TutorialJS

2018

Indice

* [AngularJS](#AngularJS)…………………………………………………………………………………………………2
* [Direcctivas en AngularJS](#Directivas)………………………………………………………………………………3
* [IonicJS](#IonicJS)………………………………………………………………………………………………………….11
* [Cordova](#codova)………………………………………………………………………………………………………16
* [Android](#android)………………………………………………………………………………………………………17
* [IOS](#ios)………………………………………………………………………………………………………………18
* [Firebase](#firbase)………………………………………………………………………………………………………19
* [Recomendaciones](#recomendaciones)………………………………………………………………………………………21
* [Infografias](#infografias)……………………………………………………………………………………………………22

[Angular JS](#AngularJS)

AngularJS está construido en torno a la creencia de que la programación declarativa es la que debe utilizarse para generar interfaces de usuario y enlazar componentes de software, mientras que la programación imperativa es excelente para expresar la lógica de negocio.1​ Este framework adapta y amplía el HTML tradicional para servir mejor contenido dinámico a través de un data binding bidireccional que permite la sincronización automática de modelos y vistas. Como resultado, AngularJS pone menos énfasis en la manipulación del DOM y mejora la testeabilidad y el rendimiento.

Objetivos de diseño:

Disociar la manipulación del DOM de la lógica de la aplicación. Esto mejora la capacidad de prueba del código.

Considerar a las pruebas de la aplicación como iguales en importancia a la escritura de la aplicación. La dificultad de las pruebas se ve reducida drásticamente por la forma en que el código está estructurado.

Disociar el lado del cliente de una aplicación del lado del servidor. Esto permite que el trabajo de desarrollo avance en paralelo, y permite la reutilización de ambos lados.

Guiar a los desarrolladores a través de todo el proceso del desarrollo de una aplicación: desde el diseño de la interfaz de usuario, a través de la escritura de la lógica del negocio, hasta las pruebas.

Angular sigue el patrón MVVM (Model View View-Model) de ingeniería de software y alienta la articulación flexible entre la presentación, datos y componentes lógicos. Con el uso de la inyección de dependencias, Angular

Directivas en AngularJS

Algunas directivas nativas que podemos encontrar en angularjs son:

* ngApp (ng-app)
* ngController (ng-controller)
* ngModel (ng-model)
* ngClick (ng-click)
* ngInit (ng-init)
* ngChange (ng-change)
* ngShow (ng-show) | ngHide (ng-hide)
* ngBind (ng-bind)

**ngApp (ng-app)**

Es la directiva encargada de auto arrancar una aplicación Angular, indica el elemento raíz y se debe colocar como atributo en la etiqueta que quieres que sea la raíz de la aplicación.

La directiva ngApp la declaramos de la siguiente manera:

<html ng-app>

ngApp puede contener un módulo de AngularJS, y se declara de la siguiente manera:

<html ng-app="nombre\_del\_modulo">

**ngController (ng-controller)**

Es la directiva que permite indicarle a la vista donde trabajará nuestro controlador y enlazar un $scope, todo modelo que esté dentro del ámbito de la directiva podrá ser accedido desde el controlador asignado. ngController se declara de la siguiente manera:

<body>

<div ng-controller="nombre\_de\_controlador">

<h1>Hola AngularJS desde @frontendlabs</h1>

</div>

</body>

**ngModel (ng-model)**

Es la directiva que representa el modelo o dato, permite obtener la información ingresada por el usuario en algún elemento del formulario, sea un input, select o textarea. Si desea obtener el texto que un usuario ingresa en un input, solo bastará asociarle un modelo y éste podrá ser accedido tanto en el controlador como la vista mediante el nombre del modelo. Veamos como funciona esto:

<body>

<div ng-controller="miControlador">

<label>Ingrese su nombre</label>

<input type="text" ng-model="nombre">

<span>Hola {{nombre}}</span>

</div>

</body>

En el momento en que comencemos a escribir en el input, iremos viendo como se refleja lo escrito dentro del span, esto debido al principio “Two-Way data binding”, por otro lado en el controlador también tenemos acceso al modelo mediante $scope.nombre.

**ngClick (ng-click)**

Esta directiva trabaja directamente relacionado al evento click, se le puede asociar alguna funcionalidad en cuanto el usuario haga click sobre algún elemento.

Como ejemplo veamos el siguiente código, dentro del html creamos un botón al cual le asociamos la directiva ng-click, en cuanto se detecte el evento se realizará la funcionalidad que tengamos en el método enviar(), que probablemente enviemos el nombre hacia algún servidor o la guardemos en un array:

<body>

<div ng-controller="miControlador">

<label>Ingrese su nombre</label>

<input type="text" ng-model="nombre">

<button ng-click="enviar()">Enviar</button>

</div> </body>

**ngInit (ng-init)**

Esta directiva permite evaluar una expresión en el scope donde se está trabajando, veamos un ejemplo de como funciona:

<body>

<div ng-controller="miControlador">

<div>

<button ng-click="count = count + 1" ng-init="count = 0">Enviar</button>

<span>{{count}}</span>

</div>

</div>

</body>

**ngRepeat (ng-repeat)**

Esta directiva permite iterar una colección de datos, generar un template por cada elemento de la colección y pintarlo en la vista, cada template o plantilla recibe su propio ámbito ($scope).

Vemos en el siguiente ejemplo, como inicializamos una variable que contiene una colección de objetos y luego mediante la directiva ng-repeat iteramos la colección y pintamos una etiqueta “li” por cada elemento de la colección mostrando tanto el nombre como la edad de cada alumno:

<body>

<div ng-controller="miControlador">

<div ng-init="alumnos = [

{nombre:'Paul', edad:12},

{nombre:'Carlos', edad:13},

{nombre:'Jan', edad:14},

{nombre:'Ana', edad:15},

{nombre:'Victor', edad:16}

]">

<ul>

<li ng-repeat="alumno in alumnos">{{alumno.nombre}}: {{alumno.edad}} años</li>

</ul>

</div>

</div> </body>

**ngChange (ng-change)**

Esta directiva detecta cualquier cambio que se produzca dentro de una etiqueta de entrada, sean inputs, checkbox, etc., la forma de usarla es la siguiente.

En el lado de la vista tenemos dos opciones a marcar, si esta a favor o en contra, lo cual incrementará o restará las votaciones según sea el caso, veamos:

<body>

<div ng-controller="miControlador">

<input type="checkbox" ng-model="total" ng-change="aFavor()"> A favor

<input type="checkbox" ng-model="total" ng-change="enContra()"> En contra

<h3>Total Votos: {{total}}</h3>

</div>

</body>

En el lado del controlador sería de la siguiente manera:

app.controller('miControlador', function($scope){

$scope.total = 0;

$scope.aFavor = function (){

$scope.total++;

};

$scope.enContra = function (){

$scope.total--;

};

});

**ngShow (ng-show) | ngHide (ng-hide)**

Estas directivas permiten mostrar y ocultar alguna parte de la vista según la condición que le asignemos. Como seguramente todos ya deben saber ngShow permite mostrar y ngHide permite ocultar, veamos un ejemplo sencillo de como trabajan estas dos directivas.

Tenemos dos opciones “mostrar” y “ocultar”, y el mensaje a mostrar, como vemos en el código tenemos los dos checkbox y los dos mensajes, uno de los mensajes con la directiva ng-show y la otra con ng-hide, lo que nos indica que en cuanto “dato1″ sea true(este marcada) se mostrará en caso contrario se ocultará, de forma inversa en el segundo mensaje, donde en caso “dato2″ se true se ocultará y en caso contrario se mostrará.

En primera instancia al estar ambos sin check, estarán en estado false, por lo cual el único mensaje que se mostrará es el segundo.

<body>

<div ng-controller="miControlador">

<input type="checkbox" ng-model="dato1"> Mostrar

<input type="checkbox" ng-model="dato2"> Ocultar

<h3 ng-show="dato1">@frontendlabs 1</h3>

<h3 ng-hide="dato2">@frontendlabs 2</h3>

</div> </body>

**ngBind (ng-bind)**

Esta directiva cumple la misma funcionalidad que las llaves , sin embargo, ng-bind tiene una mejor performance en cuanto a tiempo. En el siguiente ejemplo vemos la forma de uso, y tanto como ng-bind muestran el texto que se ingresa en la caja de texto.

<body>

<div ng-controller="miControlador">

<input type="text" ng-model="nombre">

<span>{{nombre}}</span>

<span ng-bind="nombre"></span>

</div>

</body>

Existen muchas más directivas, las cuales podrán encontrar en la Documentación oficial de AngularJS

Directivas Propias

Pese a que en angular se han implementado todas las posibles directivas, siempre podremos optar a crear nuestra propia directiva y es por esto que en AngularJS nos facilitan esta tarea.

Este es un ejemplo simple para poder implementar nuestra directiva.

var app = angular.module('MiModulo',[]);

app.controller('MiControlador', function($scope){

$scope.cliente = {

nombre: 'Jhon',

direccion: 'Av. Jose pardo 481, Miraflores, Lima, Perú'

};

});

//Aquí creamos la directiva

app.directive('miCliente', function() {

return {

template: 'Nombre: {{cliente.nombre}} Dirección: {{cliente.direccion}}'

};

});

IonicJS

Más de uno, y de dos, conoceréis o habréis escuchado hablar de Ionic,este framework que tanta importancia está tomando últimamente. Hagamos un poco de memoria, anteriormente hemos hablado sobre Angular JS y detallamos algunos de los frameworks que están en la red que trabajan bajo ese MVC y MVVM. Como sabéis, para un programador es muy importante tener bien organizado el código de nuestro proyecto y bien comentado, no sea que nos toque modificar algo meses después y… cualquiera se acuerda de lo que hacía esa función en concreto ;)

logo-ionic-framework-phonegap-spain

Lo primero que me gustaría explicar, y de forma escueta, es qué es un MVC y de qué se compone:

**El MVC**

El MVC (Model-View-Controller o Modelo-Vista-Controlador), es un patrón de diseño que separa los datos, la lógica y las interfaces de usuario. Como su nombre indica, está separado en tres componentes: Modelo, Vista y Controlador. Está basado en la ideología de separación de conceptos y cumple perfectamente con los objetivos de los patrones de diseño.

1.- Modelo

Es la capa encargada de los datos, es decir, la que se encarga de hacer peticiones a las bases de datos para enviar o recibir información. Estas bases

de datos pueden estar alojadas de forma local en nuestra app o de forma remota en un servidor externo.

2.- Vista

Se trata del código que nos permitirá presentar los datos que el modelo nos proporciona, como ejemplo podríamos decir que en una aplicación es el código HTML que nos permite mostrar la salida de los datos procesados.

3.- Controlador

Es la capa que sirve de enlace entre la vista y el modelo. Envía comandos al modelo para actualizar su estado, y a la vista correspondiente para cambiar su presentación.

En el caso MVVM (Modelo Vista VistaModelo) la iteracción entre la vista y el controlador será en los dos sentidos, el controlador muestra los datos en la vista y si en la vista hay un cambio de datos, se actualiza el modelo automáticamente.

**El framework Ionic**

Ionic es una herramienta, gratuita y open source, para el desarrollo de aplicaciones híbridas basadas en HTML5, CSS y JS. Está construido con Sass y optimizado con AngularJS.

Principales características

ionic-1

1.- Alto rendimiento

La velocidad es importante. Tan importante que sólo se nota cuando no está en tu app. Ionic está construido para ser rápido gracias a la mínima

manipulación del DOM, con cero jQuery y con aceleraciones de transiciones por hardware.

ionic-2

1.- AngularJS & Ionic

Ionic utiliza AngularJS con el fin de crear un marco más adecuado para desarrollar aplicaciones ricas y robustas. Ionic no sólo se ve bien, sino que su arquitectura central es robusta y seria para el desarrollo de aplicaciones. Trabaja perfectamente con AngularJS.

ionic-3

3.- Centro nativo

Ionic se inspira en las SDK de desarrollo móviles nativos más populares, por lo que es fácil de entender para cualquier persona que ha construido una aplicación nativa para iOS o Android. Lo interesante, como sabéis, es que desarrollas una vez, y compilas para varios.

ionic-4

4.- Bonito diseño

Limpio, sencillo y funcional. Ionic ha sido diseñado para poder trabajar con todos los dispositivos móviles actuales. Con muchos componentes usados en móviles, tipografía, elementos interactivos, etc.

ionic-5

5.- Un potente CLI

Con un sólo comando podrás crear, construir, probar y compilar tus aplicaciones en cualquier plataforma.

Cómo empezar

Con 4 líneas no es suficiente para explicar a fondo cómo empezar con Ionic y, aunque vamos a preparar un tutorial en español, quiero soltar algunas líneas que nos pueden ayudar a crear nuestra primera app con este framework

Los pasos para comenzar son:

1.- Instalar Ionic.

Recuerda que es necesario tener NodeJS instalado en tu ordenador. Ionic se instala igual que PhoneGap o Cordova. Aunque trabaje bajo ellos, Ionic tiene su propio instalador:

1. $ npm install -g cordova ionic

2.- Crear nuestro primer proyecto

Podemos empezar un proyecto de tres formas:

1. Con un proyecto vacío.

1. $ ionic start myApp blank

b) Con un proyecto con la estructura de menú inferior.

1. $ ionic start myApp tabs

c) Con un proyecto con menú lateral.

1. $ ionic start myApp sidemenu

ionic-start

3.- Compilar

En las siguientes líneas, os serán familiares ya que son muy parecidas a la compilación con PhoneGap o Cordova

Code: arbitrary

$ cd miApp

$ ionic platform add ios

$ ionic build ios

$ ionic emulate ios

**Ionic Creator**

Una nueva herramienta que ha creado el equipo de Ionic es “Ionic Creator“. Esta herramienta es muy parecida a codiqa, la cual, para los que trabajen bajo JQuery o JQMobile les será muy familiar. Bien, pues Ionic Creator hace la misma función: nos permite crear la parte gráfica de nuestra app online de forma visual, sin tener que tocar código. Esto es muy interesante ya que nos permite ver de forma directa e instantánea los cambios que estamos aplicando a cada una de nuestras vistas y la repercusión gráfica que tiene.

**Ionic y la compilación remota con PhoneGap Build**

Todos nuestras apps desarrolladas con Ionic pueden compilarse de forma remota utilizando https://build.phonegap.com/apps, que, como sabéis, es un servicio de Adobe que nos permite compilar nuestra app desde la nube. Y de esta forma no necesitamos un Mac para compilar para iOS o un Windows para compilar para WP.

Cordova

Cordova (versión de código abierto de PhoneGap) es un popular entorno de desarrollo de aplicaciones móviles, originalmente creado por Nitobi. Adobe compró Nitobi en 2011, le cambió el nombre a PhoneGap, y más tarde liberó una versión de código abierto del software llamado Apache Cordova.3​ Apache Cordova permite, a los programadores de software, construir aplicaciones para dispositivos móviles utilizando CSS3, HTML5, y JavaScript en vez de utilizar APIs específicas de cada plataforma como Android, iOS, o Windows Phone.4​ Permite encapsular CSS, HTML, y código de Javascript dependiendo de la plataforma del dispositivo. Extiende las características de HTML y JavaScript para trabajar con el dispositivo. Las aplicaciones resultantes son híbridas, lo que significa que no son ni una aplicación móvil nativa (porque toda la representación gráfica se realiza vía vistas de Web en vez del framework nativo) ni puramente basadas en web (porque no son solo aplicaciones web, sino que están empaquetadas como aplicaciones para su distribución y tienen acceso a las APIs nativas del dispositivo). La mezcla de código nativo e híbrido ha sido posible desde la versión 1.9.

Anteriormente se conocía como "PhoneGap", después "Apache Callback".5​6​ Como software libre, Apache Cordova permite wrappers, como Appery.io o Intel XDK.

PhoneGap es la versión paquetizada de Adobe de Apache Cordova. Como PhoneGap, muchas otras herramientas y frameworks también son construidos utilizando Apache Cordova, incluyendo Ionic, Monaca, TACO, el Intel XDK y Telerik.7​8​9​ Estas herramientas usan Apache Cordova, y no PhoneGap como sus herramientas de base.

Entre los colaboradores al proyecto Apache Cordova encontramos a Adobe, BlackBerry, Google, IBM, Intel, Microsoft, Mozilla, y otros.

Android

Android es un sistema operativo basado en el núcleo Linux. Fue diseñado principalmente para dispositivos móviles con pantalla táctil, como teléfonos inteligentes, tabletas y también para relojes inteligentes, televisores y automóviles. Inicialmente fue desarrollado por Android Inc., empresa que Google respaldó económicamente y más tarde, en 2005, compró.3​ Android fue presentado en 2007 junto la fundación del Open Handset Alliance (un consorcio de compañías de hardware, software y telecomunicaciones) para avanzar en los estándares abiertos de los dispositivos móviles.4​ El primer móvil con el sistema operativo Android fue el HTC Dream y se vendió en octubre de 2008.5​ Android es el sistema operativo móvil más utilizado del mundo, con una cuota de mercado superior al 80% al año 2017, muy por encima de IOS.6​7​8​9​

El éxito del sistema operativo se ha convertido en objeto de litigios sobre patentes en el marco de las llamadas guerras de patentes entre las empresas de teléfonos inteligentes.10​11​ Según los documentos secretos filtrados en 2013 y 2014, el sistema operativo es uno de los objetivos de las agencias de inteligencia internacionales.12​

La versión básica de Android es conocida como Android Open Source Project (AOSP).13​

El 25 de junio de 2014 en la Conferencia de Desarrolladores Google I/O, Google mostró una evolución de la marca Android, con el fin de unificar tanto el hardware como el software y ampliar mercados.

El 17 de mayo de 2017, se presentó Android Go. Una versión más ligera del sistema operativo para ayudar a que la mitad del mundo sin smartphone consiga uno en menos de cinco años. Incluye versiones especiales de sus aplicaciones donde el consumo de datos se reduce al máximo.

IOS

iOS es un sistema operativo móvil de la multinacional Apple Inc. Originalmente desarrollado para el iPhone (iPhone OS), después se ha usado en dispositivos como el iPod touch y el iPad. No permite la instalación de iOS en hardware de terceros.

Actualmente es el segundo sistema operativo móvil más utilizado del mundo, detrás de Android, con una cuota de mercado de entre 10-15% al año 2017. La última versión del sistema operativo es el iOS 11, aparecida en el mes de septiembre del 2017, disponible en dispositivos con procesadores 64-bits (desde el IPhone 5S en adelante.

Los elementos de control consisten de deslizadores, interruptores y botones. La respuesta a las órdenes del usuario es inmediata y provee una interfaz fluida. La interacción con el sistema operativo incluye gestos como deslices, toques, pellizcos, los cuales tienen definiciones diferentes dependiendo del contexto de la interfaz. Se utilizan acelerómetros internos para hacer que algunas aplicaciones respondan a sacudir el dispositivo (por ejemplo, para el comando deshacer) o rotarlo en tres dimensiones (un resultado común es cambiar de modo vertical al apaisado u horizontal).

En el marco de las filtraciones acerca de los programas de vigilancia mundial de 2013-2014 de Edward Snowden, Der Spiegel publicó que la NSA estadounidense tiene grupos de trabajo dedicados a descifrar los sistemas de seguridad de iOS; además tiene pequeños programas conocidos como scripts que permiten a la agencia vigilar a los usuarios de las distintas versiones del sistema iOS su geolocalización, notas de voz, fotos y otras aplicaciones como Google Earth, Facebook o Yahoo! Messenger.2​

iOS se deriva de macOS, que a su vez está basado en Darwin BSD, y por lo tanto es un sistema operativo Tipo Unix. iOS cuenta con cuatro capas de abstracción: la capa del núcleo del sistema operativo, la capa de "Servicios Principales", la capa de "Medios" y la capa de "Cocoa Touch".

Firebase

Firebase es una plataforma para el desarrollo de aplicaciones web y aplicaciones móviles desarrollada por James Tamplin y Andrew Lee en 2011 y adquirida por Google en 2014.

Servicios

Analytics

Firebase Analytics

Firebase Analytics es una aplicación gratuita que proporciona una visión profunda sobre el uso de la aplicación por parte de los usuarios.9​

Desarrollo

Firebase Cloud Messaging

Antiguamente conocido como Google Cloud Messaging (GCM), Firebase Cloud Messaging (FCM) es una plataforma para mensajes y notificaciones para Android, iOS, y aplicaciones web que actualmente puede ser usada de forma gratuita.10​

Firebase Auth

Firebase Auth es un servicio que puede autenticar los usuarios utilizando únicamente código del lado del cliente. Incluye la autenticación mediante Facebook, GitHub, Twitter y Google. Además, incluye un sistema de administración del usuario por el cual los desarrolladores pueden habilitar la autenticación de usuarios con email y contraseña que se almacenarán en Firebase.11​

Realtime Database

Firebase proporciona una base de datos en tiempo real y back-end. El servicio proporciona a los desarrolladores de aplicaciones una API que permite que la información de las aplicaciones sea sincronizada y almacenada en la nube de Firebase.12​13​ La compañía habilita integración con aplicaciones Android, iOS, Javascript, Java, Objective-C, Swift y Node.js. La base de datos es también accesible a través de una REST API e integración para varios sistemas de Javascript como AngularJS, React, Ember.js y Backbone.js.14​. La REST API utiliza el protocolo SSE (del inglés Server-Sent Events), el cual es una API para crear conexiones de HTTP para recibir notificaciones push de un servidor

Firebase Storage

Firebase Storage proporciona cargas y descargas seguras de archivos para aplicaciones Firebase, sin importar la calidad de la red. El desarrollador lo puede utilizar para almacenar imágenes, audio, vídeo, o cualquier otro contenido generado por el usuario. Firebase Storage se basa en el almacenamiento de Google Cloud Storage.15​

Firebase Firestore

Firebase Firestore es un servicio derivado de Google Cloud Platform, adaptado a la plataforma de Firebase. Al igual que Realtime Database, es una base de datos NoSQL, aunque presenta diversas diferencias. Se organiza en forma de documentos agrupados en colecciones, y en ellos se pueden incluir tanto campos de diversos tipos (cadenas de texto, números, puntos geográficos, referencias a la propia base de datos, arrays, booleanos, marcas de tiempo, e incluso objetos propios) como otras subcolecciones.

Recomendaciones

Es recomendable que en el momento de instalar un programa se sigan los pasos necesarios para evitar la pérdida del material o el mal funcionamiento del programa.

Infografías

[www.es.wikipedia.org](http://www.es.wikipedia.org)

[www.phonegapspain.com](http://www.phonegapspain.com)