Brianda suceli Toj Patzán

Empresa xik  18 Av. 4-24 zona 16, Guatemala City

capacitación de personal

Contenido

[ANGULARJS 2](#_Toc511292163)

[DIRECTIVAS EN ANGULARJS 3](#_Toc511292164)

[Directivas Propias 10](#_Toc511292165)

[LONICJS 11](#_Toc511292166)

[El MVC 12](#_Toc511292167)

[Principales características: 13](#_Toc511292168)

[Cordova 16](#_Toc511292169)

[ANDROID 20](#_Toc511292170)

[CARACTERISTICAS 21](#_Toc511292171)

[IOS 24](#_Toc511292172)

[CARACTERISTICAS 25](#_Toc511292173)

[FIREBASE 29](#_Toc511292174)

[Servicios 30](#_Toc511292175)

# ANGULARJS

AngularJS (comúnmente llamado Angular.js o AngularJS 1), es un framework de JavaScript de código abierto, mantenido por Google, que se utiliza para crear y mantener aplicaciones web de una sola página. Su objetivo es aumentar las aplicaciones basadas en navegador con capacidad de Modelo Vista Controlador (MVC), en un esfuerzo para hacer que el desarrollo y las pruebas sean más fáciles.

La biblioteca lee el HTML que contiene atributos de las etiquetas personalizadas adicionales, entonces obedece a las directivas de los atributos personalizados, y une las piezas de entrada o salida de la página a un modelo representado por las variables estándar de JavaScript. Los valores de las variables de JavaScript se pueden configurar manualmente, o recuperados de los recursos JSON estáticos o dinámicos.

AngularJS se puede combinar con el entorno en tiempo de ejecución Node.js, el framework para servidor Express.js y la base de datos MongoDB para formar el conjunto MEAN.

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.

## DIRECTIVAS EN ANGULARJS

Las directivas son marcas en los elementos del árbol DOM, en los nodos del HTML, que indican al compilador de Angular que debe asignar cierto comportamiento a dichos elementos o transformarlos según corresponda. Son útiles para definir nuevos pedazos de código html —como botones o, por ejemplo— un footer completo, como también para crear cosas más complejas que incluyan funciones o variables dinámicas definidas directamente en el DOM.

En cuanto a la tipología nos podemos encontrar con 2 tipos de directivas: Las existentes (nativas de AngularJS); y las propias, que son aquellas directivas que creamos nosotros para un uso más personalizado.

***DIRECTIVAS NATIVAS:***

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 cómo 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 cómo 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>

**ng-change (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 cómo 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: 'John',

dirección: 'Av. José pardo 481, Miraflores, Lima, Perú'

};

});

//Aquí creamos la directiva

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

return {

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

};

});

En la parte de la vista usaríamos nuestra directiva así:

<body>

<div ng-controller="MiControlador">

<div mi-cliente></div>

</div>

</body>

Otra forma de usar nuestra directiva sería usarlo como etiqueta.

# LONICJS

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 framework 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.

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 interacción 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**

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:

a) 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

1. $ cd miApp

2. $ ionic platform add ios

3. $ ionic build ios

4. $ 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-creator-phonegap-spain

Ionic y los plugins de Cordova

Tal y como se los he comentado antes, Ionic trabaja bajo Cordova, por lo que podremos utilizar todos los plugins que estén desarrollados para el framework e incluso desarrollar los nuestros propios

# Cordova

Apache Cordova es un marco de desarrollo móvil de código abierto. Permite utilizar las tecnologías estándar web como HTML5, CSS3 y JavaScript para desarrollo multiplataforma, evitando el lenguaje de desarrollo nativo cada plataformas móviles. Aplicaciones ejecutan dentro de envolturas para cada plataforma y dependen de enlaces estándares API para acceder a de cada dispositivo sensores, datos y estado de la red.

Apache Cordova se graduó en octubre de 2012 como un proyecto de nivel superior dentro de la Apache Software Foundation (ASF). A través del ASF, futuro desarrollo Cordova asegurará administración abierta del proyecto. Siempre permanecerá libre y de código abierto bajo la licencia Apache, versión 2.0. Visite cordova.apache.org para obtener más información.

Usar Apache Cordova si eres:

Establecen un móvil desarrollador y desea extender una aplicación a través de más de una plataforma, sin tener que reimplementarlo con herramienta y lenguaje de cada plataforma.

Un desarrollador web y desea implementar una aplicación web que se envasa para su distribución en varias app store portales.

Un móvil desarrollador interesado en que se mezclan los componentes de la aplicación nativa con un WebView (ventana del navegador especial) que puede tener acceso a las API de nivel de dispositivo, o si quiere desarrollar una interfaz plugin entre componentes WebView y nativos.

Componentes básicos

Apache Cordova aplicaciones se basan en un común config.xml archivo que proporciona información acerca de la aplicación y especifica los parámetros que afectan a cómo funciona, como si responde a la orientación cambia de puesto. Este archivo se adhiere a la especificación de Empaquetado de la aplicación Web, widget, o de la W3C.

La misma aplicación se implementa como una página web, un archivo local llamado index.html, que hace referencia a cualquier CSS, JavaScript, imágenes, archivos multimedia u otros recursos son necesarios para que se ejecute de forma predeterminada. La aplicación se ejecuta como un WebView dentro de la envoltura de la aplicación nativa, que distribuye a tiendas de aplicaciones.

El WebView Cordova-habilitado puede proporcionar la aplicación con su interfaz de usuario completa. En algunas plataformas, también puede ser un componente dentro de una aplicación híbrida más grande, que mezcla la vista Web con componentes de la aplicación nativa. (Véase WebViews incrustación para más detalles).

Una interfaz plugin está disponible para Cordova y componentes nativos para comunicarse con los demás. Esto te permite invocar un código de JavaScript. Idealmente, las API de JavaScript para ese código nativo son consistentes a través de múltiples plataformas de dispositivos. A partir de la versión 3.0, las extensiones proporcionan enlaces a APIs estándar. Plugins de terceros proporcionan enlaces adicionales a funciones no necesariamente disponibles en todas las plataformas. Puede encontrar estos plugins de terceros en el registro de plugin y utilizarlos en su aplicación. También puedes desarrollar tus propios plugins, como se describe en la guía de desarrollo de Plugin. Plugins puede ser necesario, por ejemplo, para comunicarse entre Córdoba y componentes personalizados de nativos.

**Nota:** desde la versión 3.0, cuando se crea un proyecto de Córdoba no tiene presente algún plugin. Este es el nuevo comportamiento predeterminado. Algún plugin que desee, incluso los plugins del núcleo, debe agregarse explícitamente.

Córdova no proporciona ninguna widgets UI o MV \* Marcos. Cordova proporciona solamente el tiempo de ejecución en el cual los pueden ejecutar. Si usted desea utilizar widgets UI o un marco MV \*, usted necesitará seleccionar aquellos e incluirlos en su solicitud usted mismo como material de terceros.

**Vías de desarrollo**

A partir de la versión 3.0, puede utilizar dos flujos de trabajo básicos para crear una aplicación móvil. Mientras que a menudo puede utilizar cualquier flujo de trabajo para realizar la misma tarea, cada uno de ellos ofrece ventajas:

Flujo de trabajo multiplataforma (CLI): Use este flujo de trabajo si quieres tu aplicación para ejecutar en los sistemas operativos móviles como sea posible, con poco necesidad específica de la plataforma desarrollo. Este flujo de trabajo se centra en la cordova utilidad, también conocido como el CLI, que fue introducido con 3.0 Cordova Cordova. El CLI es una herramienta de alto nivel que le permite construir proyectos para muchas plataformas a la vez, muy lejos de la funcionalidad de scripts de shell de bajo nivel de abstracción. La CLI copia un conjunto común de web activos en subdirectorios para cada plataforma móvil, hace que cualquier cambio de configuración necesarias para cada uno, construir secuencias de comandos para generar los binarios de la aplicación ejecuta. La CLI también proporciona una interfaz común para aplicar plugins para su aplicación. Para más detalles sobre el CLI, vea la interfaz de línea de comandos. A menos que usted tiene una necesidad para el flujo de trabajo centrado en plataforma, se recomienda el flujo de trabajo multiplataforma.

Flujo de trabajo centrado en plataforma: Utilice este flujo de trabajo si desea concentrarse en construir una aplicación para una sola plataforma y necesitan poder modificarlo en un nivel inferior. Tienes que utilizar este enfoque, por ejemplo, si quieres tu aplicación para mezclar los componentes nativos personalizados con componentes Cordova basados en web, como se explica en WebViews incrustación. Como regla general, utilice este flujo de trabajo si necesitas modificar el proyecto dentro del SDK. Este flujo de trabajo se basa en un conjunto de scripts de shell de nivel inferior que se adaptan para cada plataforma soportada y una utilidad de Plugman separada que le permite aplicar plugins. Mientras este flujo de trabajo puede utilizar para crear aplicaciones multiplataforma, es generalmente más difícil porque la falta de una herramienta de alto nivel significa construir diferentes ciclos y modificaciones de plugin para cada plataforma. Aun así, este flujo de trabajo permite un mayor acceso a opciones de desarrollo proporcionadas por cada SDK y es esencial para aplicaciones complejas híbrido. Vea a las diversas guías de plataforma para obtener más información sobre utilidades de shell disponible de cada plataforma.

Cuando primero comenzando, puede ser más fácil utilizar el flujo de trabajo multiplataforma para crear una aplicación, como se describe en la interfaz de línea de comandos. Entonces tienes la opción de cambiar a un flujo de trabajo centrado en plataforma si necesitas el SDK proporciona un mayor control. Utilidades de shell de nivel inferior están disponibles en cordova.apache.org en una distribución independiente de la CLI. Para los proyectos generados inicialmente por la CLI, estas herramientas de shell también están disponibles en el proyecto de varios platforms/\*/cordova directorios.

**Nota:** una vez que se cambia de los flujos de trabajo basados en CLI a uno centrado en la específica de la plataforma SDK y herramientas de shell, no puedes volver. La CLI mantiene un conjunto común de código fuente multiplataforma, que en cada uno a construir aplicaciones para escribir código fuente específica de la plataforma. Para conservar las modificaciones en los activos específicos a una plataforma, puede necesita cambiar las herramientas de plataforma centrada en la cáscara, que ignoran el código multiplataforma, y en cambio se basa en el código fuente específica de la plataforma.

**Instalación de Cordova**

La instalación de Córdoba será diferente dependiendo del flujo de trabajo anterior que usted elige:

**Flujo de trabajo multiplataforma:** ver la interfaz de línea de comandos.

**Flujo de trabajo centrado en plataforma:** ver las guías de la plataforma.

Después de instalar Cordova, se recomienda que consulte a las guías de plataforma para las plataformas móviles que va a desarrollar para. También se recomienda que también revise la guía de privacidad, seguridad y próximos pasos. Para configurar Cordova, consulte el archivo config.xml. Para acceder a la función nativa en un dispositivo de JavaScript, se refieren a las APIs de Plugin. Y se refieren a las otras guías incluidas según sea necesario.

# 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.

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

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.

## CARACTERISTICAS

**Diseño de dispositivo**: La plataforma es adaptable a pantallas de mayor resolución, VGA, biblioteca de gráficos 2D, biblioteca de gráficos 3D basada en las especificaciones de la OpenGL ES 2.0 y diseño de teléfonos tradicionales.

**Almacenamiento:** SQLite, una base de datos liviana, que es usada para propósitos de almacenamiento de datos.

**Conectividad:** Android soporta las siguientes tecnologías de conectividad: GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, Bluetooth, Wi-Fi, LTE, HSDPA, HSPA+, NFC y WiMAX, GPRS, UMTS y HSDPA+.

**Mensajería:** SMS y MMS son formas de mensajería, incluyendo mensajería de texto, además del servicio de Firebase Cloud Messaging (FCM) siendo la nueva versión de Google Cloud Messaging (GCM) bajo la marca Firebase con los nuevos SDK para realizar el desarrollo de mensajería en la nube mucho más sencillo.

**Navegador web:** El navegador web incluido en Android está basado en el motor de renderizado de código abierto WebKit, emparejado con el motor JavaScript V8 de Google Chrome. El navegador por defecto de Ice Cream Sándwich obtiene una puntuación de 100/100 en el test Acid3.

**Soporte de Java:** Aunque la mayoría de las aplicaciones están escritas en Java, no hay una máquina virtual Java en la plataforma. El bytecode Java no es ejecutado, sino que primero se compila en un ejecutable Dalvik y se ejecuta en la Máquina Virtual Dalvik, Dalvik es una máquina virtual especializada, diseñada específicamente para Android y optimizada para dispositivos móviles que funcionan con batería y que tienen memoria y procesador limitados. A partir de la versión 5.0, se utiliza el Android Runtime (ART). El soporte para J2ME puede ser agregado mediante aplicaciones de terceros como el J2ME MIDP Runner.45​

**Soporte multimedia:** Android soporta los siguientes formatos multimedia: WebM, H.263, H.264 (en 3GP o MP4), MPEG-4 SP, AMR, AMR-WB (en un contenedor 3GP), AAC, HE-AAC (en contenedores MP4 o 3GP), MP3, MIDI, Ogg Vorbis, WAV, JPEG, PNG, GIF y BMP.44​

**Soporte para streaming:** Streaming RTP/RTSP (3GPP PSS, ISMA), descarga progresiva de HTML (HTML5 <video> tag). Adobe Flash Streaming (RTMP) es soportado mediante el Adobe Flash Player. Se planea el soporte de Microsoft Smooth Streaming con el port de Silverlight a Android. Adobe Flash HTTP Dynamic Streaming estará disponible mediante una actualización de Adobe Flash Player.

**Soporte para hardware adicional:** Android soporta cámaras de fotos, de vídeo, pantallas táctiles, GPS, acelerómetros, giroscopios, magnetómetros, sensores de proximidad y de presión, sensores de luz, gamepad, termómetro, aceleración por GPU 2D y 3D.

**Entorno de desarrollo**: Incluye un emulador de dispositivos, herramientas para depuración de memoria y análisis del rendimiento del software. Inicialmente el entorno de desarrollo integrado (IDE) utilizado era Eclipse con el plugin de Herramientas de Desarrollo de Android (ADT). Ahora se considera como entorno oficial Android Studio, descargable desde la página oficial de desarrolladores de Android.

**Google Play:** Google Play es un catálogo de aplicaciones gratuitas o de pago en el que pueden ser descargadas e instaladas en dispositivos Android sin la necesidad de un PC.

**Multi-táctil:** Android tiene soporte nativo para pantallas capacitivas con soporte multitáctil que inicialmente hicieron su aparición en dispositivos como el HTC Hero. La funcionalidad fue originalmente desactivada a nivel de kernel (posiblemente para evitar infringir patentes de otras compañías).46​ Más tarde, Google publicó una actualización para el Nexus One y el Motorola Droid que activa el soporte multitáctil de forma nativa.47​

**Bluetooth:** El soporte para A2DF y AVRCP fue agregado en la versión 1.5;48​ el envío de archivos (OPP) y la exploración del directorio telefónico fueron agregados en la versión 2.0;49​ y el marcado por voz junto con el envío de contactos entre teléfonos lo fueron en la versión 2.2. Los cambios incluyeron: 50​

**Videollamada**: Android soporta videollamada a través de Hangouts (antiguo Google Talk) desde su versión HoneyComb.

**Multitarea:** Multitarea real de aplicaciones está disponible, es decir, las aplicaciones que no estén ejecutándose en primer plano reciben ciclos de reloj.

**Características basadas en voz:** La búsqueda en Google a través de voz está disponible como "Entrada de Búsqueda" desde la versión inicial del sistema.51​

**Tethering:** Android soporta tethering, que permite al teléfono ser usado como un punto de acceso alámbrico o inalámbrico (todos los teléfonos desde la versión 2.2, no oficial en teléfonos con versión 1.6 o inferiores mediante aplicaciones disponibles en Google Play (por ejemplo PdaNet). Para permitir a un PC usar la conexión de datos del móvil Android se podría requerir la instalación de software adicional.

# 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".

## CARACTERISTICAS

**Pantalla principal**

La pantalla principal (llamada «SpringBoard») es donde se ubican los iconos de las aplicaciones y el Dock en la parte inferior de la pantalla donde se pueden anclar aplicaciones de uso frecuente, aparece al desbloquear el dispositivo o presionar el botón de inicio. La pantalla tiene una barra de estado en la parte superior para mostrar datos, tales como la hora, el nivel de batería, y la intensidad de la señal. El resto de la pantalla está dedicado a la aplicación actual.

**Carpetas**

Con iOS 4 se introdujo un sistema simple de crispetas en el sistema. Se puede mover una aplicación sobre otra y se creará una carpeta, y así se pueden agregar más aplicaciones a esta mediante el mismo procedimiento. Pueden entrar hasta 12 y 20 aplicaciones en el iPhone y iPad respectivamente. El título de la carpeta es seleccionado automáticamente por el tipo de aplicaciones dentro de ella, pero puede ser editado por el usuario.

Con la salida de iOS 7, la cantidad máxima de aplicaciones por carpeta aumentó considerablemente, pues al abrir una carpeta se muestran 9 iconos (3x3), y al agregar más aplicaciones se van creando páginas a las que se pueden acceder deslizando sobre la pantalla

**Seguridad**

Antes de la salida de iOS 7 al mercado, existía un enorme índice de robos de los diversos modelos de iPhone, lo que provocó que el gobierno estadounidense solicitara a Apple diseñar un sistema de seguridad infalible que inutilizara los equipos en caso de robo. Fue creada entonces la activación por iCloud, la cual solicita los datos de acceso de la cuenta del usuario original, lo que permite bloquear e inutilizar el equipo al perderlo o ser víctima de robo del mismo.10​11​ De igual manera, es posible conocer la ubicación vía GPS del dispositivo y mostrar mensajes en la pantalla. Hasta la fecha no existe un método comprobado para saltarse la activación de iCloud lo que convierte a iOS 7+ en el SO móvil más seguro del mercado. Con la llegada de iOS 9.1, es imposible encontrar vulnerabilidades que afecten al dispositivo, llevándole a Apple, a ocupar el puesto más alto en la lista de los softwares más seguros. No obstante, iOS 9.3 fue un completo fracaso en cuanto a seguridad y vulnerabilidad con el Jailbreak. Sin embargo, el 25 de Agosto Apple liberó iOS 9.3.5 el cual reparaba esta vulnerabilidad.

Con el lanzamiento de iOS 10, usuarios reportaron una gran vulnerabilidad con iCloud, el cual se podía desactivar "Find My iPhone" (Buscar Mi iPhone) sin ingresar la contraseña del Apple ID.

**Centro de notificaciones**

Con la actualización iOS 5, el sistema de notificaciones se rediseñó por completo. Las notificaciones ahora se colocan en un área por la cual se puede acceder mediante un deslice desde la barra de estado hacia abajo. Al hacer un toque en una notificación el sistema abre la aplicación. La pantalla inicial de iOS contiene varias aplicaciones, algunas de las cuales están ocultas por defecto y pueden ser activadas por el usuario mediante la aplicación "Ajustes", por ejemplo, Nike+iPod o la app de iCloud Drive deben ser activadas desde los Ajustes.

Todas las «utilidades», como Notas de Voz, Reloj, Brújula y Calculadora están en una carpeta llamada «Utilidades» desde la versión 4.0.12​ Varias de las aplicaciones incluidas están diseñadas para trabajar juntas, permitiendo compartir datos de una aplicación a otra. (Por ejemplo, un número de teléfono puede ser seleccionado desde un correo electrónico y guardarlo como un contacto o para hacer una llamada)

El iPod Touch tiene las mismas apps que están presentes en el iPhone, con excepción de Teléfono, Mensajes (aunque sí iMessage) y Brújula. Hasta iOS 5, en el iPhone y el iPad los iconos de música y videos estaban juntos en una sola aplicación, pero luego se separaron en 2, Música y Videos. Por defecto, en el dock del iPhone, ubicado en la parte inferior de la pantalla de inicio, están los iconos "estrella", que son Teléfono, Mail, Safari y Música. En el iPod touch, estos iconos son Música, Safari, Mail y iMessage.

El iPad también tiene las mismas aplicaciones que el iPhone, excluyendo Bolsa, Tiempo, Calculadora, Voice Memos, Teléfono, Mensajes (aunque si iMessage) y Nike+iPod, apps separadas para música y vídeo igualmente se usan (como en el iPhone). Varias apps por defecto están reescritas para tomar ventaja de la pantalla más grande. El dock por defecto incluye Safari, Mail, Fotos y Música. Desde iOS 6 en adelante, el iPad 2 y iPad 3 tienen la aplicación de Reloj.

**Multitarea Opcional**

Antes de iOS 4, la multitarea estaba reservada para aplicaciones por defecto del sistema. A Apple le preocupaba los problemas de batería y rendimiento si se permitiese correr varias aplicaciones de terceros al mismo tiempo. Para iniciar la multitarea hay que apretar 2 veces el botón del centro. La multitarea sólo es compatible desde el iPhone 3GS, iPad 1, iPod Touch (3.ª generación) A partir de iOS 4, dispositivos de tercera generación y posteriores permiten el uso de 7 APIs para multitarea, específicamente:

Audio en segundo plano

Voz IP

Localización en segundo plano

Notificaciones push

Notificaciones locales

Completado de tareas

**Game Center**

Fue anunciado en el evento donde se presentó iOS 4 el 8 de abril de 2010. Game Center se lanzó en junio de 2010 para los iPhone y iPod Touch con iOS 4 (excepto para el iPhone 2G, 3G y iPod Touch 1g ). En iOS 5 se perfeccionó, pudiendo agregar una foto a tu perfil, pudiendo ver los amigos de tus amigos y pudiendo encontrar adversarios con recomendaciones de nuevos amigos en función de tus juegos y jugadores favoritos. Actualmente iOS 9.3.5 es la última versión que cuenta con Game Center, puesto que desde el lanzamiento de iOS 10 se ha eliminado del menú de inicio y de la App Store.

**Tecnologías no admitidas**

Las versiones anteriores a iOS 8 no permiten el uso de la Plataforma Java y Adobe Flash, de hecho Steve Jobs escribió una carta abierta donde critica a Flash por ser inseguro, con errores, consumir mucha batería, ser incompatible con interfaces multitouch e interferir con el servicio de la App Store.13​ En cambio iOS usa HTML5 como una alternativa a Flash. Esta fue una característica muy criticada en su momento, tal fue el grado que a partir de iOS 8 ya se permite el uso de Adobe Flash; sin embargo los poseedores de un iPhone, iPod touch o iPad con una versión anterior a iOS 8 quedaron fuera de esto; pero de igual forma se puede añadir Adobe Flash comprando un navegador especial en la App Store o haciendo el Jailbreak para poder utilizarlo en Safari (esto último anularía la garantía).

Durante los últimos meses de 2010 en el Silverlight Firestarter event14​ se especulaba sobre la inclusión en iOS de Silverlight 5, y con "transmuxing"15​ se revelaba un gran avance para iOS en esta dirección, pero con los últimos sucesos en los que Microsoft considera redirigir sus intereses a HTML516​ el futuro de Silverlight es incierto y por el momento es casi imposible encontrar contenido Silverlight disponible en dispositivos con iOS

# 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 2014.1​.

## 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.