ng-model.

Responder: La directiva ng-bind tiene enlaces de datos unidireccionales, los datos fluyen solo desde el objeto a la interfaz de usuario, no al revés (es decir, \$ scope >> view) y la directiva ng-model tiene enlaces de datos bidireccionales, los datos fluyen entre la interfaz de usuario al objeto y viceversa (es decir, \$ alcance >> ver y ver >> \$ alcance).

Q # 45) ¿Cuál es la diferencia entre AJAX y AngularJS?

Responder: AJAX significa JavaScript asíncronico que se utiliza para enviar y obtener respuestas del servidor sin cargar la página.

Considerando que, AngularJS es un marco de JavaScript basado en lenguaje mecanografiado que sigue el patrón MVC.

Q # 46) Defina ng-if, ng-show y ng-hide.

Responder: La directiva ng-if se usa como una cláusula que elimina el elemento HTML si la expresión se vuelve falsa.

Sintaxis

La directiva ng-show se usa para mostrar el elemento HTML si la expresión se vuelve verdadera. Y si la expresión se vuelve falsa, el elemento HTML se ocultará.

Sintaxis

La directiva ng-hide se usa para ocultar el elemento HTML si la expresión se vuelve falsa.

Sintaxis

Tanto ng-show como ng-hide utilizan el método de propiedad de visualización.

P # 47) ¿Cuál es la diferencia entre ngRoute y ui-router?

Responder: ngRoute es un módulo desarrollado por el equipo de angularJS que formaba parte del marco central de angularJS. Mientras que ui-router fue desarrollado por una comunidad de terceros para superar los problemas de ngRoute.

ngRoute es un enrutamiento basado en ubicación o URL, y ui-router es un enrutamiento basado en estado que permite vistas anidadas.

P # 48) ¿Cómo configurar, obtener y borrar cookies en AngularJs?

Responder: AngularJS tiene un módulo llamado ngCookies, por lo que antes de inyectar ngCookies debe incluirse angular-cookies.js en la aplicación.

- **Configurar cookies** - El método de colocación se utiliza para configurar cookies en un formato de valor clave.

|_+_|

- **Obtener cookies** - El método Get se utiliza para obtener cookies.

|_+_|

- **Eliminar cookies** - El método de eliminación se utiliza para eliminar las cookie