

# El desarrollo de aplicaciones móviles nativas, Web o híbridas



## Índice

- 2 Introducción
- 2 Presentación de los enfoques
  - 2 Aplicaciones nativas
  - 3 La interfaz de programación de aplicaciones (API)
  - 4 Aplicaciones móviles basadas en la Web
  - 6 Aplicaciones híbridas
  - 7 Comparación de los distintos enfoques
- 8 Cómo elegir el enfoque correcto
  - 8 Escenarios para el enfoque nativo
  - 9 Escenarios para el enfoque Web
  - 9 Escenarios para el enfoque híbrido
- 10 Resumen

## Introducción

Muchas organizaciones que comienzan a implementar una estrategia móvil se enfrentan con una importante decisión que va a influir sobre los resultados de esta iniciativa. El proceso de elegir un enfoque de desarrollo para una aplicación móvil, ya sea nativa, Web o híbrida, implica muchos parámetros, como presupuesto, plazos del proyecto, destinatarios y funcionalidad de la

aplicación, entre otros. Cada enfoque conlleva beneficios y limitaciones inherentes, y encontrar la más adecuada para las necesidades de la organización puede ser una tarea difícil.

La finalidad de este documento no consiste en identificar el mejor enfoque de desarrollo, ya que no existe ninguno, sino enumerar los pros y los contras de cada uno y describir los distintos escenarios, o requisitos empresariales, que mejor se adecuan a cada uno de ellos.

## Presentación de los enfoques

### Aplicaciones nativas

Las aplicaciones nativas tienen archivos ejecutables binarios que se descargan directamente al dispositivo y se almacenan localmente. El proceso de instalación lo puede iniciar el usuario o, en algunos casos, el departamento de TI de la empresa. La manera más común de descargar una aplicación nativa es visitando una tienda de aplicaciones, como App Store de Apple, Marketplace de Android o App World de BlackBerry, pero existen otros métodos que a veces ofrece el proveedor móvil. Una vez que la aplicación ha sido instalada en el dispositivo, el usuario la ejecuta como cualquier otro servicio del dispositivo. Tras la inicialización, la aplicación nativa se conecta directamente con el sistema operativo móvil, sin ningún intermediario ni contenedor. La aplicación nativa puede acceder libremente a todas las APIs que el proveedor del SO ponga a disposición y, en muchos casos, tiene características y funciones únicas que son típicas de ese SO móvil en particular.

Para crear una aplicación nativa, los desarrolladores deben escribir el código fuente (en formato legible para los humanos) y crear recursos adicionales, como imágenes, segmentos de audio y diversos archivos de declaración específicos del SO. Utilizando herramientas provistas por el distribuidor del SO, se compila el código fuente (y a veces también se enlaza) para crear un ejecutable en formato binario que se pueda empaquetar junto con el resto de los recursos y estar listo para la distribución.

Estas herramientas, además de otros archivos y suministros, normalmente constituyen lo que se denomina el SDK (software development kit) del SO móvil.

Si bien el proceso de desarrollo suele ser similar para diferentes sistemas operativos, el SDK es específico de la plataforma, y cada SO móvil viene con sus propias herramientas. La siguiente tabla

representa las distintas herramientas, lenguajes, formatos y canales de distribución vinculados con los principales sistemas operativos móviles.

Estas diferencias entre plataformas ocasionan una de las desventajas más críticas del enfoque de desarrollo nativo: el código escrito para una plataforma móvil no se puede usar en otra, por lo cual el desarrollo y el mantenimiento de aplicaciones nativas para múltiples SO se convierte en una tarea muy ardua y costosa.

Entonces, ¿por qué pese a esta costosa desventaja muchas empresas prefieren el desarrollo nativo? Para responder esta pregunta, tendremos que comprender mejor el rol de las APIs.

### Interfaz de programación de aplicaciones (API)

Una vez que la aplicación nativa está instalada en el dispositivo móvil y es ejecutada por el usuario, interactúa con el sistema operativo móvil a través de llamadas API propietarias de las que dispone el sistema operativo. Estas se pueden dividir en dos grupos: APIs de bajo nivel y APIs de alto nivel.

	Apple iOS	Android	Blackberry OS	Windows Phone
Lenguajes	Objective-C, C, C++	Java (algunos C, C++)	Java	C#, VB.NET, etc.
Herramientas	Xcode	Android SDK	BB Java Eclipse Plug-in	Visual Studio, Windows Phone
Formato	.app	.apk	.cod	.xap
Tiendas	Apple App Store	Google Play	Blackberry App World	Windows Phone Marketplace

### APIs de bajo nivel

Es a través de las llamadas API de bajo nivel que la aplicación puede interactuar directamente con la pantalla táctil o el teclado, y así mostrar gráficos, conectarse a redes, procesar audio recibido por el micrófono, reproducir sonidos por el altavoz o auriculares, o recibir imágenes y videos de la cámara. Puede acceder al GPS, recibir información sobre orientación y, por supuesto, leer y escribir archivos en el disco en estado sólido o acceder a cualquier otro elemento de hardware disponible en la actualidad o en el futuro.

### APIs de alto nivel

Además de proporcionar los servicios de bajo nivel para acceder al hardware que acabamos de mencionar, los sistemas operativos móviles ofrecen servicios de alto nivel que son importantes para la experiencia móvil del usuario. Esos servicios incluyen procesos tales como navegar por Internet, gestionar el calendario, los contactos, álbumes de fotos y, por supuesto, la capacidad de hacer llamadas telefónicas o enviar y recibir mensajes de texto.

Aunque la mayoría de los SOs móviles incluyen un conjunto de aplicaciones incorporadas que pueden ejecutar esos servicios, existe un conjunto de APIs de alto nivel expuesto accesible para aplicaciones nativas también, lo que les permite acceder a muchos de los servicios importantes que acabamos de mencionar. Otras APIs permiten que las aplicaciones descargables accedan a diversos servicios en la nube ofrecidos por el distribuidor del SO, tales como notificaciones push o compras en tiendas de aplicaciones.

### Herramientas GUI (interfaz gráfica del usuario)

Otro conjunto importante de APIs que provee el SO es el toolkit GUI. Cada SO móvil viene con su propio conjunto de componentes para la interfaz del usuario, como botones, campos de introducción de datos, sliders, menús, barras de pestañas, cuadros de diálogo, etc. Las aplicaciones que utilizan estos componentes heredan las características y funciones de ese SO móvil específico, lo que en general constituye una experiencia muy fácil y divertida para el usuario.

Cabe aclarar que las diferentes plataformas móviles contienen una gama única de componentes de la IU (interfaz del usuario). Como resultado, las aplicaciones que están diseñadas para funcionar para múltiples sistemas operativos requieren que el diseñador conozca a la perfección los diferentes componentes IU de cada SO.

Si bien las APIs son específicas del SO y agregan bastante complejidad y costos para el desarrollo de múltiples aplicaciones nativas, estos elementos son la única manera de crear aplicaciones móviles completas que utilicen todas las funcionalidades que los dispositivos móviles modernos tienen para ofrecer.

### Aplicaciones móviles basadas en la Web

Los dispositivos móviles modernos cuentan con poderosos navegadores que dan soporte a muchas funcionalidades nuevas de HTML5, Cascading Style Sheets 3 (CSS3) y JavaScript de avanzada. Con los últimos avances logrados, HTML5 marca la transición de esta tecnología desde un “lenguaje de definición de páginas” a un poderoso estándar de desarrollo de aplicaciones complejas basadas en navegador.

Algunos ejemplos del potencial de HTML5 son: componentes IU avanzados, acceso a múltiples tipos de medios, servicios de geoposicionamiento y disponibilidad offline. Al emplear estas características y muchas otras que aún se encuentran en desarrollo, los desarrolladores pueden crear aplicaciones avanzadas, usando únicamente tecnologías basadas en la Web.

Es útil distinguir entre dos enfoques radicalmente distintos de aplicaciones Web. Todos estamos familiarizados con la navegación móvil y los sitios Web optimizados para servicios móviles. Estos sitios reconocen cuando se accede a ellos a través de un smartphone y preparan páginas HTML que han sido diseñadas para brindar una “experiencia táctil” cómoda en una pantalla pequeña. Pero algunas empresas llegan aún más lejos y mejoran la experiencia del usuario creando un sitio Web móvil que se parece a una aplicación nativa y se puede ejecutar a partir de un acceso directo que es idéntico al que se utiliza para lanzar aplicaciones nativas.

Existe un amplio abanico de posibilidades entre esos dos extremos, ya que la mayoría de los sitios Web implementan su propio mix de características. Las aplicaciones móviles basadas en la Web constituyen una tendencia muy promisoria. Para sacar el mayor provecho de esta tendencia y ayudar a los desarrolladores a construir la UI para el cliente, se han creado cada vez más herramientas JavaScript, tales como dojox.mobile, Sencha Touch y jQuery Mobile, las cuales generan interfaces de usuario con una apariencia similar a las aplicaciones analíticas. Ambas se ejecutan completamente dentro del navegador del dispositivo móvil y utilizan las últimas características de JavaScript, CSS y HTML5 disponibles en los navegadores móviles modernos.

Una de las principales ventajas de una aplicación Web es su soporte para múltiples plataformas y el bajo costo de desarrollo. La mayoría de los proveedores móviles utilizan el mismo motor de búsqueda en sus navegadores, llamado WebKit, que es un proyecto de fuente abierta conducido principalmente por Google y Apple y que ofrece la más completa implementación de HTML5 disponible en la actualidad.

Característica	Apl. Web solo móviles	Sitios Web solo móviles
Herramientas y conocimientos	Escritas totalmente en HTML, CSS y JavaScript	Escritas totalmente en HTML, CSS y JavaScript
Ejecución	Acceso directo “Instalado”, lanzado mediante apl. nativa	Navegando por un sitio mediante URL (Uniform Resource Locator)
Experiencia del usuario	Touch-friendly, interactive UI	IU mediante navegación entre páginas que muestran datos estáticos
Desempeño	IU reside localmente: aplicación con capacidad de respuesta y acceso offline	Todo el código se ejecuta desde un servidor: el rendimiento depende de la red

Pese al potencial y la promesa de las tecnologías Web en el espacio móvil, estas aún conllevan importantes limitaciones. Para comprender estas limitaciones es necesario explicar cómo funcionan las aplicaciones Web.

A diferencia de las aplicaciones nativas, que son ejecutables independientes que se conectan directamente con el SO, las aplicaciones Web se ejecutan dentro del navegador.

El navegador en sí es una aplicación nativa que tiene acceso directo a las APIs del SO, pero muy pocas de esas APIs están expuestas a las aplicaciones Web que se ejecutan dentro del mismo. Mientras que las aplicaciones nativas tienen acceso completo al dispositivo, para las aplicaciones Web muchas funcionalidades no están disponibles o sólo en forma parcial. Si bien gracias a los avances de HTML esto seguramente cambie en el futuro, estas funcionalidades no están disponibles actualmente para los usuarios móviles.

### Aplicaciones híbridas

El enfoque híbrido combina desarrollo nativo con tecnología Web. Usando este enfoque, los desarrolladores escriben gran parte de su aplicación en tecnologías Web para múltiples plataformas, y mantienen el acceso directo a APIs nativas cuando lo necesitan.

La porción nativa de la aplicación emplea APIs de sistemas operativos para crear un motor de búsqueda HTML incorporado que funcione como un puente entre el navegador y las APIs del dispositivo.

Este puente permite que la aplicación híbrida aproveche todas las características que ofrecen los dispositivos modernos.

Los desarrolladores de aplicaciones pueden optar por codificar su propio puente o bien aprovechar soluciones ya construidas, como PhoneGap, una biblioteca de fuente abierta que provee una interfaz JavaScript uniforme para funcionalidades de dispositivos seleccionados que son iguales en todos los sistemas operativos.

La porción nativa de la aplicación se puede desarrollar independientemente, pero algunas soluciones del mercado ofrecen este tipo de contenedor nativo como parte de su producto, lo que brinda al desarrollador formas de crear una aplicación avanzada que utilice todas las funciones del dispositivo usando únicamente lenguajes Web.

En algunos casos, una solución va a permitir que el desarrollador utilice cualquier conocimiento nativo que pueda tener para adaptar el contenedor nativo a las necesidades únicas de la organización.

La porción Web de la aplicación puede ser una página Web que resida en un servidor o bien un conjunto de archivos HTML, JavaScript, CSS y medios, incorporados en el código de la aplicación y almacenados localmente en el dispositivo. Ambos enfoques presentan ventajas y desventajas.

El código HTML que está alojado en un servidor permite que los desarrolladores introduzcan pequeñas actualizaciones en la aplicación sin tener que seguir el proceso de entrega y aprobación que algunas tiendas de aplicaciones requieren.

Pero este enfoque elimina cualquier disponibilidad offline, ya que el contenido no es accesible cuando el dispositivo no está conectado a la red. Por otro lado, incorporar el código Web en la aplicación misma puede mejorar el desempeño y la accesibilidad, pero no acepta actualizaciones remotas. Lo mejor de ambos mundos se puede lograr combinando los dos enfoques. Ese sistema está diseñado para alojar los recursos HTML en un servidor Web mejorando la flexibilidad, y al mismo tiempo tenerlos localmente en el dispositivo móvil para mejorar el desempeño.

### Comparación de los distintos enfoques

A modo de resumen, a continuación se comparan los tres enfoques de desarrollo.

El enfoque nativo se destaca por su desempeño y acceso de los dispositivos, pero conlleva costos y requiere actualizaciones. El enfoque Web es mucho más simple, menos costoso y más fácil

de actualizar, pero actualmente su funcionalidad es limitada y no puede alcanzar un alto nivel de experiencia del usuario como el de las llamadas API nativas.

El enfoque híbrido ofrece un término medio que, en muchas situaciones, constituye lo mejor de ambos mundos, en especial si el desarrollador desea emplearlo en múltiples sistemas operativos.

Como se puede observar en la tabla de arriba, ninguno de los enfoques en sí mismo ofrece todos los beneficios todo el tiempo. Para elegir el enfoque más adecuado hay que tener en cuenta las necesidades específicas de la organización, y basarse en muchos parámetros, como presupuesto, plazos de entrega, recursos internos, mercado objetivo, funcionalidad requerida de la aplicación, infraestructura de TI, etc.

Hay algo que es muy claro: La mayoría de las empresas actuales tienen que encontrar un punto medio, por un lado, entre la experiencia del usuario y la funcionalidad de las aplicaciones y, por el otro, entre los costos de desarrollo y el tiempo de salida al mercado. El desafío consiste en elegir el enfoque de desarrollo correcto que logre un equilibrio entre los requisitos de la organización con sus limitaciones vinculadas al presupuesto y al tiempo de salida al mercado.



Característica	Aplicación nativa	Aplicación híbrida	Aplicación Web
Lenguaje de desarrollo	Solo nativo	Nativo y Web o solo nativo	Solo Web
Portabilidad y optimización de código	Bajo	Alto	Alto
Características de acceso específicas del dispositivo	Alto	Mediano	Bajo
Uso de conocimiento existente	Bajo	Alto	Alto
Gráficos avanzados	Alto	Mediano	Mediano
Flexibilidad de actualizaciones	Bajo (Siempre Tiendas)	Mediano (Con frecuencia Tiendas)	Alto
Experiencia de instalación	Alta (A partir de la tienda)	Alta (A partir de la tienda)	Mediana (Mediante navegador móvil)

## Cómo elegir el enfoque correcto

A continuación se enumeran escenarios para ayudar a las organizaciones a elegir un enfoque.

### Escenarios para el enfoque nativo

**Aptitudes nativas existentes.** Uno de los principales argumentos en contra del enfoque nativo es su falta de soporte para múltiples plataformas. Las organizaciones que necesitan desarrollar una aplicación para muchas plataformas móviles deben contratar nuevos empleados o enseñarles a los desarrolladores internos una variedad de lenguajes nativos. Las organizaciones que sí cuentan con esas habilidades nativas en su propia empresa pueden aprovecharlas sin grandes inversiones extra.

**Un único SO móvil.** En algunos casos, una organización va a querer lanzar una aplicación móvil para una audiencia limitada de destinatarios, que utilice un único SO móvil. Por ejemplo, piense en un escenario en el cual una aplicación interna se distribuya dentro de una organización que da en préstamo un dispositivo BlackBerry a sus empleados. En este caso, puede que lograr una cobertura para múltiples plataformas no sea una

prioridad y, como para desarrollar una única aplicación nativa se necesitan pocas habilidades y herramientas, este enfoque puede resultar ideal.

**Funcionalidad nativa.** Algunas aplicaciones se construyen en torno a una única funcionalidad. Por ejemplo, Skype: el protocolo VoIP (voz sobre IP) y el acceso a los contactos del usuario son elementos clave de la aplicación y, por las tecnologías disponibles actualmente, sólo se pueden desarrollar de manera nativa. Para esas aplicaciones, los lenguajes de Internet aún no están lo suficientemente evolucionados y no son capaces de lograr la funcionalidad deseada.

**Requisitos de una IU sofisticada.** Para aplicaciones de juegos que requieren una IU con muchas funciones y capacidad de respuesta en tiempo real, las tecnologías Web aún no ofrecen una solución adecuada. Para ese tipo de aplicaciones, a los desarrolladores aún les conviene emplear el enfoque nativo.

### Escenarios para el enfoque Web

**Distribución directa.** Algunas organizaciones prefieren distribuir sus aplicaciones de manera que estén controladas internamente y que no estén sujetas a lo que a veces se puede convertir en un proceso de aprobación largo e incierto. En esos casos, usar únicamente lenguajes Web puede evitar el proceso de la tienda de aplicaciones y permitir que la organización controle totalmente la distribución de la aplicación y sus actualizaciones periódicas.

**Aplicación piloto.** Al comparar los costos y el tiempo de lanzamiento del desarrollo de una aplicación nativa con respecto a una Web, el enfoque Web puede crear una versión piloto de la aplicación que se convierta en una táctica atractiva y poco costosa. Una vez comprobado el concepto, la organización puede optar por crear una nueva aplicación desde cero o bien usar porciones del código existente para una aplicación híbrida.

**Visibilidad.** Además de la distribución que acabamos de mencionar, otro beneficio de crear una aplicación Web es su visibilidad en los resultados de los buscadores, cosa que, en muchos casos, exponen la aplicación a una audiencia más grande de la que está disponible a través de la tienda de aplicaciones.

### Escenarios para el enfoque híbrido

**Lograr un término medio.** Al emplear el enfoque híbrido, las empresas pueden acceder a lo mejor de los dos mundos. Por un lado, el puente nativo permite que los desarrolladores aprovechen al máximo todas las distintas características y funcionalidades de los dispositivos móviles modernos. Por el otro, todas las porciones del código que se escriben usando lenguajes Web se pueden compartir entre distintas plataformas móviles, por lo cual el proceso de desarrollo y de mantenimiento continuo se hace centralizado, más corto y menos costoso.

**Habilidades dentro de la empresa.** Las habilidades necesarias para el desarrollo Web son muy comunes y fáciles de encontrar en muchas organizaciones. Al elegir el enfoque híbrido, con la solución correcta, los desarrolladores Web son capaces de construir aplicaciones únicamente con habilidades Web, como HTML, CSS y JavaScript, además de ofrecer al usuario una experiencia similar a la nativa.

**Expectativas futuras.** HTML5 está aumentando rápidamente su disponibilidad y sus funcionalidades. Muchos analistas predicen que muy probablemente reconvierta en la tecnología por defecto para el desarrollo de aplicaciones para el cliente. Al escribir la mayor parte de la aplicación en HTML, y usar código nativo solo donde se necesita, las empresas pueden asegurarse de que las inversiones que hagan hoy no van a ser obsoletas el día de mañana, ya que la funcionalidad HTML es cada vez más sofisticada y cumple con un número cada vez mayor de requisitos móviles de las organizaciones modernas.

## Resumen

Las aplicaciones móviles siguen desempeñando un rol central en el entorno de negocio, es por ello que las organizaciones de todo el mundo están aumentando sus servicios de misión crítica.

Muchas empresas están procurando encontrar el mejor enfoque de desarrollo para lograr sus metas, pero muchas se están dando cuenta rápidamente de que cada enfoque conlleva limitaciones inherentes, y de que ningún enfoque en sí mismo puede abordar las crecientes necesidades y complejidades de la empresa móvil moderna.

Como explicamos en este documento, la respuesta no es un único enfoque de desarrollo sino una solución flexible que pueda aprovechar los beneficios de cada uno para dar soporte no sólo al desarrollo de una primera aplicación móvil sino a todas las aplicaciones futuras, cualquiera sea el enfoque empleado.

Elegir entre enfoques de desarrollo híbridos, nativos y Web, si bien es una importante decisión, no es la única que las empresas deben tomar.

Las empresas que estén creando su estrategia móvil también deben tener en cuenta el futuro de este mercado, representado por las siguientes tendencias y desarrollos:

- Una mayor fragmentación de dispositivos móviles y tecnologías, lo que, a su vez, va a seguir aumentando los costos generales y las complejidades que conlleva el desarrollo, la integración y la gestión de las aplicaciones móviles.

- Una adopción móvil acelerada por parte de los consumidores y dentro de la empresa, lo que incrementa los requisitos en materia de seguridad, escalabilidad y control.
- Nuevas características de los dispositivos y tecnologías complementarias, tales como la comunicación NFC (en áreas cercanas), la geolocalización, la realidad aumentada, las redes sociales, etc., que sin lugar a dudas van a generar nuevos tipos y situaciones de uso de aplicaciones móviles.
- Nuevos canales de distribución para las aplicaciones, tanto públicos como privados, que permitirán a las organizaciones colocar fácilmente las aplicaciones en manos del usuario, implementar actualizaciones rápidamente y gestionar su portafolio completo de aplicaciones sin tener que pasar por un largo proceso de presentación y aprobación.

Teniendo en cuenta todos estos parámetros, las empresas deben elegir una solución que no sólo sea lo suficientemente flexible para dar soporte a todos los tipos de aplicaciones sino que también sustente una integración segura y escalable de las aplicaciones dentro de la infraestructura de TI, y les permita supervisar y controlar su portafolio completo de aplicaciones a partir de una única interfaz centralizada.

Si desea más información, o descargar una versión de prueba gratuita, visite el sitio:

[ibm.com/software/mobile-solutions/worklight/](http://ibm.com/software/mobile-solutions/worklight/)



## Para más información

Si desea más información acerca de los activos IBM Worklight para el desarrollo de aplicaciones móviles, póngase en contacto con su representante o socio de IBM, o visite el siguiente sitio: [ibm.com/software/solutions/mobile-enterprise](http://ibm.com/software/solutions/mobile-enterprise)

Además, IBM Global Financing puede ayudarle a adquirir las funcionalidades de software que su empresa necesita de la manera más económica y estratégica posible. Vamos a asociarnos con empresas crediticias calificadas para personalizar una solución de financiamiento adecuada a sus metas de negocio y desarrollo, que le permita gestionar eficazmente el efectivo y mejorar su costo total de propiedad. Financie sus inversiones críticas de TI e impulse su negocio con IBM Global Financing. Para más información, visite: [ibm.com/financing](http://ibm.com/financing)



---

© Copyright IBM Corporation 2012

IBM Corporation  
Software Group  
Route 100  
Somers, NY 10589

Elaborado en los Estados Unidos de América  
Abril de 2012

IBM, el logo de IBM e [ibm.com](http://ibm.com) y WebSphere son marcas comerciales de International Business Machines Corp., registradas en muchas jurisdicciones de todo el mundo. Los nombres de otras empresas, productos y servicios pueden ser marcas registradas de IBM o de terceros. Existe una lista actual de las marcas comerciales de IBM en “Copyright and trademark information” de [ibm.com/legal/copytrade.shtml](http://ibm.com/legal/copytrade.shtml)

Java y todas las marcas y logos basados en Java son marcas comerciales o marcas registradas de Oracle y/o sus subsidiarias.

Este documento es vigente a partir de la fecha inicial de publicación, y puede ser modificado por IBM en cualquier momento. No todos los productos y servicios están disponibles en todos los países en los cuales IBM está presente.

Es responsabilidad del usuario evaluar y verificar el funcionamiento de cualquier otro producto o programa con productos y programas IBM. LA INFORMACIÓN DE ESTE DOCUMENTO SE PRESENTA TAL Y COMO FUE CREADA, SIN NINGUNA GARANTÍA, EXPRESA O IMPLÍCITA, INCLUSO SIN NINGUNA GARANTÍA DE COMERCIABILIDAD, ADECUACIÓN PARA UN PROPÓSITO PARTICULAR NI NINGUNA GARANTÍA O CONDICIÓN DE NO VIOLACIÓN. Los productos de IBM están garantizados según los términos y condiciones de los contratos bajo los cuales fueron provistos.



Reciclar

---