# Capítulo 5 - Aplicaciones Móviles

## **Introducción**

Como veremos en este capítulo, las aplicaciones móviles se encuentran distribuidas en distintos sistemas operativos como Android, iOS, Windows, entre otros en menor medida. Estos aplicativos permiten interactuar en un mundo en los que los dispositivos móviles toman mayor relevancia en el mercado de Internet, por la alta demanda de usuarios. Entre la diversidad de aplicaciones se encuentran las nativas, las webs y las híbridas. Cada uno de éstas, cuentan con distintas metodologías de desarrollo, pero enfocadas a un mismo paradigma. Por un lado, HTML5 como tecnología emergente empuja al desarrollo de aplicaciones web y por otro, las apps nativas sobre Android y IOS (principalmente) reinan en rendimiento. La brecha entre las ventajas y desventajas entre las App Nativas y las webs, genera mayor interés sobre las App Híbridas, tomando ventajas sobre cada partido. Con el advenimiento de tecnologías cómo Cordova, IntelXDK, Ionic y la salida de las sombras del HTML5, la comunidad de desarrolladores comienza a apostar a estos framework que se apoyan en tecnologías de Front-End para el desarrollo de aplicaciones móviles. [[1]](#footnote-1)

## **5.1 Las Aplicaciones móviles**

“Una aplicación móvil o *App*es una aplicación informática diseñada para ser ejecutada en teléfonos inteligentes, tabletas y otros dispositivos móviles y que permite al usuario efectuar una tarea concreta de cualquier tipo: profesional, de ocio, educativas, de acceso a servicios, etc; Facilitando las gestiones o actividades a desarrollar”

Al ser aplicaciones residentes en los dispositivos están escritas están escritas mayormente en Java (Android), Objective-C (IOS) y Windows Phone (C#). Su funcionamiento y recursos se encaminan a aportar una serie de ventajas tales como:

Ilustración 30 - Aplicaciones móviles

* Un acceso más rápido y sencillo a la información necesaria sin necesidad de los datos de autenticación en cada acceso.
* Un almacenamiento de datos personales que, a priori, es de una manera segura. Esto quiere decir (Android) que utiliza un sistema de permisos heredado de Linux y conocido como la Sandbox. Este sistema limita en gran medida el acceso al sistema de archivos e impide que los procesos puedan acceder a los recursos de otros procesos, como la memoria y la CPU. Sin embargo, existen procesos con privilegios elevados que pueden acceder a recursos del dispositivo sin problemas, principalmente servicios del sistema y aplicaciones preinstaladas en los dispositivos.[[2]](#endnote-1) Los permisos son declarados en un archivo denominado AndroidManifest.xml . Estos permisos definen los recursos del dispositivo que van a utilizar y que el usuario debe aceptar antes de instalarlas, como el acceso a la cámara, leer o escribir en el almacenamiento externo, el acceso a internet, etc. Android impide que las apps utilicen recursos no manifestados, provocando un fallo en las mismas o su cierre, aunque esto solo sucede durante el desarrollo de las aplicaciones y no en su distribución.
* Una gran versatilidad en cuanto a su utilización o aplicación práctica. Dada la facilidad que presenta a la hora de la utilización del usuario, dado que necesita instalar la aplicación y mantenerla actualizada. No necesita de un navegador para poder acceder
* La atribución de funcionalidades específicas.Esto se debe a que las app nativas acceden a recursos hardware con mayor facilidad (cámara, contactos, memoria, notificaciones push, etc.)
* Mejorar la capacidad de conectividad y disponibilidad de servicios y productos (usuario-usuario, usuario-proveedor de servicios).

### 5.2.1 Las Web Apps

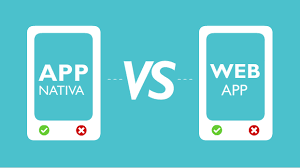
****Una Web App es una versión de una página web optimizada y adaptable a un grán número de dispositivos móviles independientemente del sistema operativo que utilice. Esta optimización es posible gracias al características provistas por lenguaje de marcado HTML5, combinado con hojas de estilo e cascada CSS3, que permiten proveer adaptabilidad, denominada en inglés “*Responsive Web Design*”. El diseño web responsivo es una filosofía de diseño y desarrollo donde el objetivo es adaptar la apariencia de las páginas webs al dispositivo(visualización) que se utiliza para visitarlas. Se caracteriza porque los layout y los contenidos multimediales son fluidos y se utiliza código media-queries de CSS3. [[3]](#footnote-2)

Ilustración 31 - App nativa vs Web App

### 5.2.2 Ventajas de las Web-App:

* No ocupa espacio de memoria de almacenamiento en los dispositivos (no es una App).
* No requiere actualizaciones ya que al ser una página web siempre se accede a la última versión.
* No consume recursos dado que no instala servicios en segundo plano y además no consume espacio dado que no es necesario instalar la aplicación para su uso.
* En líneas generales la implementación de una Web App es más económica que el de una App.

### 5.2.3 Desventajas de las Web-Apps

* No permite la promoción y distribución a través de los *markets* (Google Play, Nokia Store, App Store, Windows Phone Apps)
* Requiere de una conexión entre el cliente y el servidor (por ejemplo por internet o una WLAN).
* Menor usabilidad, al ofrecer un acceso muy limitado a los elementos y capacidades hardware del dispositivo. (geolocalización, notificaciones “push”).
* Carece de un icono de lanzamiento específico.
* Necesitan de un espacio web.
* No funcionan en segundo plano (multitarea)

Ilustración 32 – WebApps – Diseño multipropósito

## **5.3 Sistemas operativos para dispositivos móviles**

Un sistema operativo móvil o SO móvil es un conjunto de programas de bajo nivel que permite la abstracción de las peculiaridades del hardware específico del teléfono móvil y provee servicios a las aplicaciones móviles, que se ejecutan sobre él. Al igual que los PCs que utilizan Windows, Linux o Mac OS, los dispositivos móviles ejecutan sistemas operativos como Android, iOS o Windows Phone, entre otros.

A medida que los teléfonos móviles crecen en popularidad, los sistemas operativos con los que funcionan adquieren mayor importancia. La cuota de mercado de sistemas operativos móviles en el primer trimestre de 2016 era el siguiente:

* Android 84,1 %
* iOS 14,8 %
* Windows Phone 0,7 %
* BlackBerry OS 0,2 %
* Otros 0,2 %

Android tiene la mayor cuota, desde enero 2011, con más de la mitad del mercado, experimentó un creciente aumento y en solo dos años (2009 a comienzos de 2011) ha pasado a ser el SO móvil más utilizado.

Es por esto, que en principio se pensó desarrollar una App para operar el SAR para esta plataforma.

## **5.4 Android**

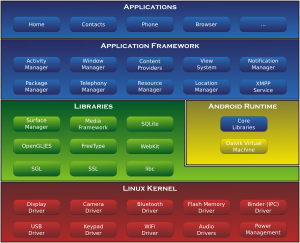
Es sin duda el líder del mercado móvil en sistemas operativos, Se encuentra basado en Linux, diseñado originalmente para cámaras fotográficas profesionales, luego fue vendido a Google y modificado para ser utilizado en dispositivos móviles como los teléfonos inteligentes y luego en tablets. Actualmente se encuentra en desarrollo para usarse en netbooks y PCs. Debido a la gran variedad de dispositivos que ejecutan Android, la *Open Handset Alliance*, compuesta por 84 compañías de hardware, software y telecomunicaciones, se dedicada al desarrollo de estándares abiertos para celulares, ayudado en gran medida a la masificación del SO de Google, hasta el punto de ser usado por empresas como HTC, LG, Samsung, Motorola entre otros.

Ilustración 33 - Arquitectura de Android



Las aplicaciones para Android se escriben y desarrollan en Java, aunque con unas APIs propias, por lo que las aplicaciones escritas en Java para PC y demás plataformas ya existentes no son compatibles con este sistema.

Ilustración 34 - Logo de Android

## **5.5 Aplicaciones móviles multiplataforma**

### 5.5.1 Diferencias entre aplicaciones y web móviles

Las aplicaciones nativas y comparten la pantalla del teléfono con las webs móviles conviven en mismo dispositivo. Mientras las primeras tienen que ser descargadas e instaladas antes de usar, a una web puede accederse simplemente usando Internet y un navegador compatible con la aplicación web. Pero, no todas pueden verse correctamente desde una pantalla generalmente más pequeña que la de un ordenador de escritorio.

Las que se adaptan especialmente a un dispositivo móvil como se mencionó anteriormente se denominan “web responsiva” (5.2.1 Las Web Apps) y son ejemplo del “diseño líquido” (adaptativo), debido a que el contenido toma la forma del contenedor. , mostrando la información según sea necesario.

Antes de la existencia del CSS3, como no había tecnología para poder crear sitios elásticos que se vieran bien en cualquier dimensión de pantalla, los desarrolladores de web estaban obligados a crear diferentes versiones de las páginas web. Gracias a las utilidades de CSS3 como las queries, las páginas pueden reaccionar ante distintas circunstancias como la anchura de la pantalla, ventana donde están aquellos elementos a maquetar. Web responsive se denomina a todas aquellas técnicas (no solamente redimensionado de pantalla) que permiten la adaptabilidad del contenido a los dispositivos terminales.

### 5.5.2 App Nativas

Una App nativa es aquella que se desarrolla de forma específica para un determinado sistema operativo, utilizando un *Software Development Kit*o SDK. Cada una de las plataformas, Android, iOS o Windows Phone, tienen un SDK diferente, por lo que si se desea que una App esté disponible en todas las plataformas se deberán de crear varias versiones, cada una con el lenguaje de la plataforma.

* Las apps para iOS se desarrollan con lenguaje Objective-C o Swift.
* Las apps para Android se desarrollan con lenguaje Java o Kotlin
* Las apps en Windows Phone se desarrollan en C# o lenguajes *managed* que se ejecuten sobre el CLR de .Net.

Cuando hablamos de desarrollo móvil casi siempre nos estamos refiriendo a Apps nativas. La principal ventaja con respecto a las app webs y app híbridas, los otros dos tipos, es la posibilidad de acceder a la totalidad de las características del hardware ofrecidas por el móvil, por ejemplo cámara, GPS, agenda de contactos, almacenamiento y la capacidad ser ejecutada sin necesidad de conectividad a internet. Además, las Apps son promocionadas por medio de las tiendas de aplicaciones, mientras que las Web Apps deben ser descargadas accedidas por el usuario a través de una URL. Las App nativas presentan un mayor rendimiento y aprovechamiento del hardware de cada smartphone o dispositivo móvil.

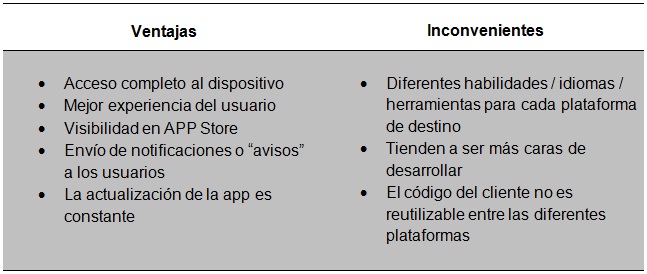


Ilustración 35 - Cuadro comparativo nativas

### 5.5.3 Desarrollo de Web Apps

Una aplicación web o Web App se desarrollada con los lenguajes HTML, Javascript y CSS que revisten de gran popularidad en la actualidad.

En el contexto de aplicaciones móviles, su principal ventaja con respecto un desarrollo nativo es la posibilidad de programar independientemente del SO en el que se usará la aplicación. De esta forma se pueden ejecutar en diferentes dispositivos sin tener que crear varias aplicaciones.

Las aplicaciones web se ejecutan dentro del propio navegador web del dispositivo a través de una URL.

La diferencia mayor con una aplicación nativa es que carece del proceso de instalación, pero con la contraparte de no poder estar visibles en la tienda de aplicaciones y por ende, la promoción y comercialización debe realizarse de forma independiente. Carecen también de la capacidad de accederse desde el lanzador del dispositivo, pero en algunas plataformas, este inconveniente puede suplirse con la creación de un acceso directo o link.

Las Web Apps móviles son una opción atractiva si el objetivo es adaptar la web a formato móvil.

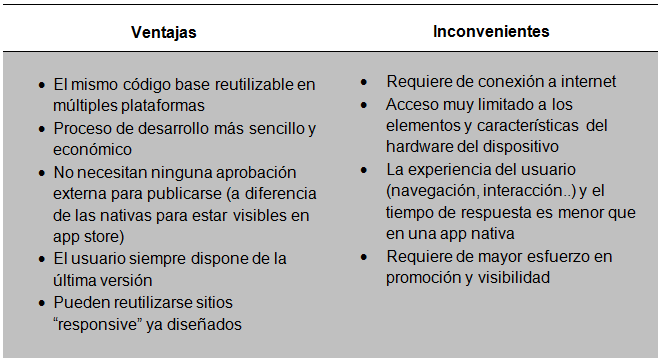


Ilustración 36 - Cuadro comparativo - Aplicaciones Web

### 5.5.4 Aplicaciones Híbridas

Una aplicación híbrida es una combinación de las dos anteriores. Las apps híbridas se desarrollan con lenguajes propios de las Web Apps, es decir, HTML, Javascript y CSS, pero también dan la posibilidad de acceder a gran parte de las características del hardware del dispositivo. La principal ventaja es que es posible empaquetarla y distribuirla en la tienda de aplicaciones de cada SO.

PhoneGap es uno de los frameworks más utilizados por los programadores para el desarrollo multiplataforma de aplicaciones híbridas. Otro ejemplo de herramienta para desarrollar apps híbridas es Cordova. Tanto PhoneGap como Apache Cordova, son los frameworks más utilizados por los programadores para el desarrollo multiplataforma de aplicaciones híbridas.[[4]](#footnote-3) También existe la opción de servirse de cross-platforms frameworks, para desarrollar una web app.

### 5.5.5 Aplicación híbrida: Web App

Cosiste en diseñar la aplicación como si fuera una Web App para ser ejecutada en el propio navegador del cliente. Sin embargo, ésa es precisamente su gran virtud, pero también su principal inconveniente. Nos encontramos con que el código es válido para todos los dispositivos, pero tanto la experiencia de usuario y como la apariencia no están a la par de una App nativa. Phonegap es el framework más popular en esta tecnología.

### 5.5.6 Aplicación híbrida: app interpretada

La aplicación interpretada significa que la aplicación es programada y luego cada terminal la traduce a su propio lenguaje de programación. Facilita el desarrollo de aplicaciones y reduce el esfuerzo considerablemente. Aunque el resultado no es idéntico a la nativa, la apariencia es bastante buena, y en muchas ocasiones puede ser la solución al problema del desarrollo de aplicaciones multiplataforma.

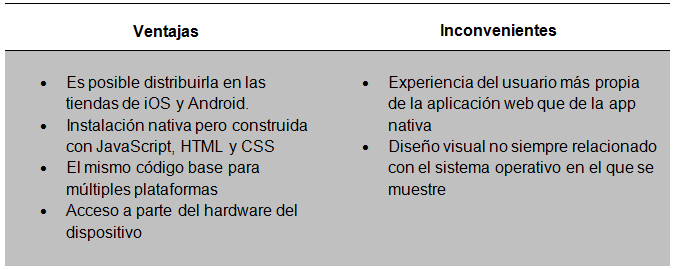


Ilustración 37 - Comparativa aplicaciones híbridas

## **5.6 Investigación en Desarrollo de Apps y WebApps**

Al momento de seleccionar el tipo de App o Web App se investigó en las distintas tecnologías para seleccionar la adecuada

Ilustración 38 - Herramientas para desarrollo de apps

### 5.6.1 Android Studio

Android Studio es el IDE oficial para el desarrollo de aplicaciones para Android, basado en la tecnología IntelliJ IDEA . Además del editor de códigos y las herramientas para desarrolladores de IntelliJ, Android Studio ofrece la posibilidad de instalarse en los distintos sistemas operativos como UN/Linux, MS Windows, MacOs. Además incluye características[[5]](#footnote-4) como:

* Integración de ProGuard(reducción de código, eliminación de atributos, clases, métodos sin utilizar)
* Firma de aplicaciones
* Renderizado en tiempo real
* Consola de desarrollador: consejos de optimización, ayuda para la traducción, estadísticas de uso
* Editor de diseño, con posibilidad de arrastrar y soltar elementos
* Herramientas Lint (detección de problemas de rendimiento, usabilidad, compatibilidad de versiones, etc)
* Soporte para crear diseños en Android Wear (Sistema operativo para dispositivos corporales de Android)
* Google Cloud Platform
* Dispositivos virtuales para simular las aplicaciones

funciones que aumentan la productividad durante la compilación de apps para Android. Provee también emuladores para diferentes plataformas de hardware, destinados a la prueba de Apps.

### 5.6.2 App Inventor

Es un entorno de desarrollo de software creado por Google Labs para la elaboración de aplicaciones destinadas al sistema operativo Android. El usuario puede, de forma visual y a partir de un conjunto de herramientas básicas, ir enlazando una serie de bloques para crear la aplicación. El sistema es gratuito y se puede descargar fácilmente de la web desde la web la aplicación generada. Las aplicaciones creadas con App Inventor están limitadas por su simplicidad, aunque permiten cubrir un gran número de necesidades básicas en un dispositivo móvil.

Con Google MIT App Inventor, se espera un incremento importante en el número de aplicaciones para Android debido a dos grandes factores: la simplicidad de uso, que facilitará la aparición de un gran número de nuevas aplicaciones; y Google Play, el centro de distribución de aplicaciones para Android donde cualquier usuario puede distribuir sus creaciones libremente.

Otra gran cualidad es la posibilidad de insertarlo en la educación dado su programación por medio de bloques gráficos, que resultan ser muy intuitivos en aquellas personas que se introducen.

Unas primeras apps para el SAR fueron realizadas bajo esta plataforma, de dicha experiencia se pudo concluir que tiene una baja curva de aprendizaje.

5.6.3 Tecnologías del lado del cliente - *Open Web Stack* (HTML, CSS y JS)

### 5.6.3.1 HTML

 Es un [lenguaje de marcado](https://es.wikipedia.org/wiki/Lenguaje_de_marcado" \o "Lenguaje de marcado) para la elaboración de [páginas web](https://es.wikipedia.org/wiki/P%C3%A1gina_web" \o "Página web). Es un estándar que sirve de referencia del software que conecta con la elaboración de páginas web en sus diferentes versiones, define una estructura básica y un código (denominado código HTML) para la definición de contenido de una página web, como texto, imágenes, videos, juegos, entre otros.

### 5.6.3.2 CSS

Como se mencionó anteriormente (Sección 5.5.1), CSS (Hoja de estilos en cascada) es un lenguaje de diseño gráfico para definir y crear la presentación de un documento estructurado escrito en un lenguaje de marcado.

### 5.6.3.3 JS

JavaScript es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos,​ basado en prototipos, imperativo, débilmente tipado y dinámico.

Se utiliza principalmente en su forma del lado del cliente (client-side), implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas, ​ aunque existe una forma de JavaScript del lado del servidor.

### 5.6.3.4 ¿Qué es SASS?

Es un metalenguaje de estilos en cascada. Es un lenguaje que preprocesado produce CSS, denominado SassScript. Existen dos formatos SCSS (Sintaxis con bloques) y SASS (Sintaxis con indentación). CSS3 consiste en una serie de selectores y pseudo-selectores que agrupan las reglas que son aplicadas.

SassScript proporciona los mecanismos necesarios para ofrecer variables, nesting (anidamientos), mixins, y herencia de los selectores.

Las variables Permite reutilizar valores que podemos manejar desde un solo sitio de forma sencilla y centralizada

Un mixin permite aprovechar un trozo de nuestro código al que podemos llamar repetidamente y que encapsula en una o más líneas de código.

La sintaxis de .sass y .scss no puede ser interpretada directamente por los navegadores, por ende es necesario la compilación.

### 5.6.3.5 Angular JS

Es un framework MVC (Model, View, Controller) de JavaScript para el Desarrollo Web Front End que permite crear aplicaciones SPA Single-Page Applications (una única página).

### 5.6.3.6 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.

* 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.
* Vista: Se trata del código que nos permitirá presentar los datos que el modelo nos proporciona.
* 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.

### 5.6.4 Cordova

Apache Cordova es un entorno de desarrollo de aplicaciones móviles, originalmente creado por Nitobi y comprado por Adobe [[6]](#footnote-5). Más tarde fue liberado como Apache Cordova. Permite, construir aplicaciones para dispositivos móviles utilizando CSS3, HTML5, y Javascript. Las aplicaciones resultantes son híbridas, lo que significa que no son ni una aplicación móvil nativa o App (porque toda la representación gráfica se realiza vía vistas Web en vez del framework de UI nativo) ni puramente basadas en web (porque no son solo aplicaciones web, sino que están empaquetadas como aplicaciones para su distribución y tienen acceso a las APIs nativas del dispositivo).

### 5.6.5 Intel XDK

Es un kit de desarrollo creado por Intel para crear aplicaciones nativas para los teléfonos celulares y las tabletas que utilizan tecnologías web como HTML5, CSS y JavaScript. Las aplicaciones se compilan mediante un servicio on-line. Hace uso de la plataforma Cordova para crear aplicaciones cross-platform, enfocado en el segmento de Apps para IoT. Posee un emulador, pervisualización de apliciones mediante el scan de un código QR, Drag and Drop y soporte de plantillas. Brinda soporte para Android, iOS, Windows Phone, entre otras plataformas.

### 5.6.6 Ionic

Es una Framework, Open Source y de distribución gratuita, para el desarrollo de aplicaciones híbridas, inicialmente pensado para móviles y tablets, basadas en HTML5, CSS y JS. Está construido con Sass y optimizado con AngularJS.

### 5.6.6 Meteor

Es una plataforma para crear aplicaciones web en tiempo real construida sobre Node.js. Meteor se localiza entre la base de datos de la aplicación y su interfaz de usuario y se encarga que las dos partes estén sincronizadas.

Meteor puede compartir código JavaScript entre el cliente y en el servidor.

### 5.6.7 Meteor y Cordova

Existe una integración del framework Meteor con Cordova, que permtite que una aplicación web crada con Meteor, sea ejecutarla en un dispositivo iOS o Android de forma híbrida, es decir, se ejecuta en una vista web incrustada en una aplicación nativa en lugar de en un navegador móvil del usuario. Un beneficio importante de empaquetar su aplicación web como una aplicación de Cordova es que todos sus recursos que erían descargados desde la web se incluyen con la aplicación, asegurando una velocidad de carga mayor, beneficiando a los usuarios con conexiones lentas. Otra característica es la compatibilidad con *hot code push*, que le permite actualizar la aplicación en los dispositivos de los usuarios sin pasar por el proceso habitual de revisión de la tienda de aplicaciones. Cordova también permite el acceso a ciertas características nativas a través de una arquitectura de complementos. Los complementos permiten utilizar funciones que normalmente no están disponibles para aplicaciones web, como acceder a la cámara del dispositivo o al sistema de archivos local, interactuar con lectores de código de barras o NFC.

Resumen

Como vimos en este capítulo, las aplicaciones móviles son aplicaciones informáticas diseñadas para ser ejecutadas en teléfonos inteligentes, tabletas y otros dispositivos móviles y que permite al usuario efectuar una tarea concreta de cualquier tipo.

Existen distintas categorías como las aplicación nativas, aplicaciónes web y aplicaciones híbridas. Además se observaron ventajas y desventajas sobre cada categoría, como por ejemplo las app nativas no necesitan conectividad a internet para funcionar frente a las app web. Las app hibridas poseen ventajas agregadas de las otras dos tipos. Por otro lado, se describieron cada una de las tecnologías que participan en el desarrollo de las Apps como Córdoba, meteor, Intel XDK y tecnologías web del lado del cliente.

.

1. <https://www.developereconomics.com/graphs/de11>

   <https://www.xatakamovil.com/mercado/desarrollo-de-aplicaciones-moviles-i-asi-esta-el-mercado> [↑](#footnote-ref-1)
2. https://source.android.com/security/overview/kernel-security [↑](#endnote-ref-1)
3. https://es.wikipedia.org/wiki/Dise%C3%B1o\_web\_adaptable [↑](#footnote-ref-2)
4. <https://es.wikipedia.org/wiki/PhoneGap>

   <https://es.wikipedia.org/wiki/Apache_Cordova> [↑](#footnote-ref-3)
5. https://es.wikipedia.org/wiki/Android\_Studio# [↑](#footnote-ref-4)
6. <https://es.wikipedia.org/wiki/Apache_Cordova>

   http://cordova.apache.org/ [↑](#footnote-ref-5)