

Entorns de programació mòbils

Julián David Morillo Pozo

PID_00176739



Universitat Oberta
de Catalunya

www.uoc.edu



Els textos i imatges publicats en aquesta obra estan subjectes –llevat que s'indiqui el contrari– a una llicència de Reconeixement-Compartir igual (BY-SA) v.3.0 Espanya de Creative Commons. Podeu modificar l'obra, reproduir-la, distribuir-la o comunicar-la públicament sempre que en citeu l'autor i la font (FUOC. Fundació per a la Universitat Oberta de Catalunya), i sempre que l'obra derivada quedi subjecta a la mateixa llicència que el material original. La llicència completa es pot consultar a <http://creativecommons.org/licenses/by-sa/3.0/es/legalcode.ca>

Índex

Introducció	5
Objectius	6
1. Història i evolució dels entorns de programació mòbils	7
2. Aplicacions web i aplicacions natives	11
3. Enumeració dels diferents entorns	13
3.1. Entorns per a dispositius de diferents venedors	13
3.1.1. Java ME	13
3.1.2. Symbian	14
3.1.3. Android	16
3.1.4. Windows Mobile	16
3.1.5. Qt <i>framework</i>	16
3.1.6. BREW	17
3.1.7. Palm OS	17
3.1.8. Flash Lite	17
3.1.9. Micronavegador	17
3.2. Desenvolupament multiplataforma	17
3.2.1. Titanium Mobile	18
3.2.2. PhoneGap	19
3.3. Entorns per a dispositius d'un venedor únic	20
4. Llenguatges de programació	21
4.1. Llenguatges de programació per al Windows Mobile	22
4.1.1. Visual C++	22
4.1.2. Visual C# i Visual Basic	23
4.1.3. JScript	23
4.1.4. ASP.NET	24
5. Exemples d'entorns	25
5.1. iPhone/iOS	25
5.1.1. Visió general del sistema iOS	25
5.1.2. Història del sistema iOS	26
5.1.3. Història de les versions del sistema iOS	27
5.1.4. Característiques del sistema iOS	28
5.1.5. Desenvolupament d'aplicacions per a l'iOS	30
5.1.6. <i>Jailbreaking</i>	30
5.1.7. Gestió de drets digitals	31
5.2. Android	31

5.2.1. Història de l'Android	32
5.2.2. Història de les versions de l'Android	34
Glossari	35
Bibliografia	36

Introducció

El desenvolupament d'aplicacions mòbils és el procés pel qual es desenvolupa un programari per a dispositius mòbils com telèfons intel·ligents (*smart phones*) o tauletes tàctils (*tablets*). La forma de distribució d'aquestes aplicacions pot variar: poden venir preinstal·lades als telèfons o bé les poden baixar els usuaris des de repositoris d'aplicacions (*app stores*) i altres plataformes de distribució de programari (*software*).

En aquest mòdul veurem els diferents entorns de programació per a aplicacions mòbils existents fent primer una revisió històrica de l'evolució que han tingut. Després farem una revisió dels diferents llenguatges de programació que es poden utilitzar dins d'aquests entorns. Finalment estudiarem a fons alguns dels entorns més populars.

Objectius

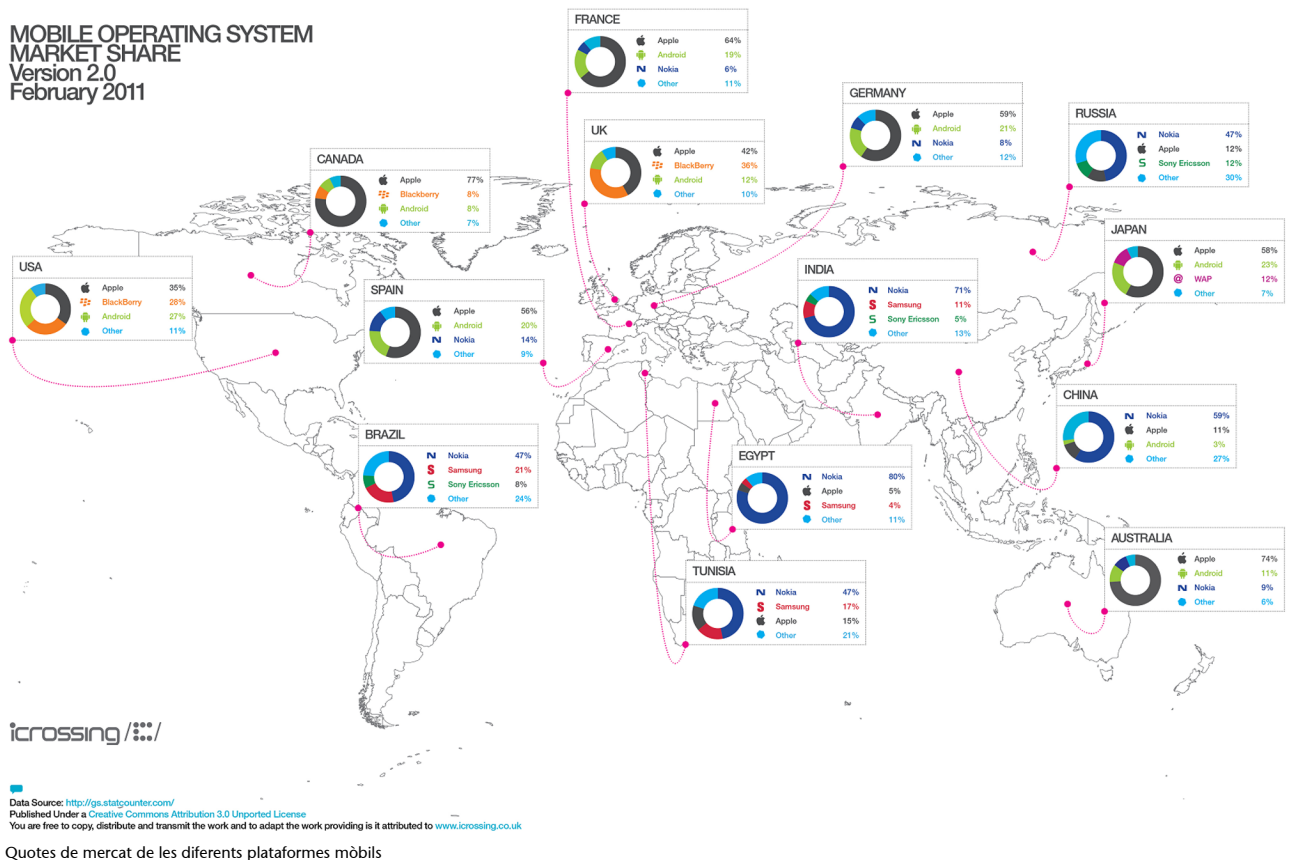
Amb l'estudi d'aquest mòdul es pretén que l'estudiant assoleixi els objectius següents:

- 1.** Conèixer i comprendre el concepte d'*entorn de programació* dins de l'àmbit del desenvolupament d'aplicacions per a dispositius mòbils.
- 2.** Identificar els diversos entorns de programació existents, coneixent-ne les arquitectures, les característiques dels sistemes operatius usats i les tècniques de programació requerides en cadascun.
- 3.** Comprendre el fet que els entorns no funcionen aïlladament, sinó que coexisteixen.
- 4.** Estar en condicions de triar l'entorn de programació idoni, d'acord amb els requisits de l'aplicació mòbil que es pretén desenvolupar.

1. Història i evolució dels entorns de programació mòbils

La indústria dels dispositius i les aplicacions mòbils és un entorn en canvi constant. Durant el 2010 es va veure com Nokia va abandonar Symbian, esperant que MeeGo i la seva associació amb Windows la rellancin al món dels telèfons intel·ligents. També s'ha vist l'espectacular creixement d'Android, que ha passat per davant de l'iOS d'Apple i de BlackBerry, i s'ha convertit en la segona plataforma darrere de Symbian.

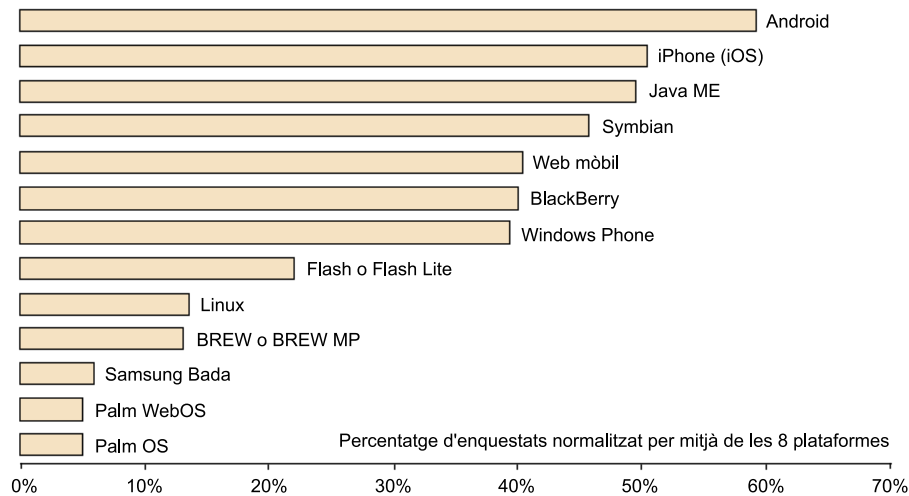
Al mercat dels fabricants d'equips originals es van veure més moviments el 2010 que en els deu anys anteriors. Apple i RIM han avançat alguns dels fabricants tradicionals (Sony Ericsson, Motorola, LG) i han reclamat un lloc al *top 5*. Segons algunes estimacions, ZTE s'hi podria unir aviat. La figura següent ofereix una visió general de com està el mercat de les plataformes mòbils a tot el món. Per països destaca el domini d'Apple als Estats Units i diversos estats d'Europa, com Espanya, i també la cada vegada més destacada presència d'Android. Nokia arrassa a l'Índia, la Xina i altres potències emergents.



La lluita per la supremacia de les plataformes mòbils és candent. Android i iPhone, i també BlackBerry o Nokia, són diversos dels actors que més destaquen.

Per tant, les principals mètriques per al desenvolupament d'aplicacions mòbils han canviat molt i ho continuaran fent. Una és la popularitat de les diferents plataformes entre els desenvolupadors. En els últims temps s'ha produït una migració en les preferències dels desenvolupadors que els ha mogut des de la "vella guàrdia" (Symbian, BlackBerry i Java) cap als nous reis del sector: l'iOS i l'Android. Segons alguns estudis, prop del 60% dels desenvolupadors ha desenvolupat aplicacions per a l'Android, tal com es pot veure a la figura següent. L'iOS d'Apple ocupa el segon lloc amb més del 50%, seguit de Java ME en tercera posició. Així podem veure com els desenvolupadors canvien el focus de la seva atenció envers unes plataformes mentre n'abandonen d'altres.

Plataformes més usades pels desenvolupadors d'aplicacions mòbils el 2010



Font: Mobile Developer Economics 2010 and Beyond. Produït per VisionMobile. Patrocinat per Telefonica Developer Communities. Juny 2010. Amb llicència Creative Commons Atribució Llicència 3.0. Qualsevol ús o modificació d'aquest treball ha d'incorporar aquest avís.

Sens dubte, el canvi més significatiu en els últims temps està essent que la distància entre l'Android i l'iOS, d'una banda, i la resta de plataformes, d'una altra, s'està fent cada vegada més gran. L'App Store d'Apple conté més de 300.000 aplicacions, mentre que estimacions recents situen el nombre d'aplicacions del Market d'Android entorn de les 130.000.

Mentrestant, Nokia ha estat posant un esforç considerable en l'Ovi Store, i de fet ha incrementat la seva popularitat entre consumidors i desenvolupadors, encara que encara li queda un llarg camí per a assolir els dos gegants dispensadors d'aplicacions.

La raó per la qual els desenvolupadors es mouen cap a l'iOS i l'Android pot tenir explicacions diverses, però els motius més importants són els que s'exposen a continuació. D'una banda, Apple ofereix una plataforma que és relativament fàcil d'aprendre i usar, i amb què el desenvolupador pot dissenyar UI¹ molt bones. A més, tenen el repositori d'aplicacions més gran, i encara que el problema de la certificació és un inconvenient per a alguns, no hi ha els problemes de portabilitat i fragmentació. L'Android, d'altra banda, ha estat guanyant ímpetu en tots els camps, i ha assaltat els mercats clau dels seus competidors.

⁽¹⁾interfície d'usuari o *user interface*

Sens dubte, té molts inconvenients derivats de la fragmentació però aquests moltes vegades es passen per alt a causa de la dependència de molts fabricants d'aquesta plataforma.

Un altre aspecte important per a comentar és la disparitat que hi ha hagut entre vendes de dispositius per a cada plataforma i el nombre d'aplicacions disponibles. Caldria esperar que les plataformes amb més penetració al mercat fossin les que dominessin en termes d'aplicacions, però res més lluny de la realitat, almenys fins al 2011.

Agafant com a referència el tercer quadrimestre del 2010, es podia veure que les dues plataformes amb la penetració més baixa, l'iOS i l'Android, tenien el nombre més gran d'aplicacions disponibles.

En el costat oposat, mentre Java ME i Flash Lite han tingut la màxima penetració al mercat de llarg, tot just es poden comparar a les noves plataformes quant a volums d'aplicacions.

En el quart quadrimestre aquest contrast es va fer fins i tot més gran. Tant la botiga de l'Android com la de l'iOS van créixer pràcticament en 100.000 aplicacions cadascuna. El Windows Phone va mostrar un creixement digne d'esmentar, i va assolir 4.000 aplicacions amb prou feines en dos mesos, encara que li queda un llarg camí per a convertir-se en una amenaça seriosa per als dos actors principals.

Per tant, es veu que cada vegada hi ha més desenvolupadors dedicats a aquest tipus d'aplicacions. No obstant això, cal assenyalar aquí que les enquestes indiquen que la majoria reconeixen pocs beneficis econòmics, mentre que només un 5% assenyalava beneficis per sobre de les seves expectatives.

Si bé és cert que assistim a una explosió de botigues d'aplicacions, això no és necessàriament una benedicció per als desenvolupadors. Molts desenvolupadors s'enfronten a problemes perquè els usuaris potencials descobreixin les seves aplicacions, enterrades sota de milers d'altres aplicacions. Es podria dir que és com anar a una botiga de discos amb 200.000 CD: només es mira el *top* 10.

En aquest sentit, una opció per als desenvolupadors és adoptar una estratègia d'**aparador múltiple**, mentre s'adapta el model de benefici a botigues d'aplicacions específiques. Per exemple, fins ara ha estat molt difícil vendre contingut per a l'Android, de manera que sembla que la millor estratègia en aquesta plataforma, almenys per a donar-se a conèixer, és que l'aplicació sigui gratis.

Per tant, la conclusió que es pot treure de tot això és que cal estar molt atent per a veure com evoluciona el desenvolupament d'aplicacions, els beneficis, la distribució, la venda al detall, la portabilitat i la fragmentació, com a factors importants entre molts d'altres.

2. Aplicacions web i aplicacions natives

Abans d'entrar a fons en els entorns de programació d'aplicacions per a dispositius mòbils, establirem de manera resumida l'àmbit en el qual ens mourem. A continuació es defineixen termes clau i es comparen els avantatges i inconvenients dels dos paradigmes de desenvolupament més comuns.

Per començar, definirem què s'entén per *aplicació web* i *aplicació nativa* i considerarem els avantatges i inconvenients de cadascuna.

Una **aplicació web** és bàsicament un lloc web específicament optimitzat per a un dispositiu mòbil. Les característiques que defineixen una aplicació web són que la interfície d'usuari es construeix amb tecnologies web estàndard, que és disponible en un URL² (públic, privat o protegit per una contrasenya) i que està optimitzada per als dispositius mòbils. Una aplicació web no està instal·lada al dispositiu mòbil.

⁽²⁾localitzador uniforme de recursos o *uniform resource locator*

En el cas de l'aplicació web, el lloc pot ser qualsevol, des d'una web anunci d'un petit negoci estàndard a un calculador d'hipoteques o un controlador de calories diari –el contingut és irrellevant.

Les **aplicacions natives**, per contra, estan instal·lades al dispositiu mòbil, tenen accés al maquinari o *hardware* (altaveus, acceleròmetre, càmera, etc.) i estan escrites en algun llenguatge de programació compilat com a Objective-C.

Diferents aplicacions tenen diferents requisits. Algunes aplicacions s'adapten millor a les tecnologies web que d'altres. Conèixer els avantatges i inconvenients de cada paradigma ajudarà a decidir quin camí és apropiat per a cada situació.

El principal avantatge del desenvolupament d'aplicacions natives és que es pot accedir a totes les característiques de maquinari del dispositiu.

Els inconvenients del desenvolupament d'aplicacions natives són els següents:

- L'aplicació només funcionarà a la plataforma escollida.

- Cal desenvolupar-la usant el llenguatge de programació establert per a la plataforma.
- És més complicat distribuir pegats o actualitzacions que solucionin errors.
- El cicle de desenvolupament és més lent.

Quant al desenvolupament d'aplicacions web, aquests són els avantatges:

- Els desenvolupadors web poden usar les seves pròpies eines.
- Es poden usar els coneixements i les habilitats que ja es tinguin quant a disseny i desenvolupament web.
- L'aplicació funcionarà en qualsevol dispositiu que tingui un navegador web.
- Es poden solucionar errors en temps real.
- El cicle de desenvolupament és més ràpid.

Els inconvenients del desenvolupament d'aplicacions web són els següents:

- No es pot accedir a totes les característiques del dispositiu mòbil.
- Pot ser difícil aconseguir efectes sofisticats en la interfície d'usuari.

Quina aproximació és la millor en cada cas és un debat interessant. La naturalesa dels dispositius mòbils que cada vegada més estan sempre connectats fa que es difumini la línia entre aplicacions web i aplicacions natives. Fins i tot hi ha diversos projectes –dels quals PhoneGap és el més notable– que desenvolupen solucions que permeten als desenvolupadors web agafar una aplicació web i empaquetar-la com una aplicació nativa, sia per a l'iPhone o una altra plataforma mòbil.

Objective-C

Per a implementar una aplicació nativa per a l'iPhone cal programar en Objective-C.

3. Enumeració dels diferents entorns

Igual que el sistema operatiu d'un ordinador, un sistema operatiu mòbil és la plataforma de programari que determina les funcions i les característiques disponibles al dispositiu, com control de teclats, seguretat sense fils, sincronització amb aplicacions, correu electrònic o missatges de text. El sistema operatiu mòbil determina també quines aplicacions de tercers parts es poden instal·lar al dispositiu. Per tant, cada sistema operatiu defineix uns entorns sobre els quals podem crear aplicacions. En aquest apartat farem un repàs dels més importants.

Microprogramari

El sistema operatiu d'un dispositiu es coneix en anglès com a *firmware*.

3.1. Entorns per a dispositius de diferents venedors

En aquest subapartat s'estudiaran les plataformes de programari que poden funcionar en diferents plataformes de maquinari de diferents fabricants. En concret, s'explica la història i les característiques principals de les següents:

- Java ME.
- Symbian.
- Android.
- Windows Mobile.
- Qt *framework*.
- BREW.
- Palm OS.

3.1.1. Java ME

El 1999, Sun va desenvolupar una versió de Java especialment dissenyada per a dispositius mòbils, Java 2 Micro Edition, basada en una màquina virtual anomenada *KVM*. Aquesta primera versió només contenia una única màquina virtual i una única API (inicialment dissenyades per al Palm OS), fet que va posar de manifest la insuficiència d'aquesta solució per a la gran varietat de dispositius diferents que hi havia. D'aquesta manera, el 2000 va néixer la primera versió d'una configuració, és a dir, el *connected limited device configuration* (J2ME CLDC 1.0). Una configuració ofereix l'API bàsica per a programar dispositius, encara que no aporta totes les classes necessàries per a desenvolupar una aplicació completa. Per tant, la primera configuració no tenia les eines necessàries per a permetre als desenvolupadors escriure programes per al dispositiu Palm. El juliol del 2000 va néixer la primera implementació d'un perfil, concretament anomenada *mobile information device profile* (MIDP), que no estava destinada a PDA sinó a telèfons mòbils i a paginadors. A partir d'aquest pri-

mer perfil, J2ME va ser considerablement acceptat per la comunitat de desenvolupadors de dispositius mòbils, i s'ha expandit a una gran velocitat fins als nostres dies.

Java ME³ (anteriorment coneguda com a *J2ME*⁴) és, per tant, una edició de Java orientada a dispositius petits. És una versió retallada del Java SE amb certes extensions enfocades a les necessitats particulars d'aquest tipus de dispositius. Aquesta tecnologia consisteix en una màquina virtual i un conjunt d'API⁵ adequades per a aquests dispositius.

Aquesta plataforma normalment produeix aplicacions portables, encara que algunes vegades hi ha biblioteques específiques de cada dispositiu (normalment usades per a jocs), que les fan no portables. Malgrat això, Java ME s'ha convertit en una bona opció per a crear aplicacions per a telèfons mòbils, a causa que es pot emular en un PC durant la fase de desenvolupament i després carregar-les fàcilment al mòbil. Encara que el procés no sigui directe, com que s'utilitzen tecnologies Java per al desenvolupament, resulta bastant econòmic portar-les a altres dispositius.

S'usa moltes vegades per a proporcionar aplicacions simples a telèfons mòbils de gamma baixa. Per tant, les aplicacions (incloent-hi les dades d'aquestes aplicacions) no poden ocupar massa memòria si s'han d'executar en la majoria d'aquests telèfons. A més, han d'estar criptogràficament signades per a poder usar API com la d'accés al sistema de fitxers. Això és relativament car i rarament es fa, fins i tot per a aplicacions comercials. Java ME s'executa sobre una màquina virtual que permet un accés raonable, però no complet, a les funcionalitats del dispositiu sobre el qual s'executa l'aplicació. El procés JSR⁶ serveix per a incrementar gradualment la funcionalitat disponible per a Java ME, mentre proporciona als operadors i als fabricants la capacitat de prevenir o limitar l'accés al programari disponible.

3.1.2. Symbian

Symbian té una història que comença temps enrere, el 1981. En la cronologia següent es pot veure l'evolució del sistema operatiu Symbian:

- 1981. Psion llança el seu primer producte, el Flight Simulator.
- 1984. Psion Organiser veu la llum.
- 1990. SIBO SO (16 bits).
- 1997. EPOC SO (32 bits).
- 1998. El nom de Symbian apareix per primera vegada.

⁽³⁾Java Micro Edition

⁽⁴⁾Java 2 Platform, Micro Edition

⁽⁵⁾interfície de programa d'aplicació o *application program interface*

⁽⁶⁾sol·licituds d'especificació per a Java o *Java specification requests*

- 1999. EPOC versió 5.
- 2000. Symbian 6.0.
- 2001. Symbian 6.1.
- 2003. Symbian 7.0.
- 2004. Symbian 8.0.
- 2005. Symbian 9.0.
- 2008. Nokia compra Symbian Ltd., l'empresa que hi ha darrere del Symbian OS.
- 2009. Creació de la Symbian Foundation.
- 2010. Es publica el codi font del Symbian amb llicència EPL⁷.
- 2011. Nokia du a terme una important aliança amb Microsoft i deixa de costat el sistema operatiu Symbian, que seria reemplaçat pel Windows Phone 7.

⁽⁷⁾llicència pública Eclipse o *Eclipse public license*

Symbian és un sistema operatiu fruit de l'aliança de diverses empreses de telefonia mòbil, entre les quals hi ha Psion, Nokia, Ericsson i Motorola, amb la intenció de desenvolupar i estandarditzar un sistema operatiu que permetés els telèfons mòbils de diferents fabricants intercanviar informació.

El Symbian OS va ser durant uns anys el sistema operatiu estàndard per als telèfons intel·ligents de l'època, ja que més del 85% dels fabricants d'aquests dispositius tenien llicències per a usar-lo. El Symbian OS estava dissenyat per als requisits específics dels telèfons mòbils 2.5G i 3G.

Dissenyada des de l'inici per a dispositius mòbils, la plataforma Symbian és un sistema operatiu de temps real, multitasca, específicament pensada per a funcionar bé en sistemes amb recursos limitats, maximitzant l'eficiència i la durada de la bateria, mentre que minimitza l'ús de memòria. La Symbian Foundation manté el codi per a la plataforma de programari lliure basada en el Symbian OS i aportacions de programari de Nokia, NTT DOCOMO i Sony Ericsson, incloent-hi les interfícies d'usuari S60 i MOAP(S). La plataforma és totalment de codi obert, i la majoria es proporciona amb la llicència pública de l'Eclipse.

Popularitat del Symbian OS

S'han venut més de 300 milions d'unitats basades en el Symbian OS i durant anys ha gaudit de més del 50% de quota de mercat.

El sistema operatiu Symbian va incorporar el suport a pantalles tàctils gràcies a UIQ⁸. UIQ és una interfície gràfica d'usuari basada en l'ús d'un llapis que es pot trobar en telèfons 2.5G i 3G de les marques següents: Motorola, Sony Ericsson,

⁽⁸⁾interfície d'usuari Quartz o *user interface Quartz*

BenQ i ARIMA. Els telèfons UIQ utilitzen pantalles tàctils amb una resolució de 208-240 × 320 píxels i profunditat de color de 12, 16, 18 o 24 bits, depenent de la versió d'UIQ o del terminal. Les últimes versions d'UIQ van ser les 3.x.

3.1.3. Android

L'Android és una plataforma, basada en el Linux, de l'Open Handset Alliance, entre els 34 membres de la qual s'inclouen Google, HTC, Motorola, Qualcomm i T-Mobile. Per tant, 34 de les principals companyies de programari, maquinari i telecomunicacions donen suport a aquesta plataforma. El nucli Linux s'usa com a HAL⁹. La programació d'aplicacions es fa bàsicament en Java. És necessari l'SDK¹⁰ específic de l'Android per a desenvolupar, encara que es pot usar qualsevol IDE¹¹ Java. El codi que sigui crític quant a rendiment es pot escriure en C, C++ o altres llenguatges de codi natiu usant l'NDK¹² de l'Android.

⁽⁹⁾capa d'abstracció de maquinari o *hardware abstraction layer*

⁽¹⁰⁾equip de desenvolupament de programari o *software development kit*

⁽¹¹⁾entorn integrat de desenvolupament o *integrated development environment*

⁽¹²⁾equip de desenvolupament natiu o *native development kit*

3.1.4. Windows Mobile

La plataforma Windows Mobile va ser disponible en una gran varietat de dispositius de diferents operadors sense fils. Es podia trobar el programari Windows Mobile en productes Dell, HP, Motorola, Palm i i-mate. Els dispositius amb el Windows Mobile eren disponibles per a xarxes GSM o CDMA.

El Windows Mobile és una variant del Windows CE per a telèfons mòbils. Originalment el Windows CE es va desenvolupar per a ordinadors de butxaca i PDA amb pantalles tàctils que funcionaven amb un llapis, i es va adaptar posteriorment per a usar-lo en telèfons intel·ligents equipats amb un teclat. Per tant, els telèfons es van convertir en la base més gran de dispositius instal·lats amb el CE, encara que la quota de mercat ha caigut dràsticament des de l'aparició de l'Android i l'iPhone. El Windows Mobile suporta un subconjunt de la interfície de programació de Win32 i una GUI¹³ simplificada amb una finestra a la pantalla alhora. Les aplicacions poden usar el .NET Compact Framework. El Windows Mobile 6.5 va introduir les interfícies d'estil iPhone basades en el contacte amb els dits, mentre que el Windows Phone 7 és un redisseny substancial que usa Silverlight i XNA per a interfícies d'usuari més riques.

⁽¹³⁾interfície gràfica d'usuari o *graphical user interface*

3.1.5. Qt framework

El Qt usa C++ estàndard però fa un ús extensiu d'un preprocessor especial, anomenat MOC¹⁴, per a enriquir el llenguatge. També es pot usar el Qt en altres llenguatges de programació usant enllaços entre llenguatges. Funciona sobre les principals plataformes i té un suport internacional extens. Entre les característiques no relacionades amb la GUI s'inclouen accés a bases de dades SQL, tractament d'XML, gestió de fils, suport de xarxa i una API multiplataforma unificada per a la gestió de fitxers.

⁽¹⁴⁾*meta object compiler*

3.1.6. BREW

El BREW s'usa per a aplicacions en dispositius CDMA (però també suporta models GPRS/GSM). Les aplicacions es distribueixen via una plataforma de contingut BREW i ha tingut poca penetració a Europa. El BREW pot proporcionar control complet del dispositiu i accedir a tota la funcionalitat que té. No obstant això, el potencial que proporciona el codi natiu amb accés directe a les API del dispositiu ha causat que el procés de desenvolupament en BREW hagi hagut de ser adaptat en gran manera per a tots els venedors de programari reconeguts. Mentre que l'SDK del BREW és disponible de manera lliure, executar programari en maquinari real d'un dispositiu mòbil (en contraposició amb l'emulador proporcionat) requereix una signatura digital que es pot generar amb eines publicades per diversos participants, essencialment proveïdors de contingut per a mòbils i Qualcomm. Fins i tot llavors, el programari només funcionarà en dispositius habilitats per a test. Per a baixar-lo en telèfons normals, el programari ha de ser comprovat i provat i ha de rebre l'aprovació de Qualcomm via el seu programa de verificació True BREW.

3.1.7. Palm OS

Des de l'aparició del primer Palm Pilot el 1996, la plataforma Palm OS ha proporcionat als seus dispositius mòbils eines de negoci essencials, i també la capacitat d'accedir a Internet o a una base de dades central corporativa via una connexió sense fils.

El Palm OS va tenir una gran acceptació empresarial en l'important mercat dels Estats Units, basada en els PDA Palm.

El Palm webOS és el sistema operatiu mòbil de propietat evolució del Palm. Funciona sobre un nucli Linux que suporta multitasca. Es va llançar amb Palm Pre i Pixi, i ara és propietat de Hewlett Packard.

3.1.8. Flash Lite

S'usa en dispositius que suporten el reproductor Flash Lite.

3.1.9. Micronavegador

Els entorns basats en el concepte de *micronavegador* o *microbrowser* proporcionen una funcionalitat limitada via una interfície web.

3.2. Desenvolupament multiplataforma

En aquest subapartat descriurem entorns de treball que permeten desenvolupar aplicacions que funcionin tant en l'iPhone com en l'Android.

3.2.1. Titanium Mobile

El Titanium és un entorn de codi obert que permet desenvolupament multiplataforma. Es pot desenvolupar una aplicació que funcioni en dispositius mòbils (iOS, Android, RIM) o plataformes d'escriptoris (OS X, Windows).

El codi font de l'aplicació s'escriu tot en JavaScript, CSS i HTML5. Això és positiu en el sentit que no es necessita aprendre llenguatges complexos com Objective-C o C++.

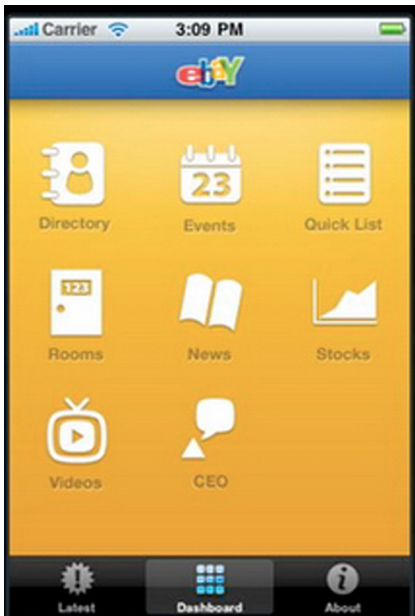
El Titanium és extensible: es pot estendre l'entorn afegint mòduls propis en Objective-C o Java per al cas de l'Android.

Amb el Titanium un desenvolupador es pot beneficiar de l'ús del següent:

Nota

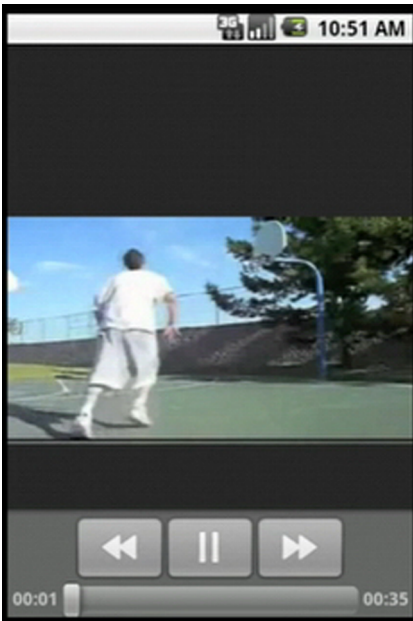
Hi ha una gran quantitat de documentació per al Titanium.

1) Interfícies natives



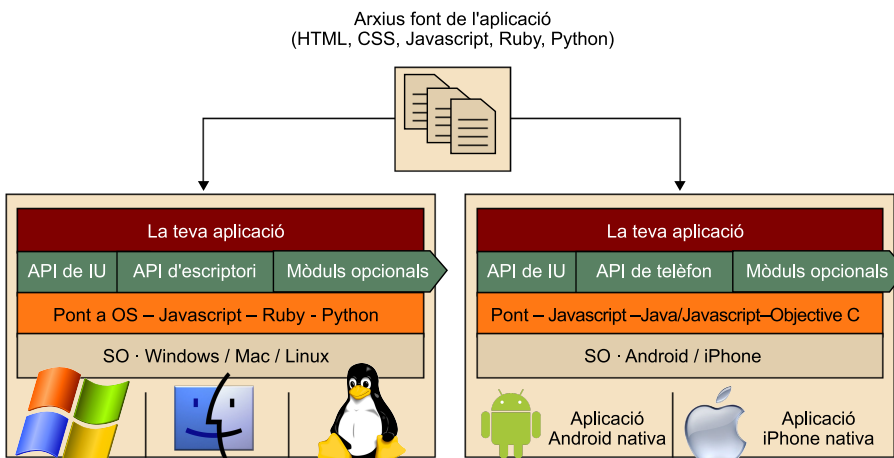
Interfície nativa

2) Aplicacions multimèdia



Aplicació multimèdia

3) Entorn mòbil i d'escriptori



El Titanium permet desenvolupar tant aplicacions mòbils com d'escriptori.

4) Llenguatge JavaScript

Tot això és possible gràcies al fet que el Titanium té un pont que tradueix el codi JavaScript al codi equivalent Objective-C o Java en temps d'execució.

3.2.2. PhoneGap

El PhoneGap permet desenvolupar aplicacions per a l'Android usant tecnologies web com HTML, CSS i JavaScript, i pot convertir aquestes aplicacions web en aplicacions natives d'Android. De fet, el PhoneGap suporta múltiples plataformes com Android, iPhone, Palm, Windows Mobile i Symbian, de mane-

⁽¹⁵⁾Institut Tecnològic de Massachusetts (Massachusetts Institute of Technology)

ra que es pot usar el mateix codi font per a crear aplicacions per a múltiples plataformes. Malgrat que es venguin com a "eines de tecnologia web", el que ofereixen el PhoneGap o altres entorns de treball similars com el Titanium és accés al maquinari de la màquina: es poden fer aplicacions en HTML i JavaScript que usin la càmera, la brúixola o l'acceleròmetre. El PhoneGap és, a més, lliure amb llicència MIT¹⁵.

Per tant, el PhoneGap és una solució de codi obert dissenyada per a donar accés JavaScript als desenvolupadors web a característiques populars dels dispositius mòbils com la càmera, el GPS, l'acceleròmetre o bases de dades SQLite locals sense la necessitat d'haver d'escriure aplicacions completes. La idea de fons és fer fàcil el desenvolupament d'aplicacions mòbils.

Per a aconseguir això, l'entorn PhoneGap actua com un pont entre les aplicacions web i els dispositius mòbils. Permet als desenvolupadors envoltar aplicacions web dins d'una aplicació nativa, i així fer el desenvolupament més fàcil per a aquells que no estan familiaritzats amb Objective-C i Cocoa.

3.3. Entorns per a dispositius d'un venedor únic

Les plataformes de programari següents només funcionen en plataformes de maquinari d'un fabricant específic:

1) **BlackBerry**. BlackBerry té suport per a correu electrònic, telèfon mòbil, missatges de text, tramesa de faxes, navegació per Internet i altres serveis d'informació sense fils, i també una interfície tàctil. Els dispositius BlackBerry disposen de sèrie d'un teclat QWERTY optimitzat per a utilitzar-lo teclejant amb els polzes. Quan van aparèixer, els dispositius BlackBerry van agafar aviat una posició dominant al mercat nord-americà dels telèfons intel·ligents. Per a BlackBerry són importants el BES¹⁶ i el BlackBerry MDS¹⁷.

2) **iOS d'Apple**. L'SDK per a l'iPhone i l'iPod usa Objective-C, que està basat en el llenguatge de programació C. En el seu moment només estava disponible en Mac OS 10.5+ i era l'única manera d'escriure una aplicació per a l'iPhone. A més, Apple ha de verificar totes les aplicacions abans d'allotjar-les en l'App Store, l'únic canal de distribució per a les aplicacions per a l'iPhone i l'iPod Touch. No obstant això, es poden llançar aplicacions per a iPhones piratejats no aprovades per Apple via l'instal·lador Cydia. Aquest sistema s'usa també per a la tauleta iPad.

jQuery i jQTouch

El PhoneGap permet desenvolupar aplicacions amb biblioteques de JavaScript com jQuery/jQTouch.

⁽¹⁶⁾BlackBerry Enterprise Server

⁽¹⁷⁾Mobile Data System

4. Llenguatges de programació

Com s'ha vist prèviament, hi ha dues classes principals d'aplicacions per a dispositius mòbils: aplicacions natives i aplicacions web. Un tercer cas de paradigma és el marcat per Java. Per a aquest cas, molts dels nous mòbils suporten alguna versió d'MIDP¹⁸ i el desenvolupament en aquest entorn és bastant senzill. Per a la instal·lació, no obstant això, és una mica més complicat. En general s'instal·len aplicacions per mitjà d'enllaços a Internet, però alguns operadors o fabricants posen límits en les aplicacions que es poden instal·lar al mòbil.

⁽¹⁸⁾Mobile Information Device Profile

Per a aplicacions natives, depèn del sistema operatiu del mòbil. Per a molts, desenvolupar aplicacions natives pot costar diners (per a eines i SDK) i també hi ha problemes en la manera com es distribueixen les aplicacions. La instal·lació i la depuració d'errors són també diferents depenent del sistema operatiu.

Per tant, el llenguatge de programació que s'usi serà probablement dictat pel dispositiu i la plataforma per a la qual es vol desenvolupar una aplicació, a més de l'aplicació que es vol crear.

A continuació s'enumeren els diferents llenguatges amb què es poden desenvolupar aplicacions natives per a diferents plataformes:

- Si es vol fer una aplicació per a l'iPhone o l'iPod Touch, s'usarà **Objective-C**.
- Si es vol fer una aplicació Android, s'usarà **Java**.
- Si es vol fer una aplicació per a BlackBerry, s'usarà **Java Micro Edition**.
- Si es vol fer una aplicació per al Symbian OS, es pot usar C++, **Java** o **.NET Compact Framework**.
- Si es vol fer una aplicació per al Windows Mobile, les opcions són **Visual C++**, **Visual C#**, **Visual Basic**, **JScript** i **ASP.NET**.

Per tant, la plataforma o el dispositiu dictaran quins llenguatges de programació es tenen com a opcions. Si es vol desenvolupar per a una plataforma que permet tant C++ com Java, llavors el tipus d'aplicació que es planeja desenvolupar podria dictar quin llenguatge és la millor opció.

A continuació es mostra una compilació dels llenguatges de programació més populars per a dispositius mòbils.

4.1. Llenguatges de programació per al Windows Mobile

Es pot escollir entre diverses opcions de llenguatges de programació a l'hora de desenvolupar aplicacions per a dispositius amb el Windows Mobile. En aquest subapartat veurem aquestes opcions i farem una breu descripció de cadascuna.

4.1.1. Visual C++

Es coneix C++ com un llenguatge de desenvolupament "natiu", a causa que interactua directament amb el maquinari d'un dispositiu Windows Mobile, sense la intervenció de cap altra capa (al contrari que Visual C#, per exemple). Programar usant C++ pot ser un desafiament, ja que no és un llenguatge trivial d'aprendre. Alguns errors en un programa C++ poden bloquejar potencialment tot el dispositiu.

Els avantatges d'usar Visual C++ són la velocitat d'execució, la mida de l'aplicació i la flexibilitat. Les aplicacions escrites en C++ s'executen molt ràpid i consumeixen els recursos mínims.

Lectura recomanada

Una bona manera d'aprendre Visual C++ és investigar la Visual C++ Express Edition de Visual Studio (que és gratis), mirar el vídeo d'entrenament i els *webcasts*, i llegir la documentació. Encara que l'Express Edition de Visual Studio no permet desenvolupar aplicacions per al Windows Mobile, gairebé tot el que s'apregui sobre desenvolupament d'aplicacions es pot aplicar directament a dispositius mòbils.

Les aplicacions Visual C++ poden interaccionar amb el dispositiu Windows Mobile cridant a les API Win32. Aquestes API són funcions que fan accions concretes, com fer que se senti un so o dibuixar un botó a la pantalla. Hi ha milers d'aquestes API (el Windows Mobile suporta un subconjunt del conjunt complet d'API Win32 per a escriptori), i estan documentades a la secció titulada "Windows Mobile Features (Native)" de l'MSDN⁽¹⁹⁾ de Microsoft. Quan es navega per aquesta secció, cal anar amb compte que algunes API només són disponibles per al Windows Embedded CE –una plataforma que està relacionada però separada del Windows Mobile.

Si es té experiència desenvolupant per al Windows usant Visual C++, la transició al Windows Mobile no és especialment complicada. Caldria aprendre a instal·lar i usar les eines específiques, i després aprendre a utilitzar les característiques específiques del dispositiu, cosa que permetrà explotar les capacitats dels dispositius.

Per a començar una aplicació Visual C++, cal arrencar el Visual Studio i seleccionar *Archivo > Nuevo > Proyecto* i seleccionar *Smart Device* en el node Visual C++.

Errors en un programa C++

+

Accedir a memòria que ha estat alliberada o oblidar-se d'alliberar memòria en un programa C++ pot bloquejar un dispositiu Windows Mobile.

Jocs d'acció

Els jocs d'acció en temps real són bons exemples de programes que es beneficien de la velocitat d'execució de C++.

⁽¹⁹⁾Microsoft Developer Network

Si som nous tant programant com amb el Windows Mobile, seria una bona idea començar amb Visual C# i llavors fer la transició a Visual C++.

4.1.2. Visual C# i Visual Basic

Visual C# i Visual Basic .Net són llenguatges de desenvolupament més senzills que Visual C++. No solament són relativament fàcils d'aprendre, sinó que a més tenen suport per al .NET Compact Framework.

Les eines de desenvolupament per a C# i Visual Basic .NET inclouen un dissenyador complet d'interfície d'usuari WYSIWYG²⁰. Es poden arrossegar i col·locar botons i altres controls directament a la finestra de l'aplicació, i llavors fer doble clic per accedir al codi que hi ha per sota. Aquest sistema fa que crear interfícies d'usuari per a les aplicacions sigui extremament ràpid i fàcil.

Com a part de la biblioteca Compact Framework, hi ha disponibles classes extra que cobreixen des d'estructures de dades fins a intercepció de missatges de text. Per a fer ús de les característiques específiques del Windows Mobile, es proporcionen un conjunt de classes extra. Aquestes classes proporcionen accés a les característiques del dispositiu, com per exemple la llista de contactes o la càmera.

Si es té experiència desenvolupant aplicacions per a Windows usant Visual C#, la transició hauria de ser relativament senzilla. El Compact Framework és un subconjunt del .NET Framework, per la qual cosa el codi pot necessitar lleugeres modificacions per a algunes funcionalitats.

Visual C# és una bona manera d'aprendre programació. Per a aprendre tot el necessari tant de Visual C# com de Visual Basic, es pot acudir a l'MSDN de Microsoft.

Per a començar una aplicació Visual C# o Visual Basic .NET, cal arrencar el Visual Studio i seleccionar *Archivo > Nuevo > Proyecto* i seleccionar *Smart Device* en el node corresponent al llenguatge que ens interressi.

4.1.3. JScript

El navegador web inclòs amb els dispositius Windows Mobile –Internet Explorer Mobile– suporta JScript. JScript és un superconjunt del llenguatge conegut com a *JavaScript*. Els programes JScript són fitxers de text pla que executa el navegador web. Poden estar incrustats en una pàgina HTML o emmagatzemats en fitxers separats.

Una aplicació JScript s'executa dins del navegador web i usa la finestra del navegador web per a l'entrada/sortida d'informació. És possible fer ús de tècniques de programació AJAX²¹ per a proporcionar un grau d'interacció amb

.NET Compact Framework

El .NET Compact Framework és una biblioteca de classes que fan gran quantitat de tasques usades freqüentment en programació, per simplificar el desenvolupament d'aplicacions.

⁽²⁰⁾"What you see is what you get".

⁽²¹⁾JavaScript asíncron i XML o *asynchronous JavaScript and XML*

l'usuari i comunicar-se amb un servidor remot. A causa de la naturalesa de JScript, les aplicacions no poden accedir a dades locals que no siguin simplement galetes, la qual cosa introdueix algunes limitacions.

4.1.4. ASP.NET

Mentre que JScript és una solució del costat del client per a programar aplicacions d'Internet, ASP.NET és una solució del costat del servidor. Amb ASP.NET es poden escriure aplicacions en C# o Visual Basic .NET que siguin en un servidor web i facin tasques complexes, incloent-hi crear controls d'interfície d'usuari i accedir a bases de dades. ASP.NET aïlla les característiques del dispositiu de l'aplicació, i fa senzill executar una aplicació en diversos tipus de dispositius diferents.

Eina de desenvolupament

No es necessita cap eina de desenvolupament especial: un editor de text és suficient per a crear una aplicació JScript. El programa es pot desar de manera local o s'hi pot accedir des d'un servidor web.

5. Exemples d'entorns

En aquest apartat es descriuran de manera més detallada dos entorns: iPhone/iOS i Android.

5.1. iPhone/iOS

L'iOS comprèn el sistema operatiu i les tecnologies que s'usen per a executar aplicacions de manera nativa en dispositius com l'iPad, l'iPhone i l'iPod Touch. Encara que comparteix una herència comuna i moltes tecnologies de base amb el Mac OS X, l'iOS es va dissenyar per a satisfer les necessitats d'un entorn mòbil, en què les necessitats dels usuaris són lleugerament diferents. Si s'han desenvolupat prèviament aplicacions per a Mac OS X, es trobaran moltes tecnologies familiars, però també tecnologies que només són disponibles en l'iOS, com el suport per a interfície tàctil o acceleròmetre.

L'SDK de l'iOS conté el codi, la informació i les eines necessàries per a desenvolupar, verificar, executar, depurar errors i adaptar aplicacions per a l'iOS. Les eines Xcode proporcionen l'entorn bàsic per a editar, compilar i depurar errors en el codi. Xcode també proporciona el punt de llançament per a verificar les aplicacions en un dispositiu iOS i en un simulador iOS –una plataforma que imita l'entorn bàsic iOS però s'executa en un ordinador Macintosh local.

Aquest subapartat proporciona una descripció d'alt nivell de les característiques bàsiques que es poden trobar en l'iOS per a orientar sobre aquesta plataforma.

5.1.1. Visió general del sistema iOS

L'iOS (conegut com a *iPhone OS* abans del 2010) és el sistema operatiu per a dispositius mòbil d'Apple. Originalment desenvolupat per a l'iPhone, s'ha anat estenent per a donar suport a altres dispositius Apple com l'iPod Touch, l'iPad i Apple TV. Apple no dóna llicències per a la instal·lació de iOS en maquinari de terceres parts. El gener del 2011 l'App Store d'Apple contenia més de 300.000 aplicacions iOS, que s'havien baixat col·lectivament més de 10 bilions de vegades. En l'últim trimestre del 2010, tenia el 16% de quota del mercat dels sistemes operatius per a telèfons intel·ligents en termes d'unitats venudes, el tercer per darrere de l'Android de Google i el Symbian. El 2010 es va emportar el 59% del consum web mòbil (sense incloure l'iPad) a Amèrica del Nord.

La interfície d'usuari de l'iOS es basa en el concepte de *manipulació directa*, usant gestos multicontacte. Els elements de control de la interfície consisteixen en lliscadors, interruptors i botons. La resposta a les peticions de l'usuari és immediata i proporciona una interfície fluida. La interacció amb l'iOS inclou gestos com *tocar fort*, *tocar de manera més feble*, *subjectar* i *deixar anar*, els quals tenen definicions específiques en el context del sistema operatiu iOS i la interfície multicontacte d'aquest sistema.

Algunes aplicacions usen els acceleròmetres interns per a respondre quan se sacseja el dispositiu (un resultat comú és l'ordre *desfer*) o fer girar en tres dimensions (un resultat comú és canviar d'orientació vertical a horitzontal i viceversa).

L'iOS està derivat del Mac OS X, amb el qual comparteix la fundació Darwin, i és, per tant, un sistema operatiu semblant a l'Unix per naturalesa.

En l'iOS hi ha quatre capes d'abstracció:

- La capa Core OS.
- La capa Core Services.
- La capa Media.
- La capa Cocoa Touch.

5.1.2. Història del sistema iOS

El sistema operatiu va aparèixer amb l'iPhone en la Macworld Conference & Expo el gener del 2007, i es va llançar el juny d'aquell any. Al principi, els missatges de màrqueting d'Apple no especificaven un nom diferent per al sistema operatiu, i deien simplement que "l'iPhone executa l'OS X". Inicialment, no suportava aplicacions de tercers parts. Steve Jobs argumentava que els desenvolupadors podien programar aplicacions que "es comportarien com a aplicacions natives en l'iPhone". L'octubre del 2007, Apple va anunciar que s'estava desenvolupant un SDK natiu i que planejaven posar-lo "en mans dels desenvolupadors al febrer". El març del 2008 Apple va llançar la primera versió beta juntament amb un nou nom per al sistema operatiu: iPhone OS.

Les grans vendes dels dispositius mòbils d'Apple van encendre l'interès en l'SDK. El setembre previ, Apple havia llançat l'iPod Touch, que tenia la majoria de les capacitats de l'iPhone no relacionades amb la telefonia. Apple, a més, va vendre més d'un milió d'iphones durant les vacances del 2007. El gener del 2010, Apple va anunciar l'iPad, dispositiu amb una pantalla més gran que l'iPhone i l'iPod Touch, i dissenyat per a navegar per Internet, continguts multimèdia i lectura de llibres electrònics.

Espai de l'iOS

El sistema operatiu iOS usa uns escassos 500 MB de la capacitat d'emmagatzemament del dispositiu, depenent del model.

El juny del 2010, Apple va rebatejar l'iPhone OS com a *iOS*. El nom *IOS*²² havia estat usat per Cisco durant una dècada per al seu IOS, usat en encaminadors Cisco. Per a evitar plets potencials, Apple va pagar la llicència per a usar la marca IOS de Cisco.

(22) sistema operatiu d'interconnexió de xarxes o *inter-network operating system*

5.1.3. Història de les versions del sistema iOS

La versió 4, anunciada l'abril del 2010, presentava multitasca, correu electrònic organitzat en fils i diverses característiques orientades a negocis. En la WWDC²³ 2010, Apple va anunciar que l'iPhone OS s'havia rebatejat com a *iOS*. Apple va pagar la llicència per a la marca IOS a Cisco Systems (que posseeix l'IOS), la mateixa companyia amb què Apple havia tingut una disputa sobre la marca iPhone. Apple va llançar l'iOS 4 el juny del 2010, tres dies abans que l'iPhone 4, en un intent de reduir la càrrega en els servidors d'Apple. L'iOS 4 va ser la primera versió del sistema operatiu que era una actualització gratuïta per a l'iPod Touch; Apple cobrava 9,99 \$ per actualitzacions anteriors. Apple va anunciar prèviament que els usuaris de l'iPad amb programari 3.x rebrien una actualització gratuïta a la versió important següent (4.x).

(23) Conferència Mundial de Desenvolupadors d'Apple

L'iOS 4.0.1 incloïa una solució per a l'indicador de la força del senyal rebut. Es va llançar el juliol del 2010, el dia abans que Apple fes una conferència de premsa per a explicar la seva resposta als molt publicats problemes de l'antena de l'iPhone. Apple també va llançar l'iOS 3.2.1 per a l'iPad, que millorava la connectivitat Wi-Fi, la reproducció de vídeo i el copiar i enganxar d'arxius PDF²⁴ adjunts de la tauleta tàctil, a més d'altres actualitzacions.

(24) format de document portàtil o *portable document format*

L'agost del 2010 es van llançar l'iOS 4.0.2 per a l'iPhone i l'iPod Touch, i l'iOS 3.2.2 per a l'iPad, per a arreglar algunes vulnerabilitats de seguretat.

El setembre del 2010 es va llançar l'iOS 4.1 per a l'iPhone i l'iPod Touch; l'actualització arreglava alguns errors detectats pels usuaris, millorava la durada de la bateria i afegia una nova característica anomenada *Game Center*, que permetia als jugadors jugar partides amb altres jugadors, pujar puntuacions altes i desbloquejar assoliments, i afegia suport inicial per a l'iPod Touch 4th Generation i l'Apple TV 2G. L'iOS 4.1 també afegia fotografia HDR²⁵, una característica que només l'iPhone 4 era capaç d'usar. L'iOS 4.1 també afegia una nova característica, anomenada *Ping*, que és una eina de descobriment i xarxa social de música.

(25) alt rang dinàmic o *high dynamic range*

Game Center

Apple va acabar traient el Game Center de l'iPhone 3G a causa d'informes de baix rendiment.

El novembre del 2010 es va llançar l'iOS 4.2 per als desenvolupadors. Mai no es va llançar al públic a causa que es va trobar un error en la part de Wi-Fi en l'edició limitada. Finalment, el que Apple va fer va ser llançar l'iOS 4.2.1 al públic.

L'iOS 4.2.1 es va llançar el novembre del 2010 amb suport per a tots els dispositius Apple A4, tercera i segona generació de dispositius, amb l'exclusió de l'Apple TV. Proporcionava suport inicial de l'iOS 4.x a l'iPad, més AirPlay i AirPrint a tots els dispositius compatibles. A més, conté canvis menors en l'aplicació YouTube i modifica l'animació de multitasca.

L'iOS 4.2.5 es va llançar com a versió de demostració per a la versió CDMA de l'iPhone 4. Aquesta variant iPhone 4 era disponible per als clients de Verizon Wireless als Estats Units. Aquesta versió tenia lleugers canvis en la interfície d'usuari específics per a la versió CDMA del mòbil.

La versió beta de l'iOS 4.3 es va llançar als desenvolupadors el gener del 2011.

5.1.4. Característiques del sistema iOS

Sempre que s'encén el dispositiu o es pressiona el botó d'inici, es presenta la pantalla principal amb icones d'aplicacions i un receptacle en la part inferior on els usuaris poden col·locar les aplicacions usades amb més freqüència. La pantalla té una barra d'estat al llarg de la part superior per a mostrar dades com l'hora, el nivell de bateria o la potència de senyal. La resta de la pantalla es dedica a l'aplicació actual.

Amb l'iOS 4 va arribar la introducció d'un sistema de carpetes simple. Es pot arrossegar qualsevol aplicació i deixar-la anar a sobre d'una altra per crear una carpeta. A partir d'aquell moment, es poden afegir deu aplicacions més a la carpeta usant el mateix procés (el nombre màxim que una carpeta pot gestionar és dotze en l'iPhone i l'iPod Touch i vint en l'iPad). Se selecciona un títol per a la carpeta de manera automàtica segons el tipus d'aplicacions que hi ha a dins, però l'usuari en pot editar també el nom.

La pantalla d'inici de l'iOS conté aplicacions per defecte. Algunes d'aquestes aplicacions no són visibles per defecte i l'usuari hi pot accedir mitjançant l'aplicació *Settings* o un altre mètode.

Totes les utilitats com notes de veu, calculadora o brúixola són en una carpeta anomenada *Utilidades* en el 4.0. Moltes de les aplicacions que hi ha incloses estan dissenyades per a compartir dades.

L'iPod Touch manté les mateixes aplicacions que són presents per defecte en l'iPhone, amb l'excepció de les aplicacions (anteriors a la quarta generació) de telèfon, missatges, brúixola i càmera. L'aplicació *iPod* present en l'iPhone es divideix en dues aplicacions en l'iPod Touch: *Música* i *Vídeos*. La fila inferior d'aplicacions s'usa per a delinear els propòsits principals de l'iPod Touch: *Música*, *Vídeos*, *Safari* i *App Store* (aquesta configuració es va canviar en

Activació de Nike+iPod

Nike+iPod s'activa mitjançant l'aplicació *Settings*, mentre que *AirPrint* s'activa quan l'usuari imprimeix un fitxer.

Compartició de dades

En el sistema iOS es pot seleccionar un número de telèfon d'un correu electrònic i marcar-lo com un contacte o marcar-lo per fer una trucada

l'actualització 3.1). Per a la quarta generació d'iPod Touch, inclou *FaceTime* i *Càmera*, i la configuració del receptacle inferior canvia a *Música*, *Mail*, *Safari*, *Vídeo*.

L'iPad ve amb les mateixes aplicacions que l'iPod Touch excloent-hi *Stocks*, *Timepo*, *Reloj*, *Calculadora*, i l'aplicació *Nike+iPod*. Es proporcionen aplicacions de música i vídeo separades, com en l'iPod Touch, encara que (com en l'iPhone) l'aplicació de música es diu *iPod*. La majoria de les aplicacions estan completament refetes per a beneficiar-se de la pantalla més gran de l'iPad. La configuració per defecte del receptacle inferior inclou *Safari*, *Mail*, *Fotos* i *iPod*.

Multitasca

Abans de l'iOS 4, la multitasca era limitada a una selecció de les aplicacions que Apple incloïa en els dispositius. A Apple li preocupava que executar múltiples aplicacions de tercers parts de manera simultània descarregués la bateria massa ràpid. A partir de l'iOS 4, en dispositius iOS de tercera generació en endavant, hi ha suport per a multitasca per mitjà de 7 API:

- Àudio en segon pla.
- Veu sobre IP.
- Localització en segon pla.
- Notificacions *push*.
- Notificacions locals.
- Acabament de tasques.
- Canvi ràpid d'aplicació.

Pressionar dues vegades el botó d'inici activa l'intercanviador d'aplicació. Llavors apareix una interfície lliscadissa des de la part inferior: escollint la icona corresponent es canvia a aquesta aplicació. A l'esquerra hi ha icones que funcionen com a controls de música. L'usuari també pot finalitzar aplicacions.

Game Center

El Game Center és una "xarxa social de joc" multijugador en línia llançada per Apple. Permet als usuaris "convidar amics a jugar a un joc, començar un joc multijugador, controlar els assoliments i comparar les puntuacions més altes en un tauler de líders".

El Game Center es va anunciar durant una presentació de l'iOS 4 l'abril del 2010. Es va llançar una versió prèvia per als desenvolupadors registrats d'Apple a l'agost. Finalment es va llançar el setembre del 2010 amb l'iOS 4.1 en l'iPhone, l'iPhone 3GS i l'iPod Touch de la segona a la quarta generació. El Game Center va fer el debut públic en l'iPad amb l'iOS 4.2.1. No hi ha suport per a l'iPhone 3G i l'iPhone original. No obstant això, el Game Center és disponible de manera no oficial per a l'iPhone 3G.

5.1.5. Desenvolupament d'aplicacions per a l'iOS

Les aplicacions han d'estar escrites i compilades específicament per a l'iOS i l'arquitectura ARM. El navegador web Safari suporta aplicacions web com altres navegadors. Hi ha disponibles aplicacions natives autoritzades de terceres parts, per a dispositius amb l'iOS 2.0 o posterior, per mitjà de l'App Store de Apple.

SDK

L'octubre del 2007, en una carta oberta, Steve Jobs va anunciar que el febrer del 2008 es posaria a disposició de desenvolupadors externs a Apple un SDK. L'SDK es va llançar el març del 2008 i permet als desenvolupadors fer aplicacions per a l'iPhone i l'iPod Touch, i també verificar-les en un "simulador iPhone". No obstant això, carregar una aplicació als dispositius és només possible després de pagar l'iPhone *developer program*. Des del llançament de l'Xcode 3.1, l'Xcode és l'entorn de desenvolupament per a l'iOS SDK. Les aplicacions iPhone, com l'iOS i el Mac OS X, estan escrites en Objective-C.

Els desenvolupadors poden posar qualsevol preu a les seves aplicacions per sobre d'un mínim perquè es distribueixin per mitjà de l'App Store, del qual rebran el 70% per cada venda. De manera alternativa, poden optar per llançar la seva aplicació de manera gratuïta i no pagar així cap cost per llançar o distribuir l'aplicació excepte el cost de membre.

5.1.6. Jailbreaking

L'iOS ha estat subjecte a una varietat de diferents manipulacions centrades a afegir funcionalitat sense el suport d'Apple. Abans del debut de l'App Store el 2008, la raó principal per al procés d'escapar-se de la presó o *jailbreaking* (o *jailbreak*) era instal·lar aplicacions natives de terceres parts. Apple va dir que no dissenyaria actualitzacions de programari específicament perquè aquestes aplicacions deixessin de funcionar (sempre que no fossin aplicacions que fessin desbloqueig de SIM²⁶), però el cas és que amb cada actualització de l'iOS el *jailbreak* semblava deixar de funcionar.

⁽²⁶⁾mòdul d'identificació de subscriptor o *subscriber identity module*

Des de l'arribada de l'App Store i les aplicacions de terceres parts, l'objectiu de la comunitat de *jailbreaking* ha canviat. El principal objectiu del *jailbreaking* és permetre la personalització del dispositiu, usar emuladors i millores que fa la comunitat com multitasca, l'Adobe Flash Player o accedir al sistema de fitxers de l'iPhone. La multitasca només és disponible en dispositius iOS de tercera generació en endavant, i les aplicacions en l'App Store no tenen permès modificar l'aparença del sistema operatiu.

Alguns *jailbreakers* també intenten compartir de manera il·legal aplicacions de pagament de l'App Store. Aquest objectiu ha causat alguna distensió dins de la comunitat de *jailbreaking*, a causa que no era l'objectiu original del *jailbreaking* i és il·legal. Hi ha també alguns usuaris que s'oposen a la censura de continguts d'Apple.

5.1.7. Gestió de drets digitals

La naturalesa tancada i propietària de l'iOS ha generat crítiques, particularment d'advocats de drets digitals com l'Electronic Frontier Foundation, l'enginyer informàtic i activista Brewster Kahle, l'especialista en lleis d'Internet Jonathan Zittraion i la Free Software Foundation, que va protestar en l'esdeveniment de presentació de l'iPad i ha fet de l'iPad el seu objectiu amb la seva campanya "Defective by Design". El competidor Microsoft també ha criticat el control d'Apple sobre la seva plataforma.

En el conflicte hi ha les restriccions imposades pel disseny de l'iOS, conegudes com a DRM²⁷, destinades a bloquejar els continguts que es compren a la plataforma Apple, el model de desenvolupament (que requereix una subscripció anual per a distribuir aplicacions desenvolupades per a l'iOS), el procés d'aprovació d'aplicacions centralitzat, i també el control general d'Apple sobre la plataforma per si mateixa. Particularment en disputa hi ha la capacitat d'Apple de deshabilitar o esborrar aplicacions de manera remota.

(27) gestió de drets digitals o *digital rights management*

Algunes veus dins de la comunitat tecnològica han expressat preocupació pel fet que l'iOS tancat representi una tendència cada vegada més important envers la visió d'Apple de la informàtica i fan notar el potencial d'aquestes restriccions per a reduir la innovació en programari.

No obstant això, també hi ha veus fora d'Apple que han mostrat el seu suport al model tancat de l'iOS. El desenvolupador de Facebook Joe Hewitt, que va protestar contra el control d'Apple sobre el seu maquinari com un "precedent horrible", ha argumentat després que les aplicacions tancades en l'iPad estan relacionades amb les aplicacions web i proporcionen més seguretat.

5.2. Android

L'Android és una pila de programari de codi obert per a dispositius mòbils que inclou sistema operatiu, programari intermedi i aplicacions bàsiques. Google Inc. va comprar l'empresa desenvolupadora inicial del programari, Android Inc., el 2005. El sistema operatiu de l'Android està basat en una versió modificada del nucli Linux. Google i altres membres de l'Open Handset Alliance col·laboren en el desenvolupament i llançament de l'Android. L'AOSP²⁸ està encarregat del manteniment i desenvolupament de l'Android.

(28) Android Open Source Project

En el quart trimestre del 2010, el sistema operatiu Android va ser la plataforma de telèfon intel·ligent més venuda del món, i va destronar el Symbian de Nokia de la primera posició per primera vegada en deu anys. Altres fonts indiquen que el Symbian estava encara lleugerament per davant en vendes si es tenien en compte alguns telèfons de models antics Symbian no-Nokia.

L'Android té una gran comunitat de desenvolupadors programant *apps*⁽²⁹⁾ que estenen la funcionalitat dels dispositius. El 2010 hi havia vora 200.000 aplicacions disponibles per a l'Android. L'Android Market és la botiga en línia gestionada per Google per mitjà de la qual també es poden baixar aplicacions de llocs de terceres parts. Els desenvolupadors programen principalment en el llenguatge Java, controlant el dispositiu mitjançant biblioteques Java desenvolupades per Google.

(29) programes d'aplicació o *application programs*

Llançament de l'Android

L'arribada de la distribució Android el novembre del 2007 es va anunciar amb la fundació de l'Open Handset Alliance, un consorci de setanta-nou companyies de maquinari, programari i telecomunicacions amb l'objectiu de desenvolupar estàndards oberts per a dispositius mòbils. Google va llançar la majoria del codi Android amb la llicència Apache, una llicència de programari lliure i codi obert.

La pila de programari del sistema operatiu Android consisteix en aplicacions Java que s'executen en un entorn d'aplicacions basat en Java i orientat a objectes a sobre de biblioteques base Java que s'executen en una màquina virtual Dalvik que fa compilació JIT⁽³⁰⁾. També hi ha biblioteques escrites en C, que inclou el gestor de superfície, OpenCore Media Framework, el sistema gestor de base de dades relacional SQLite, l'API gràfica 3D OpenGL ES 2.0, el WebKit *layout engine*, el motor gràfic SGL⁽³¹⁾, SSL, i Bionic *libc*. El sistema operatiu Android consisteix en 12 milions de línies de codi, incloent-hi 3 milions de línies d'XML, 2,8 milions de línies de C, 2,1 milions de línies de Java i 1,75 milions de línies de C++.

(30) just a temps o *just in time*

(31) Scene Graph Library

5.2.1. Història de l'Android

L'octubre del 2003, Andy Rubin, Rich Miner i d'altres van fundar Android Inc. a Palo Alto, Califòrnia (Estats Units).

En paraules de Rubin, l'objectiu era desenvolupar "dispositius mòbils més elegants que tinguin més en compte la localització i les preferències dels seus amos".

Entre altres empleats inicials importants s'inclouen Andy McFadden, que va treballar amb Rubin en WebTV, i Chris White, que va liderar el disseny i la interfície de WebTV, abans d'ajudar a fundar Android.

Rubin, cofundador de Danger Inc., Miner, cofundador de Wildfire Communications Inc. i vicepresident de tecnologia i innovació a Orange, i els altres empleats inicials van portar una considerable experiència en la indústria sense

fills a la companyia. Malgrat els assoliments obvis del passat dels fundadors i primers empleats, Android Inc. va funcionar de manera reservada, admetent simplement que estava treballant en programari per a telèfons mòbils.

L'agost del 2005, Google va adquirir Android Inc. Els empleats principals d'Android Inc., incloent-hi Andy Rubin, Rich Miner i Chris White, van quedar-se a la companyia després de l'adquisició.

En el moment de l'adquisició, a causa del poc coneixement que es tenia sobre el treball d'Android Inc., es va conjecturar que Google estava planejant entrar al mercat dels telèfons mòbils.

A Google, l'equip liderat per Rubin va desenvolupar una plataforma per a dispositius mòbils basada en el nucli Linux. Google va posar al mercat la plataforma per als fabricants de dispositius mòbils i els operadors amb la premissa de proporcionar un sistema flexible i actualitzable. Google va alinear junts una sèrie de socis dedicats a components maquinari i programari i va indicar als operadors que estava obert a diversos graus de cooperació per part seva.

Les especulacions sobre la intenció de Google d'entrar al mercat de les comunicacions mòbils van continuar durant desembre del 2006. Informes de la BBC³² i *The Wall Street Journal* van indicar que Google volia el seu sistema de cerca i les seves aplicacions en telèfons mòbils i estava treballant de valent per aconseguir-ho. Mitjans de comunicació escrits i en línia aviat van publicar rumors que Google estava desenvolupant un dispositiu amb la marca de fàbrica Google. Alguns van especular que mentre Google definia especificacions tècniques, estava mostrant prototips a fabricants de telèfons mòbils i operadors de xarxa.

⁽³²⁾British Broadcasting Corporation

El setembre del 2007, *InformationWeek* va cobrir un estudi d'*Evalueserve* que indicava que Google havia registrat diverses patents d'aplicacions en l'àrea de la telefonia mòbil.

El novembre del 2007, es va anunciar a si mateixa l'Open Handset Alliance, un consorci de diverses companyies que inclou Texas Instruments, Broadcom Corporation, Google, HTC, Intel, LG, Marvell Technology Group, Motorola, Nvidia, Qualcomm, Samsung Electronics, Sprint Nextel i T-Mobile. L'objectiu de l'Open Handset Alliance és desenvolupar estàndards oberts per a dispositius mòbils. El mateix dia, l'Open Handset Alliance també va anunciar el seu primer producte, l'Android, una plataforma per a dispositius mòbils construïda sobre la versió 2.6 del nucli Linux.

El desembre del 2008, s'hi van unir catorze nous membres, incloent-hi Packet-Video, ARM Holdings, Atheros Communications, Asustek Computer Inc., Garmin Ltd., Softbank, Sony Ericsson, Toshiba Corp. i Vodafone Group Plc.

Amb l'excepció de breus períodes d'actualització, l'Android ha estat disponible amb una llicència de programari lliure de codi obert des d'octubre del 2008. Google va publicar tot el codi font (incloent-hi les piles de xarxa i telefonia) amb una llicència Apache. Google també manté pública la llista de problemes revisats perquè qualsevol la pugui veure i comentar.

5.2.2. Història de les versions de l'Android

L'Android ha vist diverses actualitzacions des del llançament original. Aquestes actualitzacions en el sistema operatiu base normalment arreglen errors i afegixen noves funcionalitats. Generalment cada nova versió del sistema operatiu Android es desenvolupa amb un nom en codi basat en un article de postres.

Les versions més recents de l'Android són:

- 2.0/2.1 (**Eclair**), que va millorar la interfície d'usuari i va introduir suport per a HTML5 i Exchange ActiveSync 2.5.
- 2.2 (**Froyo**), que va introduir millores de velocitat amb l'optimització del JIT i el motor Chrome V8 JavaScript, i va afegir suport per a Adobe Flash.
- 2.3 (**Gingerbread**), que refinava la interfície d'usuari, millorava el teclat de programari i característiques de copiar i enganxar, i afegia suport per a NFC.

Glossari

AJAX *f* Acrònim d'*asynchronous JavaScript and XML* (JavaScript asíncron i XML), és una tècnica de desenvolupament web per a crear aplicacions interactives o RIA (*rich Internet applications*). Aquestes aplicacions s'executen en el client, és a dir, en el navegador dels usuaris mentre es manté la comunicació asíncrona amb el servidor en segon pla. D'aquesta manera és possible fer canvis sobre les pàgines sense necessitat de recarregar-les, la qual cosa significa augmentar la interactivitat, velocitat i usabilitat en les aplicacions.

Dalvik *f* Màquina virtual que utilitza la plataforma per a dispositius mòbils Android. Ha estat dissenyada per Dan Bornstein amb contribucions d'altres enginyers de Google.

jailbreak *m* Jailbreak (en català, escapar-se de la presó o, més literalment, trencapresons) és un procés que permet als usuaris dels dispositius iPhone, iPod Touch, iPad i Apple TV de totes les generacions executar aplicacions diferents a les allotjades en l'App Store, el lloc oficial de baixada de programes per a aquests dispositius.

Llicència Apache *f* Llicència de programari lliure (*Apache license* o *Apache software license* per a versions anteriors a la 2.0) creada per l'Apache Software Foundation (ASF). La llicència Apache (amb versions 1.0, 1.1 i 2.0) requereix la conservació de l'avís de *copyright* i l'advertència, però no és una llicència *copyleft*, ja que no requereix la redistribució del codi font quan es distribueixen versions modificades.

Llicència MIT *f* Llicència de programari que ha emprat el Massachusetts Institute of Technology (MIT) al llarg de la seva història, i que potser s'hauria de dir més correctament *licència X11*, ja que és la llicència que portava aquest programari que mostrava la informació de manera gràfica. Sia com a MIT o X11, però, el text és idèntic.

MIDP *m* El *mobile information device profile* (MIDP) és una implementació que permet escriure aplicacions que es poden baixar i serveis per a dispositius mòbils que es connectin a la Xarxa.

webcast *m* Fitxer multimèdia distribuït per Internet usant tecnologies de reproducció a temps real per a distribuir una única font de contingut a diversos observadors simultanis. Es pot distribuir en directe o per comanda.

Bibliografia

Marco, M. J.; Marco, J. M.; Prieto, J.; Segret, R. (eds.) (2010). *Escaneando la informática*. Barcelona: Editorial UOC. ISBN: 978-84-9788-110-4.

Stark, J. (2010). *Building iPhone Apps with HTML, CSS, and JavaScript*. Sebastopol, CA: O'Reilly. ISBN: 978-0-596-80578-4.

Enllaços d'Internet

<http://en.wikipedia.org/>

<http://www.visionmobile.com/>

<http://blog.abrahambarrera.me/>

<http://developer.appcelerator.com/>

<http://www.phonegap.com/>

<http://softlibre.barrapunto.com/>

<http://www.nitobi.com/>

<http://msdn.microsoft.com/>

<http://forums.techarena.in/>

<http://stackoverflow.com/>

<http://developer.apple.com/>