**Tutorial de los temas**

* AngularJS
* IoinicJS
* CORDOVA
* ANDROID
* IOS
* FIREBASE

Clave: 7

Grado: 5to bachillerato en computación

Sección: ”A”

**Índice**

AngularJS………………………………………………………………………………………………………………………3-5

Lonic.js………………………………………………………………………………………………………………………….6-10

Córdova…………………………………………………………………………………………………………………………11-13

Android…………………………………………………………………………………………………………………………14-16

IOS………………………………………………………………………………………………………………………………..17-19

Firebase………………………………………………………………………………………………………………………..19-23

Recomendaciones………………………………………………………………………………………………………..24-28

Infografías…………………………………………………………………………………………………………………….29

**AngularJS:**

**Qué necesitas para desarrollar con AngularJS**

Lo único que necesitas para desarrollar con AngularJS es un editor de texto y un navegador. Así de sencillo, seguramente todo el mundo tenga ya configurado su editor preferido, así que te puedes saltar si quieres estos siguientes párrafos. Para los que no tengan claro esto, os dejo un par de comentarios.

* **Editor de código:** puede ser cualquiera que estés acostumbrado a usar, no necesitas cambiarlo para programar con Angular. Lo que es interesante es que sea un editor para programadores, que te permita diversas ayudas al escribir el código, coloreado de sintaxis, ayudas contextuales, etc. Notepad++, Sublime Text, Komodo Edit, Brackets, etc. Cualquiera es bueno. Si estás acostumbrado a IDEs más complejos como Eclipse o PhpStorm, puedes quedarte también con ellos.
* **Navegador:** Puedes usar cualquier navegador para ver un desarrollo basado en Angular. Generalmente tendrás incluso varios navegadores para probar tu página en cada uno de ellos y comprobar que todo está en orden. Solo te recomendamos tener a Google Chrome entre tu batería de navegadores, pues luego hablaremos de una extensión que existe para éste que te permite examinar y depurar páginas donde AngularJS está trabajando.

**Descarga de AngularJS**

Si quieres trabajar con AngularJS tienes que incluir el script del framework en tu página. Esto lo puedes hacer de varias maneras, o bien te descargas la librería por completo y la colocas en un directorio de tu proyecto, o bien usas un CDN para traerte la librería desde un servidor remoto. En principio es indiferente a nivel didáctico, así que nosotros vamos a comenzar del modo más sencillo, que es utilizar el CDN.

Accedes a la página de AngularJS: https://angularjs.org/

Pulsas el botón de descarga y encontrarás diversas opciones. Escoges la versión del framework (si es que te lo permite) y que esté minimizada (minified). Luego encontrarás un campo de texto donde está la URL de la librería (esa URL está marcada con las siglas "CDN" al lado). Ya sabes que el CDN te ofrece un contenido, en este caso un script Javascript, que está alojado en otro servidor, pero que lo puedes usar desde tu página para mejorar la entrega del mismo.

**Nota:** La versión minimizada pesa menos en Kb, por lo que será la adecuada para cualquier sitio que tengas en producción. Sin embargo la versión sin comprimir "Uncompressed" tiene el código completo, con comentarios, sangrado, etc. lo que hace que sea más fácil de leer por humanos. Puedes usar si lo deseas la versión sin comprimir en la etapa de desarrollo.

Será algo como https://ajax.googleapis.com/ajax/libs/angularjs/1.2.23/angular.min.js aunque todo depende de la versión que haya en el momento en el que tú leas esta guía de iniciación.

**Incluir AngularJS en una página web**

Una vez tienes tu CDN puedes incluir el script de Angular en la página con la etiqueta SCRIPT. Ese script lo puedes colocar en el HEAD o bien antes del final del BODY, en principio no habría diferencias en lo relativo a la funcionalidad, pero sí hay una pequeña mejora si lo colocas antes de cerrar el cuerpo.

Simplemente, si lo colocas en el HEAD estás obligando a que tu navegador se descargue la librería de AngularJS, retrasando quizás la descarga de áreas de la página con contenido. Si lo colocas antes de cerrar el BODY facilitas la vida a tu navegador, y por añadido a tus usuarios, pues podrá descargar todo el HTML, ir renderizando en la pantalla del usuario los contenidos sin entretenerse descargando AngularJS hasta que sea realmente necesario.

**Declarar directivas**

Hay un paso más para dejar lista una página donde quieras usar AngularJS. Es simplemente colocar la directiva ng-app en la etiqueta que englobe la aplicación. Más adelante hablaremos con más detalle de las directivas y daremos algunos tips para usarlas mejor. Por ahora puedes quedarte simplemente con la necesidad de informar a AngularJS del contenedor HTML donde va a desplegar su "magia".

Típicamente pondrás ng-app en la etiqueta HTML de inicio del documento.

<!DOCTYPE html>

<html lang="en" ng-app>

<head>

<meta charset="UTF-8">

<title>Ej de AngularJS</title>

</head>

<body>

... Aquí el cuerpo de tu página ...

<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.2.23/angular.min.js"></script>

</body>

</html>

Así como ng-app, existen muchas otras directivas para enriquecer tu HTML, aun veremos alguna más en este artículo.

**IoinicJS:**

# **Qué es y cómo empezar con Ionic Framework**

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](http://www.phonegapspain.com/que-es-y-como-empezar-con-angularjs/) 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 ;)



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**

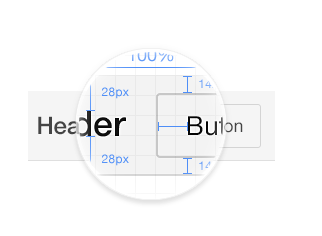
[Ionic](http://ionicframework.com/) es una herramienta, gratuita y open source, para el desarrollo de aplicaciones híbridas basadas en HTML5, CSS y JS. Está construido con [Sass](http://sass-lang.com/) y optimizado con [AngularJS](https://angularjs.org/).

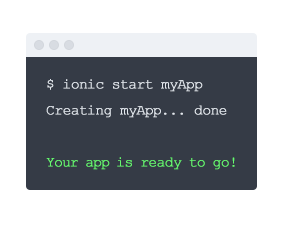
**Principales características**

  
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.

  
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.

  
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.

  
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.

  
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](http://phonegap.com/) o [Cordova](http://cordova.apache.org/). 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



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

**Code:** arbitrary(select)

1.  
2.  
3.  
4.

$ 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](https://creator.ionic.io/)“. Esta herramienta es muy parecida a [codiqa](https://codiqa.com/), 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.

**Cordova:**

Pre-requisitos

Antes de ejecutar cualquiera de las herramientas de línea de comandos, necesita instalar el SDK para cada plataforma. (Vea las guías de la plataforma para más detalles)

Para añadir soporte o reconstruir un proyecto para cualquier plataforma, necesitará ejecutar la interfaz de línea de comandos desde la misma máquina que soporta el SDK de la plataforma. La CLI admite las siguientes combinaciones:

* iOS (Mac)
* Amazon Fire OS (Mac, Linux, Windows)
* Android (Mac, Linux, Windows)
* BlackBerry 10 (Mac, Linux, Windows)
* Windows Phone 8 (Windows)
* Windows (Windows)
* Firefox OS (Mac, Linux, Windows)

En el Mac, es disponible a través de la aplicación de la *Terminal de* la línea de comandos. En el PC, está disponible como *Command Prompt* bajo *Accessories*.

**Nota**: para las plataformas Windows, todavía puedes desarrollar en equipos Mac ejecutando Windows en una máquina virtual o en modo de arranque dual. Para las opciones disponibles, consulte la guía de la plataforma de Windows Phone 8 o la guía de la plataforma Windows.

Lo más probable es que ejecute la CLI de diferentes máquinas, más tiene sentido mantener un repositorio de código fuente remota, cuyos activos tire hacia abajo para directorios de trabajo local.

Instalar la CLI de Cordova

La herramienta de línea de comandos de Cordova se distribuye como un paquete de npm en un formato listo para usar. No es necesario compilarlo desde su código fuente.

Para instalar la herramienta de línea de comandos de cordova, siga estos pasos:

1. Descargue e instale [Node.js](http://nodejs.org/). Después de la instalación, usted debe ser capaz de ejecutar node y npm en la línea de comandos. Si lo desea, opcionalmente puede utilizar una herramienta como nvm o nave para manejar la instalación de Node.js.
2. Descarga e instala un [cliente de git](http://git-scm.com/), si ya no tienes uno. Después de la instalación, usted debe ser capaz de ejecutar a git en la línea de comandos. Aunque no utilices git manualmente, la CLI usa de fondo para descargar algunos archivos cuando se crea un nuevo proyecto.
3. Instale el módulo cordova utilizando el manejador de paquetes de Node.js npm. El módulo cordova sera descargado automáticamente por npm.

* en OS X y Linux:
* $ sudo npm install -g cordova

En OS X y Linux, prefijando la npm mando con sudo puede ser necesario instalar este desarrollo utilidad en otro modo restringido directorios tales como /usr/local/share . Si usted está utilizando la herramienta opcional nvm/nave o tener acceso de escritura al directorio de instalación, podrá omitir el sudo prefijo. Hay [más consejos](http://justjs.com/posts/npm-link-developing-your-own-npm-modules-without-tears) sobre el uso de npm sin sudo , si desea hacerlo.

* en Windows:
* C:\>npm install -g cordova

El -g bandera arriba dice npm instalar cordova en todo el mundo. De lo contrario será instalado en el node\_modules subdirectorio del directorio de trabajo actual.

Puede que necesites añadir el npm Directorio a su PATH para invocar a nivel mundial instalada npm módulos. En Windows, npm generalmente se puede encontrar en C:\Users\username\AppData\Roaming\npm . En OS X y Linux se puede encontrar generalmente en/usr/local/share/npm.

El registro de instalación puede producir errores para cualquier plataforma desinstalado SDK.

Después de la instalación, usted debe ser capaz de ejecutar cordova en la línea de comandos sin argumentos y debe imprimir el texto de ayuda.

Crear la aplicación

Vaya al directorio donde mantener su código fuente y ejecutar un comando como el siguiente:

$ cordova create hello com.example.hello HelloWorld

Puede tomar algún tiempo para que el comando completar, así que tenga paciencia. Ejecutar el comando con el -d opción muestra información acerca de su progreso.

El primer argumento *Hola* especifica un directorio que se generen para su proyecto. Este directorio ya no debería existir, Córdoba lo creará para usted. Página de inicio de su aplicación, junto con diversos recursos bajo css, js y img, que seguir común web convenciones de nomenclatura de archivos de desarrollo las casas su subdirectorio www. Estos activos se almacenarán en el sistema de archivos local del dispositivo, no sirve de forma remota. El archivo config.xml contiene metadatos importantes necesarios para generar y distribuir la aplicación.

El segundo argumento com.example.hello proporciona un identificador de dominio reverso-estilo su proyecto. Este argumento es opcional, pero sólo si también omite el tercer argumento, puesto que los argumentos son posicionales. Puede editar este valor más adelante en el config.xml de archivos, pero tenga en cuenta que puede haber código generado fuera de config.xml utilizando este valor, tales como nombres de paquetes Java. El valor predeterminado es io.cordova.hellocordova , pero se recomienda que seleccione un valor apropiado.

El tercer argumento HelloWorld da título de pantalla de la aplicación. Este argumento es opcional. Puede editar este valor más adelante en el config.xmlde archivos, pero tenga en cuenta que puede haber código generado fuera de config.xml utilizando este valor, tales como nombres de clase de Java. El valor predeterminado es HelloCordova , pero se recomienda que seleccione un valor apropiado.

Añadir plataformas

Todos los comandos posteriores necesitan ejecutarse en el directorio del proyecto, o cualquier subdirectorios dentro de su ámbito de aplicación:

$ cd hello

Antes de poder construir el proyecto, tienes que especificar un conjunto de plataformas de destino. Su capacidad para ejecutar estos comandos depende de si tu maquina soporta cada SDK, y si ya tienes instalación cada SDK. Ejecutar cualquiera de éstos desde un Mac:

$ cordova platform add ios

$ cordova platform add amazon-fireos

$ cordova platform add android

$ cordova platform add blackberry10

$ cordova platform add firefoxos

Ejecutar cualquiera de éstos desde una máquina Windows, donde *wp* se refiere a las diferentes versiones del sistema operativo Windows Phone:

plataforma $ cordova agregar $ wp8 cordova plataforma añadir windows plataforma $ cordova añadir plataforma amazon-fireos $ cordova añadir android $ cordova plataforma añadir $ blackberry10 cordova plataforma agregar firefoxos

Ejecutar esto para comprobar su sistema actual de plataformas:

$ cordova platforms ls

(Tenga en cuenta que los comandos platform y platforms son sinónimos).

Ejecutar cualquiera de los siguientes comandos sinónimos para quitar una plataforma:

$ cordova platform remove blackberry10

$ cordova platform rm amazon-fireos

$ cordova platform rm android

Ejecución de comandos para agregar o quitar afecta a plataformas el contenido del directorio de *platforms* del proyecto, donde cada plataforma especificado aparece como un subdirectorio. El directorio de origen *www* se reproduce dentro del subdirectorio de cada plataforma, que aparece por ejemplo en platforms/ios/www o platforms/android/assets/www. Porque la CLI constantemente copia archivos desde la carpeta *www* fuente, sólo debe editar estos archivos y no los ubicado debajo de los subdirectorios de *plataformas* . Si utilizas software de control de versión, debe agregar esta carpeta *www* fuente, junto con la carpeta *se*

**Android:**

# [**TUTORIAL DE AYUDA Y PRIMEROS PASOS**](https://androidayuda.com/tutorial-de-ayuda-y-primeros-pasos/)

Tanto si ya habías oído hablar del sistema operativo para Smartphone de Google como si es la primera vez que escuchas hablar de él, existen muchas posibilidades de que el próximo teléfono inteligente que vayas a adquirir funcione con Android. La siguiente guía te ayudará con los primeros pasos en este nuevo universo para que puedas sacarle todo el jugo a tu teléfono desde el primer momento.

En las siguientes secciones iremos aconsejándote por todos los pasos necesarios para familiarizarte con tu nuevo teléfono. Desde consejos para comprar un terminal hasta las mejores aplicaciones o trucos avanzados, todos podréis encontrar una manera fácil y rápida de familiarizaros con el sistema operativo que ha revolucionado los dispositivos móviles.

**1. ¿Qué teléfono vas a comprar?**

Si todavía no tenéis un teléfono Android pero os apetecería dar el salto quizás estáis en un mar de dudas debido a la cantidad de opciones y precios que ofrece el mercado. Esta es una de las primeras cosas que debéis tener en cuenta a la hora de pensar en Android: la fragmentación.

Pero **¿qué significa esa fragmentación?** Pues simplemente que existe una variedad enorme de marcas, modelos, precios y configuraciones diferentes de teléfonos con sistemas operativos Android. Por lo tanto, antes de decidiros en un dispositivo en particular debéis tener una serie de aspectos en cuenta.

Lo primero que hay que pensar evidentemente es para qué queréis el teléfono. Si sólo lo necesitáis para**jugar a juegos sencillos**, navegar y escuchar música no necesitáis gastaros un dineral en un terminal con **4 núcleos y varios gigas de RAM**. Los teléfonos Android de “clase media” funcionan muy bien y son muy competitivos en precio.

Muchos de vosotros habréis estado ya investigando terminales y configuraciones, y quizás os preguntáis qué son esos nombres como **Gingerbread, Ice Cream Sandwich o Jelly Bean**… pues se trata de versiones del sistema operativo. Android 3.2.6 “Gingerbread” por ejemplo, es la versión más común en estos momentos, y tras ella tenemos la versión 4 llamada “Ice Cream Sandwich”. A finales de año, los terminales con la nueva versión 5.0 “Jelly Bean” comenzarán a aparecer en el mercado.

Cuanto más actualizada sea vuestra versión de vuestro SO más alta será la probabilidad de que vuestro teléfono sea compatible con la mayoría de las aplicaciones de **Google Play** y por lo tanto podréis sacarle partido a vuestro terminal durante más tiempo. Pero ya hablaremos un poco más tarde sobre ello.

Como resumen, deberíais tener en cuenta la mejor opción considerando las siguientes variables: procesador, batería, precio y versión del SO. El teléfono que mejor puntúe en esos aspectos debería ser competente para satisfacer vuestras necesidades medias de manera suficiente por un tiempo más que razonable.

**2. ¿Plan o no plan?**

Tras elegir el afortunado terminal que pasará a engrosar nuestra más o menos nutrida lista de teléfonos móviles llega quizás la segunda decisión más importante. Android es un sistema concebido para que vuestro teléfono esté conectado a internet continuamente, pero ello no quiere decir que no se pueda usar sin un plan de datos.

Lo más normal es que vuestro terminal tenga la opción de conectarse a **redes WIFI**, por lo que mucha gente que no necesita consultar su correo electrónico constantemente o navegar desde un interminable atasco se decide por conectarse simplemente cuando alguna red se lo permite. Ya sea desde nuestro hogar, el trabajo, la facultad, la casa de algún amigo o utilizando cualquiera de las redes abiertas que inundan cada a vez más las ciudades españolas, las posibilidades de conectarse a Internet de manera gratuita son cada vez más numerosas.

Por el contrario, si no tenéis una tarifa de datos muchas de vuestras aplicaciones de redes sociales, correo, navegación o cualquier otra que utilice información en tiempo real serán poco menos que inservibles. De vosotros depende la decisión.

**3. Primera configuración**

Ya sea un teléfono de gama media y precio ajustado o una auténtica “mala bestia” con un procesador de última generación que nos haya costado el sueldo de un mes, todos los Android deben pasar por la misma fase: la **configuración inicial**.

¿Y ahora qué? Esta es la pregunta que la mayoría de los novatos en Android nos hemos hecho la primera vez al encender nuestro nuevo y flamante terminal. La **configuración inicial** es un paso obligatorio en todo teléfono Android que asociará el terminal a tu cuenta de Google para poder sincronizar tus contactos, calendario y otros servicios que utilices en tu dispositivo con la nube de Google.

Para muchos puede parecer un paso extraño, pero a la larga podréis admirar las ventajas de no volver a temer la pérdida de todos vuestros contactos si perdéis vuestro teléfono, por ejemplo.

**4. Explora**

Una vez configurada tu cuenta de Google, el teléfono se pone en marcha… es el momento de sumergirse en el universo de menús y configuraciones que te ofrece Android. A pesar de que la personalización de un terminal es algo que para muchos nunca acaba, los primeros pasos os ayudarán a perder el miedo a**añadir o eliminar widgets de escritorio**, el aspecto de los menús o del launcher de vuestro terminal.

Lo más importante en esta etapa es conocer las opciones de modificación que vuestro terminal ofrece y sobre todo comprender cómo funciona. Android es un sistema que ofrece infinidad de opciones de configuración y cuanto más os adentréis en él más o picará el gusanillo de ir un poco más allá.

También es el momento de comenzar a utilizar las aplicaciones que más vais a usar, como por ejemplo **GMail, Market, Calendario, el navegador por defecto** o vuestra agenda de contactos. Todas estas aplicaciones se pueden sustituir por otras en Google Play, pero en general, las aplicaciones nativas en Android son muy competentes y para mucha gente más que suficientes. También os recomendamos comenzar a trastear con los diferentes escritorios, una de las ventajas de Android es poder disponer de varios en paralelo y agrupar temáticamente en ellos las aplicaciones que más utilicemos. Gracias a los widgets, también podemos hacer funcionar pequeñas aplicaciones como relojes o calendarios directamente en esos escritorios.

**5. Descarga algunas aplicaciones del Market**

Un vez que ya nos hemos familiarizado con la configuración y funcionamiento básico de nuestro teléfono llega el momento más divertido: sumergirse en el [Market o Google Play](https://market.android.com/). La tienda de aplicaciones de Google es el pilar clave de la filosofía de Android, su mejor cualidad y su mayor defecto al mismo tiempo.

**Ios:**

### **¿Cómo puedo configurar mi correo?**

Pulsa Mail en la pantalla de inicio de tu dispositivo iOS.

Selecciona tu proveedor de correo electrónico (**Hotmail**, **GMAIL**, **Yahoo**, etc). Si tu proveedor de correo electrónico no aparece en la lista, pulsa Otras > Añadir cuenta.

Si ya has configurado una cuenta de correo electrónico y quieres modificar sus ajustes, pulsa Ajustes > Correo, contactos, calen., pulsa el nombre de tu cuenta y, a continuación, tu dirección de correo electrónico. Después pulsa cualquier información que quieras modificar.

**¿Qué es y cómo funciona la App Store?**

App Store es un servicio para el [iPhone](https://www.aboutespanol.com/tecnologia-4147629), el iPod Touch y el iPad.

Permite a los usuarios buscar y descargar aplicaciones informáticas de iTunes Store. Estas aplicaciones están disponibles para ser compradas o bien gratuitas, dependiendo de cada una. Las aplicaciones pueden ser descargadas directamente al iPhone.

**¿Cómo creo una cuenta de iTunes?**

Para crear una cuenta de iTunes, puedes hacerlo desde el propio dispositivo o desde la computadora.

Si optas por la primera opción, solo tienes que acceder a**Ajustes > Store > crear nuevo ID de Apple.**

Selecciona la tienda adecuada pulsando Store y elige el país en el que te encuentres.

Pulsa Aceptar para confirmar tu selección y pulsa Siguiente para continuar.

En la siguiente pantalla aparecerán los Términos y condiciones de **iTunes Store**. Pulsa el botón Acepto que se encuentra situado en la parte inferior derecha de la página. Pulsa de nuevo Acepto cuando aparezca la confirmación.

Si no lo consigues de esta forma, [aquí tienes otro tutorial](https://www.aboutespanol.com/como-crear-una-cuenta-de-itunes-1153682) donde quizás lo encuentres mejor explicado.

**¿Cómo puedo instalar aplicaciones y juegos nuevos?**

1. Entra en AppStore en el iPhone
2. Ahora busca la aplicación deseada. Puedes buscar por nombre, categoría, preferidas por los usuarios… Todo ello desde el menú inferior.
3. Una vez que la seleccionas, nos dará información adicional, capturas de la aplicación y comentarios de los usuarios.
4. Pulsa en Gratis (Free en inglés) si la aplicación es gratuita o en el precio si es de pago.
5. Pulsa ahora en el botón Instalar (Install en inglés) que aparece en lugar del anterior.
6. Ahora te pedirá la contraseña de tu cuenta con iTunes.
7. Al introducirla correctamente te llevará directamente a la pantalla principal, donde te aparecerá la nueva aplicación con una barra de estado mientras se instala.
8. Al finalizar la descarga ya tendrás la aplicación lista para usar en el menú de inicio.

**¿Cómo borrar aplicaciones?**

Busca en el menú principal la aplicación que quieres eliminar, pulsa el icono y manténlo presionado hasta que comience a temblar y aparezca una “X” que se verá en la esquina superior del icono.

Enseguida obtendrás un ventana de dialogo que pregunta si estas seguro de eliminar la aplicación. Confirma, y automáticamente se eliminaran todos los datos asociados con la aplicación.

**¿Cómo puedo actualizar una aplicación instalada?**

Consulta si hay una versión actualizada de la aplicación de tu iPhone.

Abre la tienda App Store y pulsa Actualizar. Si hay aplicaciones disponibles, pulsa Actualizar todas.

Cuando se te solicite, introduce la información de tu cuenta de iTunes Store. El iPhone descargará e instalará las actualizaciones para [todas las aplicaciones de iPhone.](https://www.aboutespanol.com/tecnologia-4147629)

En el nuevo iOS se pueden actualizar automáticamente si así lo deseas.

**¿Qué son las notificaciones Push?**

Las notificaciones Push son un invento sin el que la telefonía móvil moderna no podría existir. Se trata de avisos instantáneos de eventos, nuevos correos, menciones en Twitter, etc.

Estas notificaciones son una gran ayuda cuando hablamos de aplicaciones de comunicación, como WhatsApp, Facebook o la comentada anteriormente, Twitter. Los usuarios saben de inmediato quién se está queriendo poner en contacto con ellos y qué quieren saber.

**¿Cómo puedo actualizar mi dispositivo?**

Hacerlo es muy fácil, en la primera pestaña de iTunes llamada “Resumen”, encontrarás dos grandes botones: “Restaurar” y “Actualizar”. Pulsa sobre Actualizar para que iTunes descargue de forma inmediata la nueva versión de iOS y actualice tu terminal de forma automática.

**Firebase:**

**Primeros pasos con Firebase**

**En estos artículos veremos cómo dar los primeros pasos con Firebase, tanto para la versión antigua de Firebase como para la versión nueva. El la nueva versión cambiaron algunas cosas, como el script de inicialización y una manera nueva de acceder a los servicios ofrecidos por Firebase. Te lo explicamos todo para que no te pierdas cuando estás comenzando.**

### 1.- [Introducción a Firebase](https://desarrolloweb.com/articulos/introduccion-firebase-backend-nube.html)

Servicio capaz de proveernos de un Backend en la nube con una fuente de datos en tiempo real y librerías para acceder a ella desde aplicaciones web, iOS y Android.

### 2.- [Firebase 3](https://desarrolloweb.com/articulos/nuevo-firebase.html)

Novedades y primeros pasos en la nueva versión de Firebase, que nos trae una importante cantidad de nuevos servicios para desarrollo de aplicaciones web, iOS y Android sin tener que desarrollar la parte backend.

### 3.- [Clase práctica de introducción a Firebase 3](https://desarrolloweb.com/articulos/presentacion-firebase-practico.html)

Clase para comenzar con Firebase para la web, realizando una aplicación donde podamos experimentar con varios de los servicios de Firebase 3.

**Base de datos en tiempo real**

**Te explicamos las bases de trabajo para el uso de uno de los servicios de Firebase, la base de datos en tiempo real. En estos artículos exploramos el SDK para la web con Javascript, mediante los cuales podemos acceder a los datos de la aplicación, tanto para lecturas como escrituras.**

### 4.- [Escribir datos en Firebase desde Javascript](https://desarrolloweb.com/articulos/escribir-datos-firebase-javascript.html)

Cómo realizar la escritura y modificación de un dato de tipo objeto dentro de Firebase, usando programación para web con Javascript.

### 5.- [Trabajo con colecciones en Firebase y Javascript](https://desarrolloweb.com/articulos/trabajo-colecciones-firebase-javascript.html)

Aprendemos a crear colecciones de elementos dentro de Firebase, acceder a ellas y suscribirnos a cambios en las colecciones, tanto si se generan nuevos elementos como si éstos cambian.

**Sistema de autenticación de Firebase**

**A continuación vamos a explorar las funcionalidades del sistema de autenticación de Firebase. Es otro de los servicios fundamentales del BaaS que nos permite loguear usuarios tanto con el tradicional mail/clave como con conectores sociales de los principales proveedores: Facebook, Google, Twitter y Github.**

### 6.- [Autenticación en Firebase y Javascript por medio de usuario y password](https://desarrolloweb.com/articulos/autenticacion-firebase-js-usuario-password.html)

Mecanismos para autenticación de usuarios en aplicaciones bajo el servicio de Firebase, con código Javascript.

### 7.- [Autenticación de usuarios con redes sociales en Firebase 3](https://desarrolloweb.com/articulos/autenticacion-usuarios-rrss-firebase.html)

Cómo autenticar usuarios en una aplicación usando cuentas de Facebook, Twitter, Google o Github, de una manera sencilla con Firebase.

### 8.- [Flujo de autenticación Javascript con Firebase usando usuarios de Google](https://desarrolloweb.com/articulos/flujo-autenticacion-javascript-firebase.html)

En este ejercicio vamos a mostrar cómo realizar una página que permite autenticarse usando el usuario de Google, solo con Javascript y el SDK de Firebase.

### 9.- [Datos personales de usuarios entregados por login social con Firebase](https://desarrolloweb.com/articulos/datos-personales-login-social-firebase.html)

Cómo acceder y almacenar en nuestra propia base de datos la información personal de los usuarios que hacen login social en nuestra aplicación web Firebase.

**Firebase Storage**

**A continuación veremos una serie de artículos dedicados a explicar el servicio de almacenamiento de archivos de Firebase: Storage. Es un servicio muy interesante, porque nos permite alojar cualquier cantidad de archivos, usando la nube de Google, que tiene una capacidad virtualmente ilimitada. A partir de Firebase Storage podremos enviar datos a la nube, con opciones completas de control de las subidas, tolerancia a redes con mala calidad o poco ancho de banda, incluso cortes puntulaes de red. Aprenderemos a monitorizar las subidas y controlarlas, así como realizar download de archivos, sus reglas de seguridad, etc. Al final, lo más extraordinario es que puedes hacer todo con código Javascript del lado del cliente, evitando disponer de código backend o servidores propios.**

### 10.- [Introducción a Firebase Storage](https://desarrolloweb.com/articulos/introduccion-firebase-storage.html)

Explicamos todo lo que debes saber para comenzar con el servicio Storage de Firebase, junto con un ejemplo sencillo para hacer el upload, operación put, desde una página web.

### 11.- [Controlar el proceso de la subida con Storage de Firebase](https://desarrolloweb.com/articulos/proceso-subida-storage-firebase.html)

Te explicamos cómo realizar un ejemplo más avanzado de upload de archivos a Firebase Storage en el que controlaremos el proceso y estado de la subida.

### 12.- [Reglas de seguridad en Firebase Storage](https://desarrolloweb.com/articulos/reglas-seguridad-firebase-storage.html)

Cómo construir las reglas de seguridad de Firebase Storage, manteniendo seguros los archivos en tu espacio de almacenamiento.

**Otros asuntos importantes de Firebase**

**Otros artículos sobre servicios importantes de Firebase que sin duda necesitarás para desarrollar aplicaciones.**

### 13.- [Reglas de seguridad en Firebase](https://desarrolloweb.com/articulos/reglas-seguridad-firebase.html)

Esta es una introducción a las reglas de seguridad en Firebase, por qué se hacen, cómo se implementan y ejemplos sencillos para mantener seguros los datos de tu aplicación.

### 14.- [Firebase Tools](https://desarrolloweb.com/articulos/firebase-tools.html)

Qué son las Firebase Tools, cómo obtenerlas y un resumen de los comandos que nos ofrecen para mantener, desarrollar y desplegar aplicaciones en Firebase.

**Polymerfire**

**Explicaciones sobre el uso de Firebase en aplicaciones realizadas con la librería Polymer. Polymer tiene una serie de componentes que permiten trabajar con Firebase de una manera muy ágil y sencilla. Los analizaremos con varios ejemplos que seguro que sorprenderán y gustarán al lector.**

### 15.- [Introducción a Polymerfire (Firebase con Polymer)](https://desarrolloweb.com/articulos/introduccion-polymerfire.html)

Pasos para realizar una aplicación Firebase con Polymer, usando Firebase 3 y los componentes de Polymerfire.

### 16.- [Componente firebase-query de Polymerfire](https://desarrolloweb.com/articulos/componente-firebase-query-polymerfire.html)

Uno de los componentes de Polymerfire, para el trabajo con la base de datos de Firebase, que nos ofrece el acceso a colecciones de elementos.

### 17.- [Componente firebase-auth](https://desarrolloweb.com/articulos/componente-firebase-auth.html)

El componente de Polymerfire que nos provee del servicio de autenticación de usuarios en aplicaciones Firebase y Polymer.

**Firebase desde el servidor**

**Firebase también se puede acceder desde el backend, y de hecho es algo que aplicaciones medianas y complejas necesitarán con toda probabilidad, pues no toda la lógica de aplicación o de negocio se puede implementar con seguridad del lado del cliente. Además hay ciertas funcionalidades de los servicios de Firebase que solo se puede hacer desde un entorno seguro de servidor.**

### 18.- [Firebase 3 desde el servidor NodeJS](https://desarrolloweb.com/articulos/firebase3-servidor-nodejs.html)

Cómo configurar tu aplicación de Firebase para acceder desde NodeJS, configurable en un servidor, desde el que hacer tareas administrativas.

### 19.- [Problemas comunes del hosting de Firebase y soluciones](https://desarrolloweb.com/articulos/problemas-soluciones-hosting-firebase.html)

Cómo configurar correctamente apps para alojar en el hosting de Firebase para evitar problemas, centrándonos en las aplicaciones SPA (Single Page Application) y Polymer.

**Notificaciones Push con Firebase**

**En los próximos artículos te explicaremos todo lo que debes saber para poder enviar notificaciones Push para sitios web, usando el servicio de Firebase Cloud Messaging, que es gratuito para cualquier persona o empresa. Con las notificaciones podrás aumentar la interacción de los visitantes, enviando mensajes con notificaciones, incluso sin que ellos estén conectados a tu sitio web en ese momento.**

### 20.- [Integrantes del servicio de notificaciones push desde la web](https://desarrolloweb.com/articulos/notificaciones-push-web-firebase.html)

Explicamos las cosas que debes de saber antes de enviar notificaciones al usuario a través de web con Firebase.

### 21.- [Envío de notificaciones push con Firebase y Javascript](https://desarrolloweb.com/articulos/notificaciones-push-firebase-javascript.html)

Cómo configurar un sitio web para poder solicitar y recibir el envío de notificaciones Push con el navegador, usando el API de Firebase y Javascript del lado del cliente.

**Recomendaciones.**

AngularJS.

[AngularJS](https://angularjs.org/) es uno de los *frameworks* JavaScript que más popularidad ha alcanzado, no sólo por llevar conceptos como [Inyección de Dependencias](https://en.wikipedia.org/wiki/Dependency_injection) y Modelo Vista Controlador ([MVC](https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller)) al desarrollo con JavaScript (ya que otros *frameworks* como [Ember.js](http://emberjs.com/) o [Backbone.js](http://backbonejs.org/) tienen algunas de esas características) sino que también tiene el soporte de una organización como Google, lo cual genera cierta confianza junto con el [creciente número de la comunidad](https://www.airpair.com/js/javascript-framework-comparison#3-community) de desarrolladores que lo usan.

Sin embargo es relativamente común encontrar comentarios, principalmente de quienes venimos de trabajar con lenguajes y *frameworks* más estructurados (Java, C# o inclusive Ruby), que indican que desarrollar aplicaciones web de tamaño y complejidad considerable usando estos *frameworks*JavaScript terminan siendo difíciles de mantener en el mediano plazo debido no solamente a la naturaleza dinámica de JavaScript como lenguaje, sino también a la falta de consenso en la adopción de prácticas entre las comunidades de desarrolladores.  
  
Google en su momento publicó una serie de [estándares con estilos y mejores prácticas](http://angularjs.blogspot.com.co/2014/02/an-angularjs-style-guide-and-best.html) tanto en código JavaScript como en la estructura de carpetas y archivos para desarrollo de aplicaciones web usando AngularJS, que aunque no son de obligatorio seguimiento tenían como propósito unificar un poco los criterios de la comunidad desarrolladora y proponer un esquema más fácil de mantener en todo el ciclo de vida las aplicaciones.  
  
Actualmente el propio equipo de Google recomienda usar los [estándares propuestos por la comunidad](https://github.com/johnpapa/angular-styleguide) (principalmente John Papa y Todd Motto), lo cuales aunque un poco largos definitivamente vale la pena revisarlos, aunque aquí se tratarán algunos allí documentados.  
  
Teniendo en cuenta dichos estándares se desarrollará una pequeña aplicación para consultar el clima de una ciudad dentro de una lista disponible, con el propósito de mostrar cómo se pueden aplicar algunas de estas prácticas sugeridas.  
  
**Notas**:

* Al momento de redactar este post Angular **2** se encuentra en fase Beta, por lo cual se usará la versión **1.5**
* Se asumirá un conocimiento básico de AngularJS (qué son y cómo se usan los controladores, directivas, fábricas/servicios y expresiones), por lo que se recomienda para quienes no conozcan este framework se tomen el tiempo de revisar algunos tutoriales introductorios, por ejemplo:
  + Tutorial Oficial (en Inglés): <https://docs.angularjs.org/tutorial>
  + Serie de Videos (en

IoinicJS:

El desarrollo de aplicaciones híbridas ha aumentado considerablemente en los últimos años. Sin embargo, no todos los frameworks de desarrollo son igual de versátiles y hay veces en las que hay que decidirse por uno a la hora de desarrollar nuestra aplicación híbrida.

Si estás interesado en empezar a desarrollar aplicaciones híbridas, este artículo es para ti. En este post, desde el Mobility Lab de [BEEVA](https://www.beeva.com/) os voy a presentar los dos frameworks para desarrollo híbrido más famosos: **Ionic** y **React Native**.

Antes de comenzar, ¿qué diferencias hay entre una aplicación nativa y una aplicación híbrida?

Las aplicaciones nativas:

* Son desarrolladas en el **lenguaje requerido** por la plataforma para la que se está desarrollando (Java para Android, Objective-C para iOS, etc.).
* Presentan un **comportamiento distinto** en función de la plataforma (navegación, animaciones, etc.).
* **No tienen restricciones** para acceder a las capas nativas de los dispositivos (cámara, GPS, etc.).

Por otro lado, las aplicaciones híbridas:

* Son esencialmente páginas **web incrustadas** en una aplicación móvil a través de un WebView.
* Están desarrolladas usando HTML5, CSS y JavaScript.
* Pueden ejecutar el mismo código independientemente de cuál sea la plataforma objetivo, aunque también cabe la posibilidad de usar código específico para cada plataforma objetivo.
* Pueden acceder a alguna capa nativa de los dispositivos (cámara, GPS, etc.) usando plugins de [PhoneGap](http://phonegap.com/) y [Cordova](https://cordova.apache.org/).

Una vez presentado el concepto de aplicación híbrida, podemos hablar de Ionic y React Native.

Android:

Los celulares o tablets con Android, el sistema operativo de Google, pueden experimentar cambios en su rendimiento con el paso del tiempo. Esto puede estar relacionado a problemas internos que poco a poco retrasa la el funcionamiento diario hasta el padecimiento.

Para evitar que tu dispositivo te colme la paciencia, seguí paso a paso estas seis recomendaciones.

**Limpiar la memoria Cache**

Tus aplicaciones almacenan constantemente pequeñas porciones de datos, que normalmente aceleraría el rendimiento del teléfono. Pero si te estás quedando sin espacio, los datos almacenados en caché van a doler más de lo que ayudan. Para limpiar la memoria cache simplemente tenés que dirigirte a configuración, almacenamiento y hacer click en memoria cache. El teléfono preguntará si querés borrarla y presionas si.

Mirá también

[Celulares: el boom de los soft que le agregan un truco especial a las fotos](https://www.clarin.com/next/celulares-boom-soft-agregan-truco-especial-fotos_0_rynSfPdzM.html" \t "_blank)

**Deshabilitá o acelerá animaciones**

Las animaciones hacen que todas las transiciones e interacciones del sistema operativo sean más fluidas, hasta que el teléfono empieza a desacelerarse y esas animaciones empiezan a demorar todo. Si las animaciones ya no fluyen tan bien o tardan en abrir las aplicaciones, acelerarlas o apagarlas completamente mejorará la experiencia en general al liberar un poco de poder de procesamiento.

Para hacerlo tenés que ir a *configuración*, tocar la última opción que es Acerca del teléfono, presionar rápidamente varias veces el *Numero de compilación o Build Number* hasta que aparezca el mensaje en el que se activaron las opciones de desarrollador. Volvemos a configuración y ahora la anteúltima opción es "*opciones de desarrollador*", ahí bajamos hasta las opciones de Escala de Animación, Transición y Duración de animación. Ahí configuramos la opción que prefieras. 0.5 en todas para acelerar todas las animaciones o desactivar todas para eliminarlas completamente.

IOS:

### **1. Líquidos: no mojes tus dispositivos**

**Los fluidos son el peor contrincante para cualquier aparato electrónico**, incluyendo nuestros smartphones y tablets. Procura mantener tu smartphone a salvo de agua, lluvia o cualquier otro fluido. Para lograr ésto una funda adecuada para tu móvil podría ser una buena opción. Recuerda: ¡tu móvil no necesita ducharse!

### **2. No sitúes varios dispositivos juntos**

**Ya sean dos smartphones o un smartphone y una tablet**, si uno de tus dispositivos se sobrecalienta, transmitirá fácilmente el calor de uno a otro. No los dejes demasiado cerca uno de otro, nuestros móviles generan cierta radiación por calor simplemente con estar encendidos.

### **3. Mantén tu dispositivo alejado de cualquier fuente de calor**

**No dejes que un sol directo “bañe” tu dispositivo, sobre todo ahora** que comienza a llegar el buen tiempo y las temperaturas comienzan a subir. Mantenlo lejos de las fuentes de calor, en casos extremos puedes retirar la funda de tu dispositivo, ya que ésta retiene y concentra el calor (si está fabricada a partir de plásticos TPU, el problema podría ser mayor).

## Atento al uso que haces de tu móvil

### **4. Evita largos períodos de uso**

**El uso de algunas aplicaciones móviles o apps, en concreto los juegos** que necesitan explotar al máximo el chip gráfico de tu aparato, o [utilizan el GPS (como Pokémon GO](https://voltaico.lavozdegalicia.es/2017/02/como-solucionar-pokemon-go-gps-android/)) subirán la temperatura de tu terminal indudablemente. Deja descansar tu móvil, tu vista también te lo agradecerá.

FIREBASE:

**Firebase Invites: Recomendaciones**

* [**Contenido**](https://firebase.google.com/docs/invites/best-practices?hl=es-419#top_of_page)
* [Haz que las opciones de invitar y compartir sean fáciles de encontrar](https://firebase.google.com/docs/invites/best-practices?hl=es-419#haz-que-las-opciones-de-invitar-y-compartir-sean-fciles-de-encontrar)
* [Compila una hoja para compartir personalizada](https://firebase.google.com/docs/invites/best-practices?hl=es-419#compila-una-hoja-para-compartir-personalizada)
* [Personaliza tus invitaciones](https://firebase.google.com/docs/invites/best-practices?hl=es-419#personaliza-tus-invitaciones)
* [Ofrece incentivos](https://firebase.google.com/docs/invites/best-practices?hl=es-419#ofrece-incentivos)

Para incrementar las recomendaciones de boca en boca de tu app, sigue estas prácticas recomendadas cuando implementes Firebase Invites.

**Haz que las opciones de invitar y compartir sean fáciles de encontrar**

Puedes mantener las opciones de invitar y de compartir en la parte superior de tu app para que los usuarios siempre las tengan presentes.

* **Ubica** la opción de invitar en un lugar prominente, como el menú de nivel superior. Además, asegúrate de que el significado de los elementos del menú de invitación sea claro:
  + **Recomendado**: "Invitar amigos"
  + **Recomendado**: "Recomendar y ganar dinero"
* **Ubica** la opción de compartir en un lugar prominente, como a lo largo del contenido para compartir en tu app. Permite que los usuarios compartan la app y el contenido específico de la app.
* **No** las escondas en la app. Por ejemplo, no ubiques la opción de invitar en "Acerca de la app".
* Compila una hoja para compartir personalizada

La hoja para compartir predeterminada de Android muestra todos los canales de entrega posibles (que son demasiados), ya sea ordenados de manera alfabética o por la cantidad de tiempo de uso de la app. Ninguna de estas opciones es ideal. En su lugar, crea una hoja para compartir personalizada que incluya, en primer lugar, "SMS y correo electrónico" (la opción de Firebase Invites) y luego otros canales importantes, como Facebook y Twitter.

Si la hoja para compartir es clara y sencilla, aumentan las probabilidades de que los usuarios compartan contenido con sus contactos. Además, debido a que Firebase Invites combina el correo electrónico y los SMS en un canal, los usuarios pueden compartir contenido con todos sus amigos, ya sean contactos de correo electrónico o de dispositivos.

Cuando crees la hoja para compartir, redacta el canal para compartir de forma clara.

* **Recomendado**: "SMS y correo electrónico".
* **Recomendado**: "Compartir por SMS y por correo electrónico".

Infografías:

