Alma Mater Studiorum · Università di Bologna

FACOLTÀ DI SCIENZE MATEMATICHE, FISICHE E NATURALI
Corso di Laurea Magistrale in Informatica

Rassegna dei framework per lo sviluppo di applicazioni mobili

Relatore:
Prof. Antonio Messina

Tesi di laurea di:
Odeta Qorri

Sessione II
Anno Accademico 2010/2011

A tutti coloro che
hanno sempre creduto in me

Indice

Elenco delle figure

Elenco delle tabelle

Capitolo 1

Introduzione

Lo stato attuale del mercato continua a dimostrare come l'interesse del pubblico verso i dispositivi mobili sia in costante crescita; se fino a pochi anni fa il termine smartphone era ai più privo di significato, ora è abbastanza comune per giovani e adulti essere concentrati su piccoli dispositivi dove si possono contattare i clienti via e-mail, verificare lo stato di avanzamento dei processi aziendali, consultare il web e così via. A queste due categorie di utenza così eterogenee se ne possono aggiungere altre, come quella dei teenager immersi sulla versione mobile di Facebook, quella dei super tecnologi che non vogliono rimanere indietro rispetto alle novità e così via.

Non è quindi un caso se gli smartphone sono il secondo gruppo di dispositivi al mondo, in termini di quantità di connessioni, che utilizzano Internet abitualmente, e nonostante le varie alternative di sistemi operativi per dispositivi mobili (e quindi non solo cellulari, ma anche di tablet PC), non commettiamo nessuna inesattezza identificando i sistemi operativi Mac OS, Blackberry OS, Windows Phone e Android come i prodotti leader.

Ma avere almeno quattro sistemi operativi da dover servire corrisponde, per uno sviluppatore di applicazioni, a dover sviluppare altrettante versioni dello stesso prodotto. Nel caso di una software house di grosse dimensioni spesso il problema non si pone: esse possono disporre di un team che lavora sullo sviluppo del core del prodotto in maniera indipendente dalle piattaforme

target, e un altro team che si occupa di fornire le interfacce comuni per avere un determinato comportamento sulle varie piattaforme. Il problema, però, è che questo tipo di soluzione non è adatto ai singoli sviluppatori o alle software house di piccole dimensioni, le quali, comunque, sono i principali fornitori di applicazioni nei mercati di queste piattaforme.

Una possibile soluzione a questo problema è il framework per lo sviluppo multipiattaforma. Ovviamente è difficile pensare a un'applicazione ludica di questo tipo: come è noto i giochi più movimentati richiedono una grande potenza di calcolo per le operazioni di rendering e quindi non si può pensare di appesantire ulteriormente il loro tempo di esecuzione, ma per le restanti tipologie di applicazioni la cross-platform può essere la soluzione a questa grave problematica.

L'obiettivo di questa tesi è introdurre l'utilizzo di diverse tipologie di framework, prendendo come riferimento i sistemi operativi per i dispositivi mobili.

Lo scopo di questo lavoro di tesi è quello di presentare soluzioni (attraverso i framework) per lo sviluppo di applicazioni mobili su diverse piattaforme software. Il presente lavoro di tesi è organizzato come segue.

Nel primo capitolo si introducono i concetti fondamentali dei dispositivi mobili e il mobile computing.

Nel secondo capitolo si introducono i smartphones, le differenze tra loro, l'evoluzione fino ai giorni d'oggi e l'incremento dell'utilizzo dei dispostivi mobili.

Nel terzo capitolo viene offerta una panoramica sui sistemi operativi più diffusi per PDA e smartphone: Symbian, Blackberry, iOS, Windows Mobile ed Android. Vengono, inoltre, presentati in dettaglio le differenze dei sistemi operativo.

Nel quarto capitolo vengono introdotti alcuni approcci per lo sviluppo di dispositivi mobili. In particolare, viene introdotta la definizione di un'applicazione web e un'applicazione nativa. Inoltre, vengono valutati i pro e i contro di questi due approcci; viene valutato l'approccio migliore e infine

vengono definiti i linguaggi di programmazione di ciascun approccio.

Nel quinto capitolo si concentra sui framework di sviluppo multipiattaforma e in particolare su Rhodes, PhoneGaP, Titanium Mobile e Sencha Touch. Vengono descritte le loro architetture, le loro funzionalità e vengono mostrate due applicazioni create con PhoneGap e Titanium Mobile.

1.1 I dispositivi mobili

Gli ultimi decenni sono stati caratterizzati da una crescita esplosiva nella distribuzione di dispositivi mobili di piccole dimensioni, che vanno dal personal digital assistant a smartphone e sensori wireless. Allo stesso tempo, tante tecnologie di comunicazione sono state distribuite come IEEE 802.11, Bluetooth e reti cellulari. Questo scenario ha portato alla definizione del paradigma del mobile computing, abilitato dai cosiddetti Mobile Distributed Systems (MDSs). Dal momento che questi sistemi diventano sempre più complessi, la loro affidabilità inizia a diminuire a causa di interazioni complesse tra moduli software. Il software sofisticato può contenere bug, può essere difficile da comprendere ed analizzare, e matura in qualità nel tempo. Come conseguenza, la tolleranza del consumatore per le applicazioni va in crisi. Quindi, l'affidabilità di questi dispositivi è direttamente correlata con le opportunità del business. Allo stesso tempo, molte applicazioni basate sui dati sono state introdotte sul mercato del mobile computing. Il phone-base banking, la prenotazione dei biglietti e l'e-trading sono esempi di applicazioni dove l'utente si aspetta di avere un comportamento corretto nonostante gli errori accidentali e/o azioni di malintenzionati. L'affidabilità diventerà ancora più critica con l'emergere di nuove applicazioni per i sistemi mobili, ad esempio, robot di controllo, la telemedicina e video-sorveglianza. In tali scenari, un fallimento dell'applicazione potrebbe causare una grave perdita, come ad esempio nel caso di un robot che esegue azioni incontrollate.

La comunicazione tramite i telefoni cellulari è considerato fin ad ora la tecnologia wireless di comunicazione più utilizzata in tutto il mondo. Uno

studio effettuato da EMC (Hopkinton, MA) [2] ha mostrato che ci sono circa tre miliardi di utenti di telefoni cellulari in tutto il mondo, e si stima che tale numero sarà sempre in crescita nei prossimi anni. Oggigiorno, i telefoni cellulari sono diventati così popolari nel mondo che in alcuni posti come Singapore hanno sostituito le linee tradizionali fisse come prima scelta di comunicazione. Il mercato del mobile wireless nel mondo crescerà sempre di più secondo la Telecommunications Industry Association (TIA) [3]. La rete cellulare e i telefoni cellulari hanno avuto una crescita esponenziale. L'evoluzione della rete cellulare, i servizi, gli standard e i telefoni cellulari potranno comportare nuove direzioni di sviluppo nel mondo del mobile.

1.2 Il mobile computing

Alcuni credono che il futuro della tecnologia informatica possa essere il *mobile computing*[1] [1]. Il *mobile computing* non implica soltanto l'uso di un telefono cellulare e il controllo degli orari, e-mail, il tempo e le news occasionalmente con un computer laptop oppure un PDA[2]. Infatti, il futuro del *mobile computing* riflette una visione più vasta e profonda che la percezione corrente di usare i dispositivi mobili per comunicazioni interpersonali e vari servizi, che rappresenta uno dei componenti del paradigma del mobile com-

[1] In informatica, l'espressione mobile computing (traducibile in italiano come "calcolo mobile") designa in modo generico le tecnologie di elaborazione o accesso ai dati (anche via Internet) prive di vincoli sulla posizione fisica dell'utente o delle apparecchiature coinvolte.

[2] Un computer palmare (detto anche palmare), spesso indicato in lingua inglese con l'acronimo PDA (Personal Digital Assistant), o con il termine palmtop, è un computer di dimensioni contenute, tali da essere portato sul palmo di una mano (da cui il nome), dotato di uno schermo tattile. Originariamente concepito come agenda elettronica (in inglese electronic organizer), o sistema non particolarmente evoluto dotato di un orologio, di una calcolatrice, di un calendario, di una rubrica dei contatti, di una lista di impegni/attività e della possibilità di memorizzare note e appunti (anche vocali) (personal information manager), è stato invece prodotto inizialmente da Apple come un vero minicomputer completo da portare in palmo di mano e si è nel corso degli anni arricchito di funzioni sempre più potenti ed avanzate.

puting. L'evoluzione rapida delle teconologie mobili wireless e il bisogno di accesso alle informazioni disponibili in qualsiasi orario e da ogni dispositivo, e per di più, sempre, ovvunque, da tutti i dispositivi, ha naturalmente posto diversi problemi riguardo alla tecnologia nel campo del mobile computing, aumentando il ruolo delle tecnologie mobili e dei telefoni cellulari nel futuro e seguito delle soluzioni individuate. Nel secondo capitolo si presenteranno alcune delle più significative soluzioni ai problemi del mobile nel contesto temporale.

Capitolo 2

Smartphone

Gli smartphone [4] sono dei dispositivi portatili che abbinano le funzionalità classiche del telefono cellulare a quelle della gestione di dati personali. Possono derivare dall'evoluzione di un PDA, cioè il classico palmare, cui è aggiunta la funzione di telefono e possono essere dotati di uno schermo tattile (touch screen). Gli smartphone sono stati originariamente concepiti come agenda elettronica, dotati di una calcolatrice, di un calendario, di una rubrica di contatti, di una lista di attività e della possibilità di memorizzare note e appunti, nel corso degli anni i PDA si sono arricchiti di funzioni sempre più potenti e avanzate, cui si aggiungono funzioni di telefono. Oggigiorno ogni telefono in commercio si avvicina a uno smartphone, cioè un terminale che permette di gestire file, navigare su Internet e acquisire applicativi anche di terze parti. Grazie a questi nuovi modelli di telefono cellulare è possibile ottenere un pacchetto di utilities svincolate dalla funzione di comunicazione classica dei vecchi terminali: fotocamera, bluetooth, UMTS[1], la possibilità di

[1] In telecomunicazioni lo *UMTS*, acronimo di Universal Mobile Telecommunications System (UMTS), è uno standard di telefonia mobile cellulare 3G (acronimo di 3rd Generation che indica le tecnologie e gli standard di terza generazione. I servizi abilitati dalle tecnologie di terza generazione consentono il trasferimento sia di dati "voce" (telefonate digitali) che di dati "non-voce" (ad esempio, download da internet, invio e ricezione di email ed instant messaging anche se la killer application utilizzata come traino dal marketing degli operatori 3G per l'acquisizione di nuova clientela è la videochiamata)).

leggere e spedire e-mail e accedere a social network (Facebook, Twitter, ecc.). Tutte queste caratteristiche inizialmente erano incorporate nei dispositivi di fascia più alta ma negli ultimi anni sono state rese disponibili anche a livelli di prezzo più ragionevoli. Tutti i telefoni in commercio sono sostanzialmente piccoli computer con un sistema operativo e funzionalità di comunicazione. Gli smartphone si distinguono per essere dotati di un sistema operativo più evoluto e aperto all'installazione di applicazioni addizionali da parte dell'utente. Questo significa che devono essere disponibili e facilmente accessibili strumenti di sviluppo per i programmatori che desiderano creare tali software, che le applicazioni devono aver accesso alle funzionalità del telefono e che deve esistere un modo per installare il software.

Il telefono cellulare viene, dunque, considerato il nuovo personal computer. Il computer desktop sta diventando meno importante [21] e il mercato degli smartphone è in rapida crescita. I telefoni cellulari vengono utilizzati come computer da più persone e per scopi diversi. Gli smartphone sono generalmente più economici dei computer, più convenienti grazie alla loro portabilità, e spesso più utili nel contesto fornito dalla geolocalizzazione. Attualmente, ci sono più cellulari che computer collegati a Internet. Mentre una minoranza di questi telefoni sarebbe considerato come smartphone, è in atto un cambiamento tale per cui i telefoni di fascia alta stanno diventano i cellulari di fascia bassa. Con i profitti da applicazioni in crescita, è probabile ci sia una continua evoluzione di hardware e sistemi operativi da parte dei produttori, mantenendo i nuovi telefoni economici o gratuiti.

Stiamo assistendo ad un cambiamento nel modo in cui le persone utilizzano i computer. Le applicazioni desktop che usiamo più frequentemente sono centrate attorno alla comunicazione, piuttosto che sulla creazione di documenti. Come consumatori, leggiamo le recensioni, inviamo brevi note agli amici e condividiamo foto. La posta elettronica è stata l'applicazione del ventesimo secolo. Sia nel mondo degli affari e sia nelle nostre vite personali, queste attività di comunicazione vengono arricchite efficacemente da applicazioni mobili. Ancora più importante, le nuove applicazioni incontreranno

le esigenze delle persone che non utilizzano un computer. Lo sviluppo del software si sposterà verso lo sviluppo mobile così come la maggioranza delle persone che utilizzano i computer lo faranno indirettamente attraverso un telefono cellulare. Il centro di gravità dell'industria del software sarà sempre più rivolto al mondo dei dispostivi mobili.

2.1 Il terminale di telefonia mobile universale

I telefoni cellulari e i PDA convergono entrambi a un singolo dispositivo mobile: smartphone. Ed è evidente che uno smartphone incorpora alcune funzioni dei dispositivi mobili e altri dispositivi elettronici del consumatore. In essenza, un smartphone diventerà un terminale mobile universale nel futuro del mobile computing. Questo perché, non è soltanto un dispositivo di comunicazione ma anche un dispositivo di elaborazione. Uno smartphone può

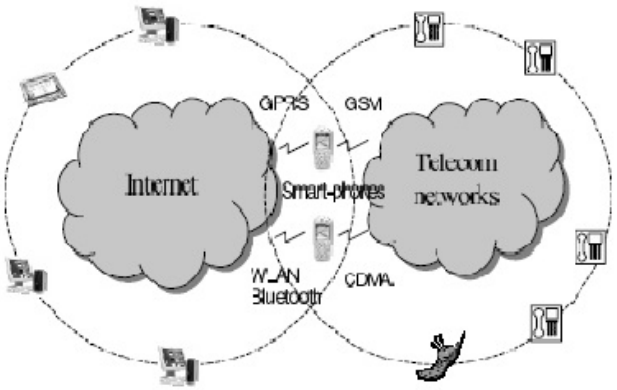

Figura 2.1: Smartphone, internet e reti di telecomunicazione

essere usato come un dispositivo di controllo wireless che permette il monitoring, il posizionamento e il controllo sui dispositivi elettronici del consumatore. Queste funzioni non sono semplicemente stipate in un smartphone usando componenti separate di hardware e software; loro sono integrate sistematicamente per permettere la condivisione dei dati e la loro interoperabilità.

Dal punto di vista dell'operatore che usa questi dispositivi, il successo degli smartphone dipende dal grado in cui questi possiedono:

- *Short message service* (SMS) [19]: assicura la spedizione e la ricezione dei messaggi di testo da un numero telefonico. Secondo lo standard GSM[2], circa 30 miliardi di messaggi SMS vengono spediti globalmente ogni mese, specialmente in Europa e in Asia.

- *Enhanced message service* (EMS): permette all'utente di spedire testo formattato, animazione, immagini e semplici melodie in un messaggio.

- *Multimedia message service* (MMS): si concentra di più sulle melodie polifoniche, imagini grandi, audio e vidio clip.

- *Global positioning system* (GPS): è lo standard 2G (seconda generazione) di telefonia mobile cellulare e attualmente il più diffuso del mondo.

- *Sistemi di navigazione con informazioni sul traffico e sistemi di informazioni geografici* (GIS): i servizi di locazione vengono facilitati dal GPS, la rete cellulare o altre technologie wireless come WiFi.

- *Messaggistica istantanea* (IM): è un servizio real-time dei testi di messaggi via server. Al contrario degli SMS, gli utenti possono entrare nelle sessioni di chat in gruppi diversi.

- *Personal Information Manager* (PIM)[3]: e-mail, calendari, rubrica, organizzatore e notepad sono applicazioni tipici del PIM. I servizi PIM basati sul web come **Yahoo!** permettono agli utenti di sincronizzare i

[2]In telecomunicazioni il *GSM*, acronimo di Global System for Mobile Communications (in principio la sigla significava "Groupe spécial mobile"), è lo standard 2G (seconda generazione) di telefonia mobile cellulare e attualmente il più diffuso del mondo: più di 3 miliardi di persone in 200 paesi usano telefoni cellulari GSM attraverso la rete cellulare.

[3]Un Personal Information Manager (abbreviato in *PIM*) è un software che permette di organizzare un certo tipo di informazioni personali per migliorare la produttività personale ed aziendale, come, ad esempio: E-mail, rubrica indirizzi, calendari, impegni e scadenze, programmazioni eventi, appunti vari

dati personali, inclusi gli e-mail, calendari personali, rubrica ecc., dal web al PDA oppure i telefoni cellulari.

- *Sincronizzazione dei dati con un computer oppure un altro dispositivo mobile*: le interfaccie di questa applicazione possono includere USB, Bluetooth, IrDA[4] oppure Wireless LAN.

- *Informazioni di varia natura*: news, meteo, film, giochi, blog, dati finanziari possono essere scaricati da un telefono cellulare.

2.2 Evoluzione degli Smartphone

Newton Message Pad

Newton [14] è il nome della famiglia di computer palmari prodotta da Apple. Nell'estate del 1993 Apple presentò il primo modello di palmare, il Newton Message Pad (OMP)[5] (Figura 2.2).

Questo prodotto, largamente in anticipo sui tempi e precursore dell'i-Phone, fu il primo tra i computer palmari per come vengono definiti oggi. La situazione finanziaria di Apple non era delle migliori e si stava cercando qualcosa di rivoluzionario per aumentare le vendite. Il Newton Message Pad fu il tentativo di Apple di creare un dispositivo tascabile che aiutasse gli utenti nel lavoro di tutti i giorni [15]. Fu praticamente il primo PDA messo sul mercato, anche se altre aziende erano già al lavoro su progetti simili. Al tempo dell'uscita del primo Newton Message Pad [16], Apple creò *ex novo* un

[4] *IrDA* (acronimo in lingua inglese per Infrared Data Association) è una organizzazione non profit di produttori elettronici, costituita nel 1994, che definisce le specifiche fisiche dei protocolli di comunicazione che fanno uso della radiazione infrarossa per la trasmissione wireless, a breve distanza, dei dati.

[5] *Il Newton Message Pad* è stato il primo computer palmare prodotto da Apple Computer e rappresenta il primo tentativo di realizzare un palmare dotato di una batteria con una discreta autonomia, con un sistema di riconoscimento della scrittura funzionale e con del software a corredo per la sincronizzazione con il computer funzionale. L'insieme di queste funzioni in un unico dispositivo ne facevano un vero e proprio "assistente personale"

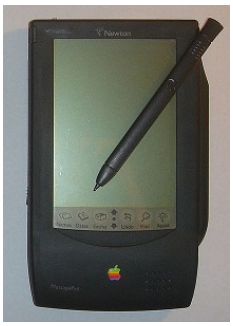

Figura 2.2: Newton Message Pad [16]

mercato, dato che il suo fu il primo palmare completo e commercializzato al pubblico, mentre gli altri "pseudo-palmari" esistenti prima del Newton erano considerati poco più che prototipi e/o esperimenti tecnologici, incompleti, pesanti, ingombranti e di difficile utilizzo. Il concetto stesso di "palmare" e il termine di personal digital assistant (PDA) fu coniato dall'amministratore delegato di Apple, John Sculley, nel 1992, in una conferenza stampa presso la fiera dell'informatica "Consumer Electronics Show"[6]

Il Newton fu equipaggiato, nella sua versione di "punta", con processore RISC[7] di ARM[8] denominato StrongArm[9] a 166 Mhz, con sistema in ROM[10] da 8 Mb, memoria RAM da 5 Mb espandibile, fornito di touchscreen (via

[6] *L'International Consumer Electronics Show*, comunemente abbreviato in International CES, è una fiera dell'elettronica di consumo allestita dalla Consumer Electronics Association (CEA) al Las Vegas Convention Center (Las Vegas, Nevada, USA).

[7] *RISC*, acronimo dell'inglese Reduced Instruction Set Computer, indica una filosofia di progettazione di architetture per microprocessori che predilige lo sviluppo di un'architettura semplice e lineare.

[8] L'architettura *ARM* (precedentemente Advanced RISC Machine, prima ancora Acorn RISC Machine) indica una famiglia di microprocessori RISC a 32-bit sviluppata da ARM Holdings e utilizzata in una moltitudine di sistemi embedded.

[9] Lo *StrongARM* è una famiglia di microprocessori sviluppati da Digital Equipment Corporation. Gli StrongARM sono dei processori basati sull'architettura ARM sviluppata dalla Advanced RISC Machines ma migliorati per offrire prestazioni superiori.

[10] *Read Only Memory* abbreviato con l'acronimo ROM, è una tipologia di memoria informatica non volatile in cui i dati sono memorizzati nella sua fase di costruzione e non possono essere modificati successivamente.

stilo), audio con output stereo a 16 bit e input mono, video 480x320 a 16 bit, porta seriale, PCMCIA (cardbus)[11], e connettività ad infrarossi. Il Newton fu un grande prodotto, ma ebbe il difetto di precorrere troppo i tempi. Il Newton Message Pad era relativamente compatto e pesava "solo" 400 grammi. L'accoglienza della stampa fu buona, il Newton prometteva la facilità d'uso tipica dei prodotti Apple combinata con un sistema di riconoscimento della scrittura e persino riconoscimento vocale (anche se solo in inglese e solo per alcuni modelli). Era uno strumento avanzato e dotato di una serie di programmi che gli permettevano di integrarsi con Mac OS. Fin dal primo modello era integrato un sistema operativo (Newton OS) con un'interfaccia estremamente essenziale, basata sull'uso di un pennino sullo schermo monocromatico e sensibile al tocco. Erano inoltre presenti la rubrica indirizzi, il calendario e un blocco note ed esisteva la possibilità di far dialogare queste applicazioni con fax e email.

Purtroppo Newton OS si avvaleva di un riconoscimento della scrittura che non funzionava molto bene. Ciò procurò una pubblicità negativa al prodotto che, nonostante tutto, continuò ad avere un successo non trascurabile. Fu venduto con un prezzo tra 999 e 799 dollari, quindi non economico, ma essendo stato progettato per i manager non aveva nemmeno un costo proibitivo. Il riconoscimento della scrittura, una delle caratteristiche più reclamizzate, era lacunoso. Nonostante la casa produttrice lo pubblicizzasse come facile da usare, era difficile da usare. Inoltre, l'utente non era mai sicuro di diventare padrone del suo uso. Sebbene le serie successive del Newton migliorassero notevolmente il sistema di riconoscimento della scrittura, il cattivo nome che si era fatto agli esordi lo seguì fino alla fine del ciclo di produzione. Apple realizzò nel corso degli anni otto modelli diversi della serie Newton, ma nessuno di essi riuscì mai ad avere successo, anzi molti di questi riportarono ingenti perdite.

[11] *PCMCIA*, è uno standard d'interfaccia elettronica, destinata a computer portatili, o altri tipi di dispositivi elettronici portatili, e accessibile dall'esterno, sviluppata dalla Personal Computer Memory Card International Association (in sigla PCMCIA) al fine di permettere l'espansione delle funzionalità di tali dispositivi portatili.

Nell'anno 1994 Apple rilasciò tre modelli:

- il Message Pad 100, che aveva solamente una versione aggiornata del sistema operativo e del software di riconoscimento della scrittura;
- il Message Pad 110, più potente dei due modelli precedenti, sfoggiava un case completamente ridisegnato;
- il Message Pad 120, presentato verso fine anno, aggiornava l'hardware del modello 110 con il sistema operativo Newton OS 2.0.

L'ultimo modello con il nuovo sistema operativo aveva un'interfaccia completamente rinnovata, la possibilità di fare backup su Mac e PC e includeva un sistema di riconoscimento della scrittura migliorato. I grossi investimenti necessari per lo sviluppo dei Newton avevano finalmente prodotto un modello completo e appetibile per il mercato professionale. Per tutto l'anno seguente non vi furono novità sui PDA di Apple. Il modello successivo arrivò solo nel 1996. Il Message Pad 130 che uscì nel 1996, non era molto diverso dai precedenti, ma aveva un monitor retroilluminato e una batteria più performante. Fino all'anno successivo non ci furono grandi novità, probabilmente anche per il periodo di grave crisi finanziaria che Apple stava passando.

Figura 2.3: Apple Newton Message Pad 2000 [62]

Nel marzo del 1997, finalmente, uscì il Message Pad 2000 (Figura 2.3), dieci volte più veloce dei precedenti. Aveva uno schermo più grande, con

maggiore risoluzione e capace di 16 livelli di scala di grigio. Al prezzo di 799 dollari, integrava una porta per la connessione a vari tipi di computer e dispositivi ed era compatibile con le schede PCMCIA. Nonostante il brutto periodo, Apple tentò un rilancio delle vendite con la presentazione di un dispositivo basato su Newton e pensato per gli studenti. Dal design assolutamente sperimentale, fu chiamato eMate 300.

eMate 300 (Figura 2.4) aveva una forma a conchiglia e un case traslucido (in un certo senso molto simile ai primi iBook che sarebbero arrivati sul mercato solo nel 1999). Dal punto di vista delle caratteristiche tecniche era uguale al Message Pad 2000.

Figura 2.4: Apple Newton Emate 300 [63]

Fu venduto solo nel settore educativo a 799 dollari. Verso la fine dello stesso anno fu rilasciato il modello 2001 con più RAM a disposizione e venduto a 999 dollari; sarà l'ultimo modello messo in circolazione. L'anno seguente, infatti, cominciò il grande cambiamento di Apple ad opera di Steve Jobs. Apple doveva ricominciare a crescere, convogliando le sue energie in pochi prodotti e tagliando le spese superflue. Lo sviluppo dei Newton costava veramente troppo e rendeva poco perchè, anche se apprezzato, rimaneva un prodotto troppo complicato. Steve Jobs non aveva mai creduto particolarmente nel Newton e decise così di chiudere il progetto nonostante le proteste degli appassionati. Dopo 5 anni e più di 500 milioni di dollari spesi in ricerca, Apple decise di eliminare il progetto Newton e di sciogliere la divisone che se ne occupava. Fino a oggi i PDA presenti sul mercato sono stati molti, ma nessuno ha raggiunto la semplicità e l'immediatezza d'uso dei Newton. Fino a oggi, appunto, perchè siamo alle soglie del ritorno di Newton in iPhone.

Chi conosce i Newton, infatti, ha subito pensato a loro quando ha visto la presentazione di iPhone. È ormai chiaro che l'idea alla base del nuovo smartphone/iPod di Apple sia una versione attuale e ampliata di quella del primo Newton.

In conclusione, si può ritenere che tutta l'esperienza accumulata nello sviluppo di Newton non sia stata persa, anzi, è servita per il lancio dell'iPhone che grazie al suo (quasi scontato) successo, ha aiutato Apple a rilanciare le vendite, promuovendo così il suo marchio in tutto il mondo.

Simon

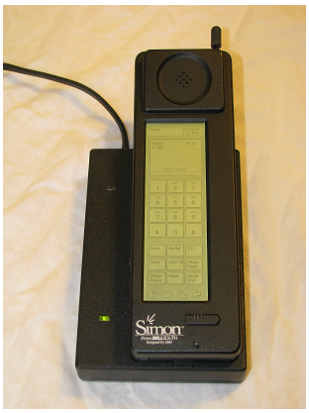

Figura 2.5: Primo smartphone Simon [5]

Il primo smartphone apparso sul mercato era Simon (Figura 2.5), progettato dalla IBM nel 1992 [5] e presentato come prototipo quello stesso anno al COMDEX, la fiera del commercio informatico tenutosi a Las Vegas, Nevada. È stato poi messo sul mercato nell'anno 1993 e venduto da BellSouth. Oltre ad essere un telefono cellulare, conteneva anche un calendario, rubrica, orologio, calcolatrice, blocco note, e-mail, invio e ricezione fax, giochi, ecc. Simon già all'epoca aveva un grande vantaggio tecnologico essendo dotato di monitor touch screen con la possibilità di digitare i numeri con le dita o con una stilo inclusa nella confezione. Considerando gli standard odierni, la qualità di Simon era piuttosto bassa, ma per l'epoca rappresentava un gioiello della tecnologia.

Nokia Communicator

Figura 2.6: Nokia 9000 Communicator [6]

La linea Communicator di Nokia [6] (Figura 2.6) è la prima della classe smartphone di Nokia iniziata nel 1996 con il modello Nokia 9000.

Questo smartphone è il risultato di una fusione tra uno dei primi modelli PDA della Hewlett Packard[12] (HP) in combinazione con uno dei migliori telefoni Nokia di quel tempo. Il Nokia 9210 (Figura 2.7) è stato probabilmente il primo vero smartphone con un sistema operativo.

Figura 2.7: Nokia 9210 [6]

Nel mese di ottobre 2001, Handspring[13] annunciò il Palm OS smartphone Treo [18] (Figura 2.8), il quale utilizzava una tastiera completa che combi-

[12] *La Hewlett-Packard Company* (nota anche con la sola sigla HP) è una multinazionale statunitense dell'informatica attiva sia nel mercato dell'hardware (dai personal computer ai server e, nel mercato di massa, per le stampanti per le quali è uno dei maggiori produttori mondiali) quanto in quello del software e dei servizi collegati all'informatica.

[13] La compagnia *Handspring* era un produttore di Palm OS fondata nel giugno 1998 e gestito da Jeff Hawkins, Donna Dubinsky, e Ed Colligan (inventori del Palm Pilot e

nava la navigazione web senza fili, e-mail, calendario e rubrica, utilizzando applicazioni di terze parti che potevano essere scaricate e sincronizzate con un computer.

Figura 2.8: Palm Treo [18]

Blackberry

BlackBerry (letteralmente "mora" in inglese) [39] è il nome commerciale di una serie di dispositivi portatili smartphone (i primi modelli, come l'850, erano cercapersone con funzionalità aggiuntive di messaggistica ed agenda) prodotti dalla società canadese Research In Motion (RIM)[14] fondata dall'ingegnere Mihalis Lazarides.

Nell'anno 2002, la RIM ha immesso sul mercato il primo modello di BlackBerry 8707g (Figura 2.9), il primo smartphone ottimizzato per l'utilizzo wireless che raggiunse una base totale di 8 millioni di clienti, di cui tre quarti nel solo Nord America, nel giugno 2007. BlackBerry è divenuto nel tempo sinonimo e simbolo della push-email (posta elettronica in tempo reale) sui dispositivi mobili. Nel 2006 RIM comprende le potenzialità offerte da questo "strumento di lavoro" e soprattutto l'appettibilità che l'email in tem-

fondatori di Palm Computing). Handspring si fuse con la divisione hardware di Palm Inc. nel 2003 per formare palmOne.

[14]Research In Motion Limited (RIM) è un'azienda canadese produttrice di dispositivi wireless con sede a Waterloo in Ontario. È principalmente conosciuta come la sviluppatrice del terminale BlackBerry.

Figura 2.9: Primo modello Blackberry 8707g [64]

po reale ha sul mercato dell'utenza privata e concepisce il primo "cellulare" BlackBerry con caratteristiche multimediali: il BlackBerry 8100 Pearl con fotocamera. Il resto è storia attuale [40]. Con l'uscita del nuovo sistema operativo Blackberry 7, fulcro dell'innovazione è indubbiamente la velocità di navigazione web che la versione 7 offre; non si trascurano altri aspetti migliorati rispetto alle precedenti versioni come l'ottimizzazione dello zoom e il panning per una navigazione migliore completando il quadro, andando così a migliorare le performances anche dal punto di vista della fruizione di videogames in HTML5[15] e video in generale. All'interno di BlackBerry 7 (e quindi dei nuovi smartphone BlackBerry Bold 9900, 9930, BlackBerry Torch 9810, BlackBerry Torch 9850 e 9860 all-touch) l'utente trova precaricate una serie di applicazioni, tra le quali la versione Premium di Documents To Go, utilissima per lavoro [41]. Spazio anche a BlackBerry Messenger e alla versione 2.0 di Facebook per BlackBerry, per gestire comunicazione e socialità facilmente e in maniera molto fluida. La funzione di ricerca vocale è stata molto potenziata ed affinata, consentendo così agli utilizzatori di poter fare ricerche sul web in piena mobilità, senza neppure digitare la parola chiave di interesse: basta pronunciarla e il "dettato" fa partire il search online, senza dover necessariamente utilizzare le mani. A oggi nonostante l'uscita

[15] L'*HTML5* è un linguaggio di markup per la progettazione delle pagine web attualmente in fase di definizione (draft) presso il World Wide Web Consortium.

sul mercato dei modelli prodotti dalle varie imprese operanti nel settore, il BlackBerry può essere considerato il dispositivo smartphone per eccellenza.

iPhone

Il dispositivo mobile iPhone è stato presentato al pubblico dalla Apple Inc. nel corso dell'anno 2007 (Figura 2.10).

Figura 2.10: Apple Iphone GSM Quad Band [66]

Il primo dispositivo iPhone [66] è stato presentato da Steve Jobs, amministratore delegato della società durante la conferenza di apertura del Macworld del gennaio 2007. Il dispositivo è comparso nei negozi Apple e quelli Cingular/AT&T negli USA dal 29 giugno dello stesso anno. Il primo modello, distribuito nel 2007, era un GSM EDGE quad-band mentre le versioni successive hanno adottato la tecnologia UMTS e HSDPA[16]. L'iPhone include una fotocamera digitale, un dispositivo Assisted GPS[17] e un lettore multimediale (le funzioni di UMTS e AGPS sono state inserite solo nelle versioni 3G e 3GS). Il dispositivo, oltre ai normali servizi di telefonia quali chiamate SMS

[16] *High Speed Downlink Packet Access* (HSDPA), è un protocollo, appartenente alla famiglia di protocolli HSPA, introdotto nello standard UMTS per migliorarne le prestazioni in download, ampliandone la larghezza di banda, ed aumentando così la capacità di trasmissione delle reti radiomobili cellulari che, in download, può raggiungere la velocità massima teorica di 14,4 Mb/s.

[17] *Assisted GPS (A-GPS)*, è un sistema che consente di abbattere i tempi necessari alla prima localizzazione durante l'uso di un terminale GPS.

ed MMS, permette di utilizzare servizi come e-mail, navigazione web, Visual Voicemail e può gestire una connessione Wi-Fi. Viene controllato dall'utente tramite uno schermo multi-touch, un sensore di movimento del dispositivo (accelerometro), una tastiera virtuale, un pulsante per tornare al menu principale, due piccoli tasti per la regolazione del volume, uno per passare dallo stato di suoneria allo stato di vibrazione ed uno per lo standby/spegnimento.

- Il 9 giugno 2008 la Apple Inc. ha annunciato l'uscita dell'iPhone 3G per l'11 luglio 2008 in 70 paesi tra cui l'Italia.
- L'8 giugno 2009 la Apple Inc. ha presentato una nuova versione dell'iPhone 3G modificata e aggiornata chiamata iPhone 3GS; questo modello è stato lanciato il 19 giugno 2009.
- Il 7 giugno 2010 la Apple Inc. ha presentato il nuovo iPhone 4.
- Il 4 ottobre 2011 la Apple Inc. ha presentato una versione modificata e aggiornata dell'iPhone 4 chiamta l'iPhone 4S.

Il dispositivo iPhone incorpora le funzioni di tre dispositivi che hanno iniziato ad essere integrati tra loro con l'avvento degli smartphone:

1. un iPod con capacità di riproduzione audio, foto e video;
2. un telefono cellulare quad-band (GSM + UMTS HSDPA + EDGE) con connettività Wi-Fi e Bluetooth 2.0 (solo tra dispositivi iphone) + EDR[18] e dotato di fotocamera da 2.0 megapixel su iPhone 2G e 3G[19], da 3.2 megapixel su 3GS e da 5.0 megapixel su iPhone 4.
3. un palmare di nuova concezione con sistema operativo derivato da Mac OS X con funzioni integrate di navigazione in internet (tramite il browser Safari), multimedialità e visione di video (tramite l'applicazione YouTube) e servizio GPS (tramite l'applicazione Google Maps).

[18] *Enhanced Data Rate (EDR)*: porta la velocità di trasmissione fino a 3 Mb/s

[19] Il termine *3*G (acronimo di 3rd Generation) indica le tecnologie e gli standard di terza generazione.

Le funzionalità relative alla telefonia sono quelle classiche di uno smartphone [42], e comprendono anche la possibilità di effettuare un'audioconferenza unendo più telefonate (durante la presentazione del prodotto, Steve Jobs ha utilizzato una conferenza a tre, ma il telefono supporta fino a 5 partecipanti, lasciando comunque disponibile una seconda linea per le chiamate in entrata). Gli SMS vengono gestiti con un sistema molto simile all'interfaccia grafica utilizzata da iChat[20], che permette di visualizzare la cronologia dei messaggi SMS scambiati organizzandoli in discussioni, in modo simile a quello che avviene nei forum su internet.

La nuova versione di iPhone, chiamato iPhone 4S, fornisce una nuova batteria [43], che garantisce 8 ore in chiamata 3G, 6 ore di navigazione in 3G, 9 ore di navigazione in WiFi e 10 ore di riproduzione video ore di riproduzione audio. Grazie al processore dual-core Apple A5 ad 1 Ghz di Frequenza, le prestazioni grafiche dell'iPhone aumenteranno di 7 volte rispetto all'attuale modello in commercio. L'assistente vocale è una funzione rivoluzionaria con la quale è possibile parlare con il proprio iPhone: si possono gestire appuntamenti, chiedere che tempo fa, sapere l'ora di altre città, farci leggere la borsa, dettare un SMS, farci leggere un messaggio, e tanto altro. Queste caratteristiche e tante altre rendono l'iPhone lo smartphone più rivoluzionario nei giorni d'oggi.

[20] *iChat* è un programma di chat sviluppato da Apple Inc. compatibile col protocollo di AOL Instant Messenger (che è un programma di instant messaging prodotto da AOL), ICQ (che è un programma per computer di instant messaging nel mondo), Google Talk, Bonjour e altri. iChat è disponibile solo per Mac OS X e utilizza un'interfaccia metallizzata in linea con Mac OS X 10.5 Leopard e usa dei riquadri a fumetti per racchiudere il testo della chat. I personaggi che chattano hanno associata un'immagine che se disponibile viene prelevata dalla Rubrica Indirizzi. L'obiettivo di iChat è di rendere quanto più personale possibile la chat tra utenti.

Sistema operativo	Linguaggio di programmazione
Symbian OS	C++
Blackberry	Java
Apple iPhone	Objective-C
Windows Mobile	C
Google Android	Java
PalmwebOS	Javascript

Tabella 2.1: Linguaggi di programmazione usati dai sistemi operativi per i dispositivi mobili

2.3 Differenze tra Smartphone

I telefoni cellulari di oggi sono generalmente suddivisi tra quelli di fascia inferiore *"feature phone"* e quelli di fascia superiore *"smartphone"*. Uno smartphone ha una tastiera QWERTY (una tastiera fisica oppure una tastiera su touch screen come iPhone o BlackBerry Storm) ed è più potente rispetto alle caratteristiche di un telefono con schermo grande ad alta risoluzione oltre ad avere più funzionalità dei dispositivi. Gli smartphone hanno un insieme eterogeneo di sistemi operativi (Tabella 2.1). Quando si sviluppa un'applicazione, come Microsoft Word o Adobe PhotoShop, gli sviluppatori delle applicazioni creano le loro applicazioni principali in un linguaggio di alto livello come C++ e condividono quel codice di base tra le varie piattaforme. Poi utilizzano le API[21] per accedere al filesystem e sviluppare l'interfaccia utente. Nel 1990, un certo numero di framework per i desktop di multipiattaforma è emerso, rendendo più facile per le aziende lo sviluppo di un

[21]Un *Application Programming Interface* (API) in italiano Interfaccia di Programmazione di un'Applicazione si indica ogni insieme di procedure disponibili ad un programmatore, che possono essere riutilizzate al momento dello sviluppo di un'applicazione. L'utilizzo delle API è quindi un metodo per ottenere un'astrazione tra hardware e programmatore, o tra software a basso ed alto livello, che permette agli sviluppatori di non dover riscrivere per ogni applicativo tutte le funzioni di base e di riprogettare o migliorare le funzioni all'interno dell'API senza cambiare il codice che si affida ad essa.

singolo codice di base che poteva essere compilato per ogni piattaforma di destinazione (tipicamente, Mac o Windows).

Considerando solo gli smartphone, i sistemi operativi principali che occupano oltre il 90% del mercato sono: Google Android, Apple iPhone, BlackBerry e Windows Mobile. Il resto del mercato lo detengono i sistemi operativi Symbian e Palm webOS. Per la maggior parte di questi sistemi operativi, vi è un linguaggio di sviluppo nativo, che è necessario per sviluppare in modo ottimale per questo piattaforma, come illustrato nella Tabella 2.1. Mentre è possibile sviluppare usando altri linguaggi, di solito ci sono svantaggi o limitazioni nel farlo. Ad esempio, si puo essere in grado di sviluppare un'applicazione Java per Symbian, ma diverse API native non sono disponibili per accedere alle capacità del dispositivo. Oltre alle differenze dei linguaggi di programmazione utilizzate, il kit di sviluppo software (SDK[22]) e i paradigmi per lo sviluppo di applicazioni sono differenti su ogni piattaforma. Se le funzionalità dei dispositivi sono quasi identiche, come la geolocalizzazione, la macchina fotografica, l'accesso ai contatti, e di archiviazione offline, le API specifiche per accedere a queste funzionalità sono diverse su ogni piattaforma.

2.4 Il mercato delle applicazioni

Alla fine degli anni '90, lo sviluppo di applicazioni per dispositivi mobili è stato considerato uno dei mercati più ricchi. Mentre ci sono stati diversi sviluppatori di applicazioni indipendenti e la maggior parte dei telefoni di fascia alta supportava l'installazione di applicazioni, il processo dell'installazione delle applicazioni è stato difficile e la maggior parte degli utenti finali non potevano aggiungere applicazioni ai loro telefoni. L'iPhone ha rivitalizzato il mondo della tecnologia per lo sviluppo di applicazioni mobili. Apple ha creato un'interfaccia molto facile da usare per l'acquisto e l'installazione

[22] *Software Development Kit* (più brevemente SDK) è un termine che in italiano si può tradurre come "pacchetto di sviluppo per applicazioni", che permette la creazione di applicazioni per un pacchetto software, piattaforma hardware, sistema operativo ecc.

di applicazioni di terze parti, e più importante, ha promosso quella capacità ai loro utenti e potenziali clienti. I sistemi operativi smartphone stanno al passo con i progressi hardware fatti e la facilità di sviluppo con strumenti migliorati. Nell'iPhone App Store, spesso le innovazioni più significative non sono puramente tecniche. L'App Store ha ridotto gli ostacoli allo sviluppo di applicazioni, fornendo un accesso facile alla distribuzione. Le persone sviluppano più applicazioni quando vi è un mercato accessibile e un canale di distribuzione vasto. L'App Market di Google, l'App World di Blackberry, e Windows Marketplace per i Mobile guidano il successo di applicazioni esistenti per i sistemi operativi dei dispositivi mobili. Nella Figura 2.12 viene visualizzata la situazione attuale delle app store che varia a seconda delle piattaforme usate, secondo Developer Economics 2011 [36].

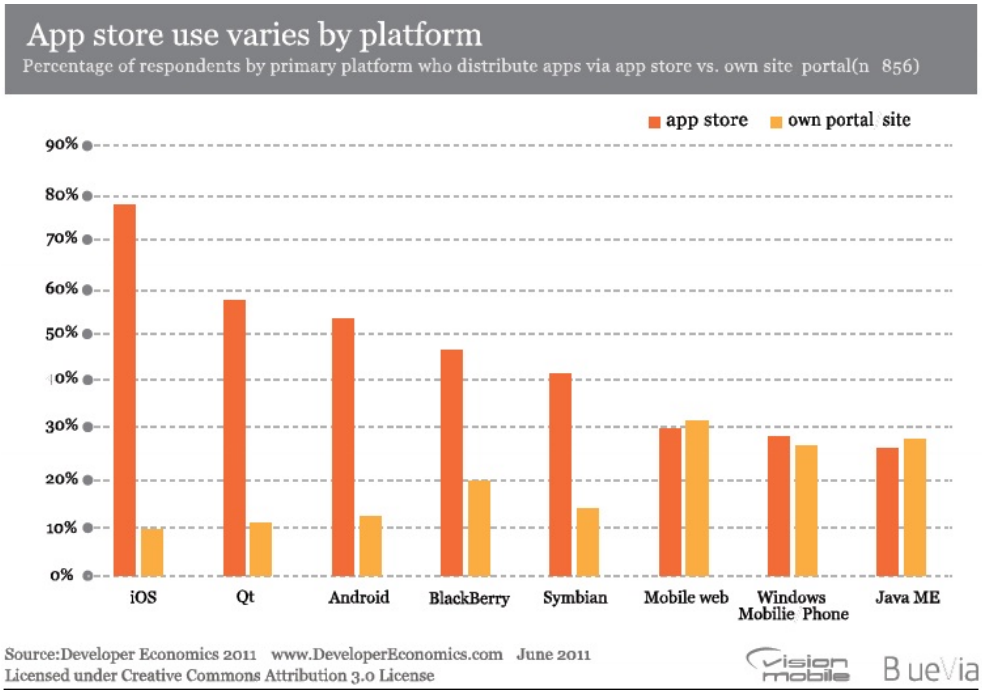

Figura 2.11: L'utilizzo degli App Store varia a seconda delle piattaforme usate [36]

- **App Store** (Figura 2.11)

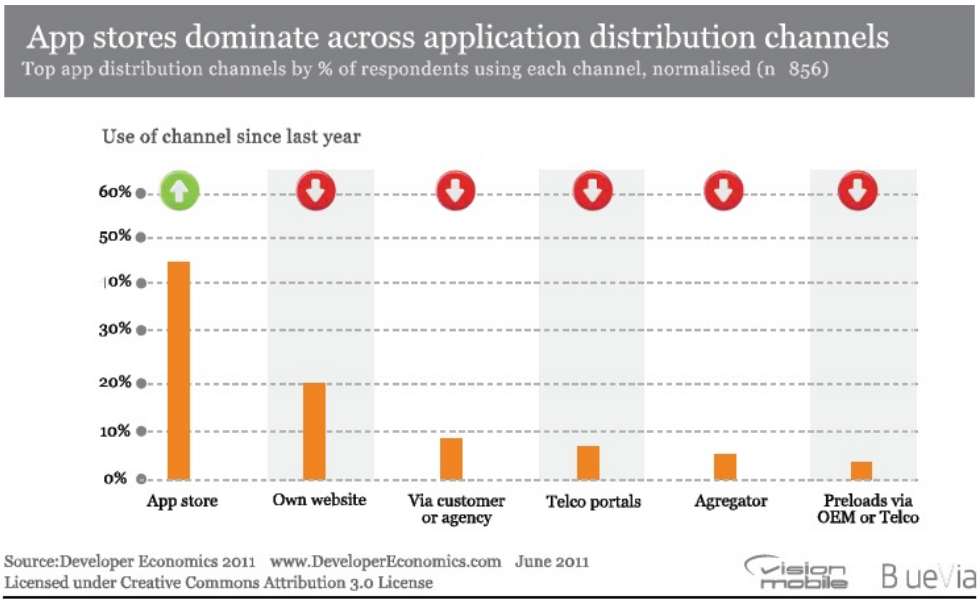

Figura 2.12: Gli App Store dominano tra i canali di distribuzione delle applicazioni [36]

Il mercato delle applicazioni gestibile con l'Apple App Store rende facile trovare, provare e acquistare l'applicazione che meglio si adatta alle esigenze degli utenti. Come si fa a decidere quali mercati hanno gli strumenti giusti per i propri utenti? I mercati delle applicazioni sono in genere costruiti da un marchio leader con accesso a milioni di clienti e, soprattutto intorno ai dati di business e processi che sono fondamentali per molte aziende. Per esempio, il Google Apps Marketplace fornisce applicazioni che si applicano alle applicazioni di Google come email, calendario e la collaborazione sui documenti. Nel frattempo, Intuit[23] ha un app store per la gestione dei dati finanziari con l'integrazione di QuickBooks[24].

[23]Intuit è un azienda americana di software per la gestione finanziaria e fiscale delle piccole aziende o famiglie.

[24]QuickBooks è una linea di software di contabilità aziendale sviluppato e commercializzato da Intuit nel 2002

Identificare quali sono i dati degli affari che saranno toccati da queste applicazioni è importante quanto scegliere il mercato giusto. Introduciamo il mercato delle applicazioni di Google chiamato Google Apps Marketplace.

- **Il mercato delle applicazioni di Google**

  Se i processi di business di un'azienda dipendono fortemente dalle applicazioni di Google come Gmail, Google Docs o Google Calendar, il Google Apps Marketplace è un posto ideale per trovare applicazioni utili per le funzioni giorno per giorno. Google ha lanciato sul mercato la sua applicazione nel marzo 2010, e fornisce applicazioni che sono progettate per funzionare in congiunzione con le email e Microsoft Office come i documenti archiviati in Gmail e Google Docs.

  – *Intuit App*

    Intuit fornisce un'applicazione che si concentra sulla organizzazione dei dati finanziari con particolare attenzione alla integrazione Quickbooks. Ragionieri, commercialisti, e titolari di aziende che lavorano a stretto contatto con Quickbooks troveranno che la Intuit App Center offre molte applicazioni utili per sfruttare i dati finanziari. Attualmente è in esecuzione un'applicazione chiamata "Get Organized".

  – *SalesForce*

    Il SalesForce è un'applicazione di scambio che è stato progettato per sfruttare i dati di marketing memorizzati nell'applicazione originale Salesforce. Salesforce può essere usato per progettare sondaggi, campagne di email marketing, e gestire contatti e clienti. Con l'aggiunta di applicazioni di terze parti attraverso lo scambio, questa applicazione è possibile integrarla con le reti come Facebook, Twitter e LinkedIn.

  – *Oneforty*

Oneforty è un mercato di applicazioni per il popolarissimo servizio Twitter. Se si usa Twitter per il networking e il marketing, Oneforty permette di portare un'applicazione rendendo questo processo più semplice ed efficace. Essi offrono anche applicazioni in forma di kit di strumenti, progettati per scopi specifici.

– *Facebook*

Facebook ha un app store che non contiene solo giochi e condivisione di foto. Se un'attivitità viene commercializzata con una pagina di Facebook, il negozio Facebook app è in grado di fornire sondaggi, strumenti di gestione degli eventi e video per aumentare l'interazione e sfruttare la rete sociale esistente.

– *Apple*

App store iPod, iPhone, e iPad non sono noti per le loro applicazioni di business, però hanno numerose applicazioni che possono essere molto utili in questo campo.

2.5 Incremento dell'utilizzo dei dispositivi mobili

Secondo un Rapporto delle Nazioni Unite nell'anno 2009, sei su dieci persone in tutto il mondo hanno abbonamenti per un cellulare [10], che supera il quarto della popolazione mondiale con un computer a casa. Gli smartphone sono ancora una frazione di telefoni cellulari, ma la crescita è forte e i numeri sono particolarmente interessanti se paragonati ai computer in vendita. La Mobile Handset DesignLine [11] riporta che gli smartphone rappresentano il 14% delle vendite globali dei dispositivi, ma le proiezioni di Gartner[25] [65] notano che le spedizioni smartphone hanno superato le unità di vendite

[25] *Gartner Inc.* è una società multinazionale leader mondiale nella consulenza strategica, ricerca e analisi nel campo dell'Information Technology con oltre 60.000 clienti nel mondo. L'attività principale consiste nel supportare le decisioni di investimento dei suoi clienti attraverso ricerca, consulenza, benchmarking, eventi e notizie. L'azienda è stata fondata

di notebook nel 2009 e che nell'anno 2011, la vendita degli smartphone ha avuto una crescita di 19% [12]. Guardando a come le persone utilizzano i loro telefoni cellulari suggerisce modelli di comportamento che guideranno le vendite dei smartphone nel futuro. Sempre più spesso, la gente sta usando i telefoni non solo per le telefonate: la navigazione web e l'utilizzo di altre applicazioni mobili sono sempre in crescita. Rapporti di mercato comScore[26] [13] informano che l'utilizzo globale di Internet mobile sta crescendo sempre di più. Lo studio ha esaminato più di 30.000 abbonati di dispositivi mobile negli Stati Uniti e ha individuato *Samsung* come il produttore di cellulari con 25,3% del mercato. Google Android ha preso il comando tra le piattaforme smartphone con 40,1% del mercato, posizionandosi al primo posto.

Top Smartphone Platforms
3 Month Avg. Ending Jun. 2011 vs. 3 Month Avg. Ending Mar. 2011
Total U.S. Smartphone Subscribers Ages 13+
Source: comScore MobiLens

	Share (%) of Smartphone Subscribers		
	Mar-11	Jun-11	Point Change
Total Smartphone Subscribers	*100.0%*	*100.0%*	*N/A*
Google	34.7%	40.1%	5.4
Apple	25.5%	26.6%	1.1
RIM	27.1%	23.4%	-3.7
Microsoft	7.5%	5.8%	-1.7
Symbian	2.3%	2.0%	-0.3

Figura 2.13: Statistiche di comScore sulle piattaforme smartphone [13]

In Africa, un recente aumento di adozione cellulare è attribuita all'uso di telefoni per il settore bancario e l'invio di denaro ai parenti tramite la messaggistica di testo.

Anche i telefonini di fascia bassa di solito permettono navigazione in rete, e-mail e testo messaggistica, ma la potenza degli smartphone consente una più ampia gamma di applicazioni. Gli smartphone non sono solo piccoli

nel 1979 da Gideon Gartner e nel corso degli anni si è espansa fino ad acquisire altre 30 aziende come Real Decisions, MetaGroup, AMR Research, Burton Group.

[26] *comScore* è una società di ricerca marketing via internet che fornisce servizi e dati per il marketing in molti settori commerciali di Internet. comScore tiene traccia di tutti i dati internet sui suoi computer per studiare il comportamento della "rete". È stata fondata nell'agosto 1999 a Reston da Gian Fulgoni e Magid Abraham.

computer che possono stare in una tasca. Per molte applicazioni, in realtà sono dispositivi più potenti di un computer portatile grazie alle loro funzionalità integrate di fotocamera, connessione e geolocalizzazione. Grazie alla lunga durata della batteria e alla portabilità di questi dispositivi, le persone scelgono proprio i dispositivi smartphone per le loro attività.

Immediatezza, semplicità, e contesto. Quando queste tre caratteristiche sono combinate con l'utilità, emergono diverse applicazioni software che trasformeranno il modo di utilizzare i telefoni cellulari.

Capitolo 3

Sistemi operativi per i dispositivi mobili

A partire dai primi anni '90 le tendenze di mercato hanno suggerito, alle società produttrici di dispositivi mobili, l'opportunità di realizzare una maggiore convergenza di funzionalità e caratteristiche tra dispositivi cellulari e computer palmari. Le maggiori case produttrici, nel tentativo di assicurarsi quote di un mercato in continua crescita, hanno dato vita ad un'evoluzione tecnologica che in due decenni ha reso possibile il potenziamento dei loro prodotti. Attualmente i computer palmari o PDA possono essere dotati di dispositivi di input quali: tastiere, touch screen e trackball (come nel caso dei Blackberry), connettività wired e wireless, slot per l'espansione della memoria, fotocamere e ricevitori GPS. Le funzionalità di cui possono disporre sono l'accesso al web, alle e-mail, la gestione dell'agenda elettronica, la riproduzione di file audio e video. Possono ospitare applicazioni di uso medico. I problemi affrontati dalle case produttrici possono essere legati a molteplici aspetti. Si pensi, ad esempio, alla necessità di avere dispositivi di dimensioni ridotte e contemporaneamente all'esigenza di garantire la fruibilità delle pagine web o delle riproduzioni video, o ancora, si pensi alla necessità di ottenere elevate capacità elaborative e, al tempo stesso, l'esigenza di limitare i consumi energetici per garantire una sufficiente autonomia

energetica. La crescente importanza del mercato dei dispositivi mobili ha avviato una serrata competizione nello sviluppo dei sistemi operativi e del software applicativo. Tra i colossi che si dividono gran parte del mercato sono presenti: Nokia, Microsoft, Apple, RIM e più recentemente Google. I diversi attori del mercato mondiale hanno operato differenti scelte commerciali e tecnologiche riguardanti la gestione delle risorse hardware. Nel seguito del capitolo vengono presentati sinteticamente i sistemi operativi di cui sono dotati la maggior parte dei dispositivi mobili attualmente prodotti: Symbian, Blackberry, iPhone OS, Windows Phone, Android.

3.1 Introduzione ai sistemi operativi per i dispositivi mobili

I dispositivi mobili sono sistemi di elaborazione tipicamente dotati, se paragonati ai dispositivi fissi, di ridotte capacità elaborative e di memoria. Tra di essi è possibile identificare:

- PDA (Personal Digital Assistant) - computer palmari;
- smartphone - dispositivi portatili che abbinano funzionalità comuni a quelle dei computer palmari a funzionalità di telefonia;
- dispositivi special-purpose - digital camera, barcode reader, microcontrollori utilizzati nei sistemi di controllo.

Questo lavoro di tesi fornisce una panoramica ai sistemi operativi per dispositivi mobili ed in particolare alle prime due categorie di dispositivi mobili identificati: PDA e smartphone. Cominciamo con l'osservare le due notevoli limitazioni che affliggono questi sistemi, rispetto ai sistemi fissi:

- scarsa disponibilità di capacità elaborative;
- scarsa durata delle batterie.

Gli aspetti costruttivi che caratterizzano i dispositivi fisici offrono una prima occasione per ottimizzare la gestione delle risorse. Ci concentriamo, ad esempio, sul vasto utilizzo delle memorie statiche. Attraverso questo tipo di memoria è possibile evitare il costoso (dal punto di vista dei consumi elettrici) meccanismo del refresh necessario per le memorie di tipo dinamico. O ancora, il recente utilizzo di memorie PSRAM[1] presenti nei dispositivi Apple di ultima generazione, che offrono tutti i vantaggi delle memorie dinamiche in termini di dimensioni, ma allo stesso tempo, presentano una valida soluzione per arginare il costo del refresh. Passando all'analisi dei sistemi operativi, si osservi che, per questo tipo di dispositivo non è possibile utilizzare alcune delle comuni soluzioni adottate nei sistemi operativi per i comuni sistemi desktop. La maggior parte dei sistemi operativi dispone di funzionalità di multithreading, multitasking e protezione della memoria. I produttori prestano grande importanza a tutto ciò che riguarda l'utilizzo efficiente della memoria. Symbian, ad esempio, offre tecniche specifiche che determinano gli errori dovuti ad una cattiva gestione della memoria (memory leak). Quest'aspetto è fondamentale se si pensa che il normale uso di un dispositivo mobile, ad esempio di uno smartphone, può portare a non riavviare il sistema operativo per lunghi periodi di tempo rendendo necessarie soluzioni per limitare gli effetti negativi dell'aging. Altra peculiarità di un sistema operativo per dispositivi mobili è la gestione della CPU volta alla minimizzazione dei consumi elettrici. Una soluzione specifica, nei sistemi operativi per dispositivi mobili basati su eventi, è quella di disabilitare completamente la CPU quando non vi siano eventi attivi. Il corretto uso di questa tecnica aiuta ad assicurare alle batterie elettriche una durata maggiore. Un ultimo aspetto che caratterizza questi sistemi operativi, in contrasto con quelli dei sistemi per ambienti desktop, è quello relativo ai driver delle periferiche. I

[1] *Pseudo Static Random Access Memory* (PSRAM) è una RAM dinamica con built-in di aggiornamento e indirizzo di controllo. Si comporta in modo simile alla RAM statica (SRAM). Essa combina l'alta densità di DRAM (Dynamic Random Access Memory), con la facilità d'uso di SRAM. Le PSRAM sono utilizzate di più in telefoni cellulari e dispositivi mobili.

Modello	CPU	GPU[2]	Memoria (DRAM[3])
iPhone	412 MHz	PowerVR MBX Lite 3D	128 MB
iPhone 3G	412 MHz	PowerVR MBX Lite 3D	128 MB
iPhone 3GS	600 MHz	PowerVR SGX535	256 MB
iPhone 4G	1 GHz	PowerVR SGX535	512 MB

Tabella 3.1: Configurazione hardware in diversi modelli iPhone

dispositivi mobili gestiscono, a differenza dei comuni sistemi desktop, un limitato numero di periferiche di input e di output. I sistemi operativi per PDA e smartphone sono pensati per gestire esclusivamente un limitato numero di periferiche e, a differenza dei sistemi operativi per ambienti desktop, non offrono semplici meccanismi di estensibilità.

La panoramica dei principali sistemi operativi per dispositivi mobili presentata nei prossimi paragrafi consente di ottenere una visione su diverse scelte operate dai diversi produttori. È possibile osservare le differenti soluzioni offerte dai diversi produttori alle medesime problematiche. Saranno chiariti i notevoli vantaggi di vari sistemi operativi rispetto a tutti gli altri sistemi operativi presenti nel mercato dei dispositivi mobili.

3.2 iOS

Il dispositivo mobile iPhone è stato presentato al pubblico dalla Apple Inc. il 09/01/2007 [24]. Senza dubbio l'iPhone è il telefono più venduto nel mondo dei dispositivi mobili. L'iPhone rende Apple l'azienda più valutata [23] e con redito più alto al mondo [36]. Nel 11/07/2008 è stato introdotto sul mercato di molti paesi, la seconda versione di iPhone chiamato iPhone 3G dove l'aggiornamento principale era il sistema GPS. L'iPhone 3GS è stato introdotto nel 08/06/2009 e l'iPhone 4 nel 24/06/2010.

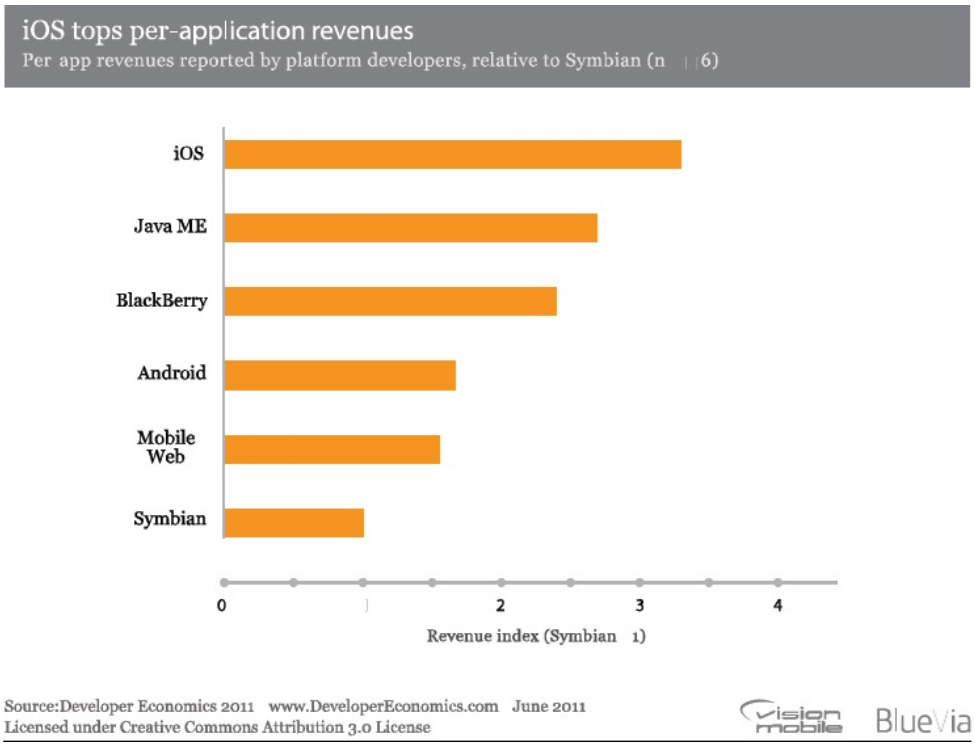

Figura 3.1: Il sistema operativo iOS classificato al primo posto con reddito più alto nel campo delle applicazioni [36]

Le loro caratteristiche prevalenti consistono in prestazioni molto veloci. Il dispositivo attualmente in commercio sottolinea l'importanza di Apple Inc. nel mercato della telefonia, con un progetto che in parte integra funzionalità di altri prodotti già in commercio sviluppati della casa. Pertanto l'iPhone può considerarsi l'insieme di tre dispositivi:

- un iPod con capacità di riproduzione audio, foto e video.
- Un telefono cellulare quad-band (GSM/UMTS HSDPA[4]/EDGE[5]) con

[4]In telecomunicazioni *l'HSDPA*, acronimo inglese di High Speed Downlink Packet Access, è un protocollo, appartenente alla famiglia di protocolli HSPA, introdotto nello standard UMTS per migliorarne le prestazioni in download, ampliandone la larghezza di banda, ed aumentando così la capacità di trasmissione delle reti radiomobili cellulari che, in download, può raggiungere la velocità massima teorica di 14,4 Mb/s.

[5]In telecomunicazioni l'*EDGE* (acronimo di Enhanced Data rates for GSM Evolution)

connettività Wi-Fi e Bluetooth e dotato di fotocamera.

- Un palmare di nuova concezione con sistema operativo derivato da Mac OS X con funzionalità GPS.

Anche per l'iPhone, così come per gli altri prodotti Apple, la scelta operata dalla casa è stata quella di legare il sistema operativo al dispositivo hardware. Diversamente dalla concorrenza non sono previsti meccanismi di portabilità del suo sistema operativo su dispositivi sviluppati da altri produttori. L'iPhone OS utilizza una versione del sistema operativo Mac OS ottimizzato per gestire le funzionalità dello smartphone. Esso è derivato precisamente dal sistema operativo per personal computer di Apple Darwin ed include, inoltre, le Core Animation (la componente presente a partire dalla versione del sistema operativo per personal computer Leopard). Il sistema operativo iPhone OS richiede per l'esecuzione meno di mezzo GB di memoria. Il processore dell'iPhone appartiene alla famiglia dei processori ARM. Il sistema operativo iPhone OS è compilato per questa tipologia di processori a differenza del sistema operativo per personal computer che è compilato per processori PowerPC[6] e X86[7]. La figura 3.2, illustra la struttura del sistema operativo.

L'iPhone OS è costituito di 4 strati.

Core OS si trova nel livello più basso e rappresenta le funzioni più primitive, le operazioni più vicine al cuore del sistema (e solitamente quindi le più complesse). Core OS gestisce l'allocazione di Memoria, informazioni locali,

o EGPRS (Enhanced GPRS) è un'evoluzione dello standard GPRS per il trasferimento dati sulla rete cellulare GSM che consente maggiori velocità di trasferimento dei dati.

[6] *PowerPC* è un'architettura di microprocessori RISC(Reduced Instruction Set Computer che indica una filosofia di progettazione di architetture per microprocessori che predilige lo sviluppo di un'architettura semplice e lineare.) creata nel 1991 dall'alleanza Apple-IBM-Motorola, conosciuta come AIM. PowerPC era il settore CPU della piattaforma dell'AIM, e ad oggi ne è l'unica parte che resta.

[7] Architettura *x86* è un'espressione generica per indicare un'architettura di una famiglia di microprocessori, inizialmente sviluppata e prodotta da Intel. È al momento l'architettura più diffusa nel mercato dei PC desktop, portatili, e server economici.

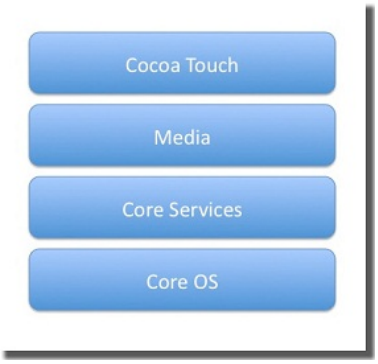

Figura 3.2: Sistema operativo iPhone [24]

sistema di input e output, il networking, l'accesso ai file di sistema, i drivers, i servizi DNS. Si può accedere alle sue funzioni tramite la libreria LibSystem.

Core Services rappresenta tutti i servizi fondamentali che le applicazioni devono gestire, come Address Book (la gestione dei contatti), come il supporto e la gestione dei tipi di dati (arrays, sets, ecc), delle date, dell'ora, delle stringhe, delle preferenze di sistema, degli URL, ecc. Inoltre gestisce i servizi di localizzazione: longitudine e latitudine, gestisce il CFNetwork.framework[8], Security.Framework, le libreria SQLite[9] per la gestione di un DataBase[10] e il supporto per XML[11].

[8]Il *.NET Framework* è la parte centrale della tecnologia .NET di Microsoft. È l'ambiente per la creazione, la distribuzione e l'esecuzione di tutti gli applicativi che supportano .NET siano essi Servizi Web o altre applicazioni.

[9]*SQLite* è una libreria software scritta in linguaggio C che implementa un DataBase Management System SQL di tipo ACID(Atomicità, Coerenza, Isolamento e Durabilità) incorporabile all'interno di applicazioni. Il suo creatore, D. Richard Hipp, lo ha rilasciato nel pubblico dominio, rendendolo utilizzabile quindi senza alcuna restrizione.

[10]Un *database* o o base di dati, indica un insieme di archivi collegati secondo un particolare modello logico (relazionale, gerarchico o reticolare) e in modo tale da consentire la gestione dei dati stessi (inserimento, ricerca, cancellazione ed aggiornamento) da parte di particolari applicazioni software dedicate.

[11]*XML* (sigla di eXtensible Markup Language) è un metalinguaggio di markup, ovvero un linguaggio marcatore che definisce un meccanismo sintattico che consente di estendere o controllare il significato di altri linguaggi marcatori.

Media tutto quello che rappresenta la grafica, media, audio, video viene gestito qui. Quelle che sono le migliori esperienze multimediali che l'iphone dispone sono gestite da questi framework.

Cocoa Touch è il livello più importante del sistema iPhone. Viene gestito da due framework UIKit e Foundation Frameworks. Questi frameworks gestiscono aspetti molto importanti come:

- Gestione delle applicazioni.
- Supporto alla grafica e alla gestione delle finestre.
- Supporto alla gestione di eventi.
- Gestione dell'interfaccia e degli oggetti che controllano il sistema.
- Supporto per testo e contenuti web.
- Gestione dell'Accelerometro.
- Gestione della fotocamera e libreria fotografica.
- Informazioni sul device.

Il linguaggio usato per sviluppare programmi per iPhone è un evoluzione dell'Objective-C. Cocoa è più precisamente un framework di Apple intorno al linguaggio Objective-C. Objective-C a sua volta è un linguaggio orientato agli oggetti costruito sopra il C. Lo sviluppo di programmi per tale sistema operativo è molto vincolato nelle scelte dalle funzionalità messe a disposizione. Si noti in particolare la presenza obbligatoria in ogni programma di una funzione "delegata" che viene richiamata all'avvio dell'applicazione, e della quale ogni oggetto ha un riferimento implicito recuperabile tramite la funzione [[UIApplication sharedApplication] delegate]. Sussiste un'altra forma di delegazione: se una classe diventa la delegata di un altra classe (per sapere quali classi possono avere un delegato è necessario consultare la documentazione del linguaggio), in questo caso può ridefinire alcuni metodi

di quella classe senza dover creare una terza classe che la derivi per poter ridefinire quei metodi.

Design Pattern Model View Controller

Lo sviluppo di programmi per iPhone in Xcode[12] è stato progettato per essere orientato al pattern MVC (Model-View-Controller) nella gestione delle schermate visualizzate a video. Quando viene creata una classe che gestisce una schermata, essa deriva dalla classe UIView- Controller e viene creato anche un file con lo stesso nome ma di estensione .xib. Esso gestisce la view vera e propria, anche se possono essere aggiunti da codice tutti i pulsanti ed oggetti grafici che si desidera. Se si sfrutta questa funzionalità e si è progettato secondo questo pattern è facile sviluppare poi il programma. L'idea di fondo comunque è quella che la classe funga da Model e controlli la logica della schermata. Il file xib, invece, gestisce la schermata grafica fungendo da View, che può essere modificata dalla model, infine è il dispositivo stesso che funge in automatico da Controller, in quanto tutti gli input esterni vengono tradotti e segnalati al model affinché la view venga cambiata.

3.3 Android

"There should be nothing that users can access on their desktop that they can't access on their cell phone."

Andy Rubin

Questa dichiarazione [7] rilasciata da Andy Rubin, direttore delle piattaforme mobili di Google, riflette esattamente l'obiettivo dello stack mobile Android (una pila include un sistema operativo mobile, middleware e le applicazioni). Android è destinato a rivoluzionare il mercato mobile portando

[12]Xcode è un ambiente di sviluppo integrato (Integrated development environment, IDE) sviluppato da Apple Inc. per agevolare lo sviluppo di software per Mac OS X e iOS (precedentemente iPhone OS).

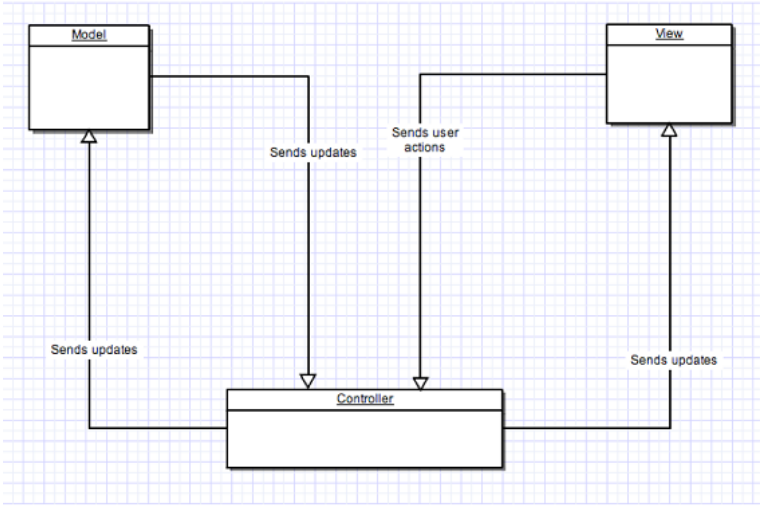

Figura 3.3: Design Pattern Model View Controller [24]

l'internet al telefono cellulare e consente il suo utilizzo nello stesso modo come sul PC. Illustreremo in questa sezione le caratteristiche principali della piattaforma mobile Android e lo confronteremo con le piattaforme mobili comunemente usate: il sistema operativo Symbian e Windows Phone.

Il termine "Android" ha la sua origine dalla parola greca *andros*, che significa *"uomo o maschio"* e il suffisso *oeides*, che significa *"simile o della stessa specie"*. Android è un software di tipo pila per i dispositivi mobili che significa un riferimento ad un insieme di sistemi di programmi oppure un insieme di programmi applicativi che formano insieme un sistema completo. Questa piattaforma software fornisce una base per le applicazioni come una vera e propria piattaforma di lavoro.

Lo stack del software viene diviso in quattro stati diversi, che comprendono 5 gruppi diversi:

- Il livello di applicazione

  La piattaforma software Android ha un insieme di applicazioni basi come browser, email, sms, mappe, calendari, contatti e molto altro ancora. Tutte queste applicazioni sono scritte usando il linguaggio di

programmazione Java. Va ricordato il fatto che queste applicazioni possono essere eseguite contemporaneamente: è possibile ascoltare la musica e leggere una mail allo stesso tempo. Questo strato sarà più utilizzato dagli utenti che usano molto spesso il telefono cellulare.

- Il framework applicativo

  Un framework applicativo è un software che viene utilizzato per implementare una struttura standard di un'applicazione di uno specifico sistema operativo.

- Le librerie

  Tutte le librerie disponibili vengono scritte in C/C++ per poi essere chiamati da un'interfaccia Java. Questo include il "Surface Manager", grafici 2D e 3D, media come MPEG-4 e MP3, il database SQL SQLite e il motore web browser WebKit.

- Runtime

  Il runtime Android è costituito da due componenti. Prima di tutto una serie di librerie di base che forniscono la maggior parte delle funzionalità disponibili nelle librerie base del linguaggio di programmazione Java. In più, la macchina virtuale Dalvik che opera da traduttore tra l'applicazione e il sistema operativo. Ogni applicazione che funziona su Android è scritta in Java. Siccome il sistema operativo non è capace di capire questo linguaggio di programmazione direttamente, i programmi Java saranno ricevuti e tradotti dalla macchina virtuale Dalvik. Il codice tradotto può allora essere eseguito dal sistema operativo. Si nota che le applicazioni saranno incapsulate in Dalvik. Una macchina virtuale propria è disponibile per ogni programma anche se alcuni programmi si eseguono parallelamente. Il vantaggio è che i programmi diversi non vengono influenzati reciprocamente, per cui un errore di programma può portare a un crash del programma ma non di tutto il sistema.

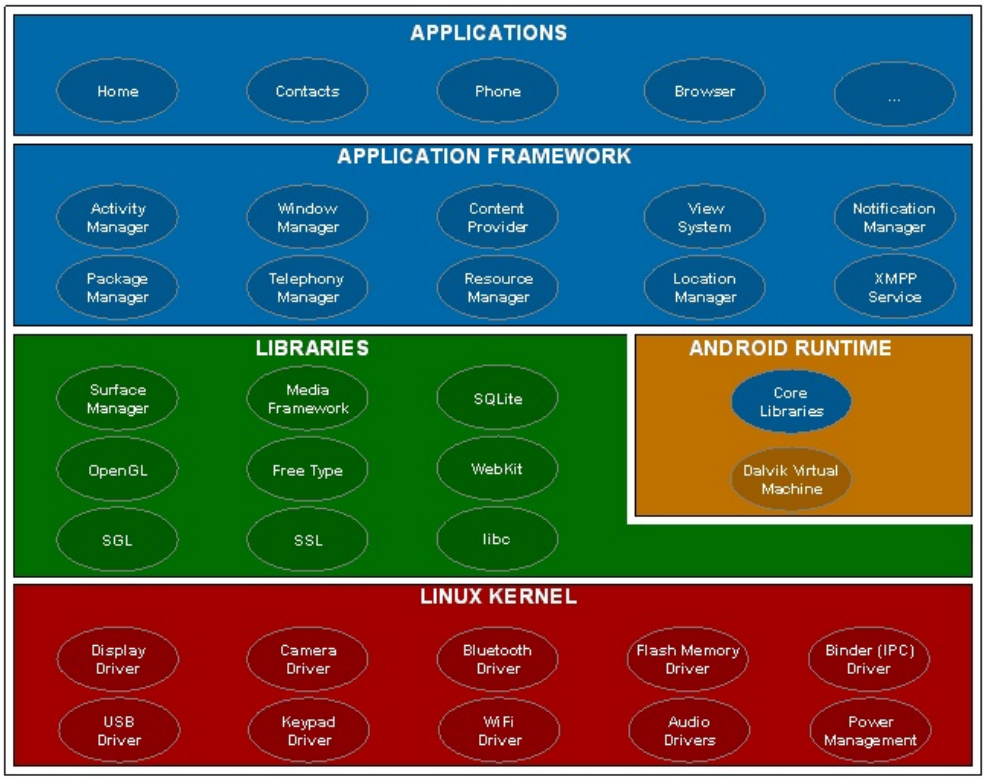

Figura 3.4: Componenti del sistema operativo Android [8]

- Kernel

  Il kernel Linux sarà utilizzato da Android per i driver dei dispositivi, la gestione della memoria, process management e il networking.

Lo schema in Figura 3.4 mostra i componenti principali del sistema operativo Android [8].

Caratteristiche importanti integrati in Android

Android offre molte caratteristiche che coprono molti settori come lo sviluppo di applicazioni, internet, media e connettività. Alcuni tra i più importanti verranno presentati nel seguente elenco:

- Framework di applicazione che consente il riutilizzo e la sostituzione dei vari componenti
- La macchina virtuale Dalvik ottimizzata per i dispositivi mobili
- Browser basato sul motore open source WebKit
- Grafica ottimizzata alimentato da una libreria grafica 2D, grafica 3D basata sulle specifiche OpenGL ES 1.0
- SQLite per la memorizzazione di dati strutturati
- Il supporto multimediale per audio, video e formati di immagini (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF)
- Telefonia GSM
- Bluetooth, EDGE, 3G e WiFi
- Camera, GPS, bussola e accelerometro
- Ambiente di sviluppo includendo un emulatore di dispositivi, strumenti per il debug, memoria e profiling delle prestazioni e un plugin per Eclipse.

3.4 Blackberry

Blackberry Device Software è il sistema operativo proprietario sviluppato da Research In Motion per la sua linea di smartphone [32]. I dispositivi Blackberry presentano come periferica di input un trackball, attraverso cui è possibile navigare tra i menù, e, solo i modelli più recenti, sono dotati di un touchscreen. Il sistema operativo, dalla versione 4.0, supporta parte delle specifiche Java MIDP 2.0 e consente la sincronizzazione con i server di posta Exchange e Lotus Domino, oltre a fornire una serie di applicazioni che vanno dalle agende elettroniche, ai riproduttori di file video e audio. Lo sviluppo di software può avvenire utilizzando le API proprietarie, benché l'uso di alcune

funzionalità sia possibile soltanto dopo un meccanismo di firma digitale, che garantisce la paternità dei prodotti sviluppati, ma non qualità e sicurezza del codice.

La Research in Motion Inc. è stata fondata nel 1984 da uno studente della facoltà ingegneria dell'University of Waterloo, Mike Lazaridis (Presidente della società) e da uno studente della facoltà di ingegneria dell'University of Windsor, Douglas Fregin (vice Presidente).

Nel 1998 è stato introdotto sul mercato il primo modello di Blackberry, il 950. Tale dispositivo è dotato di un display monocromatico LCD, di un processore a 32 bit Intel 386, di 1 megabyte di memoria flash e 204 kilobytes di memoria SRAM, di una tastiera a 31 tasti e di un trackball. La batteria è dotata di sufficiente potenza da garantire l'uso dello smartphone per circa tre settimane.

La killer application del modello 950 è stata senz'altro la posta elettronica. Era infatti possibile ricevere e-mail direttamente sul dispositivo senza dover ricorrere all'utilizzo del personal computer. Altra caratteristica notevole è stata la cifratura del traffico di rete relativo allo scambio di posta elettronica. I modelli attualmente in produzione supportano funzionalità tra cui: GPS, connettività wireless e connettività 3G/HSDPA. Come detto, la caratteristica principale di questi dispositivi è la gestione delle e-mail. Tramite il servizio push mail, la posta elettronica viene consegnata da appositi server analogamente a quanto avviene per i comuni SMS. Esistono due differenti modalità di gestione della posta: BlackBerry Internet Service (BIS) e BlackBerry Enterprise Server (BES).

La prima modalità prevede che l'operatore telefonico, una volta che l'utente abbia configurato i servizi BlackBerry, si connetta ad uno o più indirizzi di posta indicati dall'utente utilizzando gli standard. L'operatore telefonico funge in questo caso da tramite tra cassetta postale dell'utente ed il dispositivo Blackberry controllando la presenza di nuova posta ad intervalli di circa 10 - 15 minuti e inoltrando la posta da e verso lo smartphone. Le limitazioni di questo tipo di servizio sono: la mancanza della possibilità di

sincronizzare calendario e contatti, dal momento che gli standard di posta POP3 e IMAP non supportano tale funzionalità e la non istantaneità della ricezione della posta (l'operatore telefonico effettua il controllo della cassetta ad intervalli di tempo). Il vantaggio rispetto al secondo sistema è che il BIS richiede minori investimenti. La seconda modalità, BlackBerry Enterprise Server, prevede che sia presente un server di posta all'interno dell'azienda (Microsoft Exchange, Lotus Domino o Novell GroupWise). Il server BES mantiene costantemente sincronizzati i telefoni con le cassette postali sul server di posta, inoltrando le email alla rete BlackBerry appena arrivano. Blackberry fornisce un set di tools, BlackBerry® Java® Development Environment, per lo sviluppo di applicazioni Java. È altresì possibile utilizzare BlackBerry JDE per sviluppare applicazioni Java Micro Edition. Quest'ultimo strumento contiene una serie di tools per sviluppare, testare e distribuire applicazioni, incluso un simulatore del dispositivo Blackberry. È inoltre possibile l'integrazione con l'IDE Eclipse. Le linee guida per lo sviluppo di applicazioni per Blackberry ricalcano quei principi da tenere sempre in considerazione durante la progettazione di software per smartphone. Occorre considerare che gli smartphone:

- sono dotati di schermi che possono visualizzare un limitato numero di caratteri.
- Dispongono di processori meno potenti di quelli di cui sono dotati gli odierni personal computer.
- Hanno limitate disponibilità di memoria.
- Dispongono di batterie elettriche di durata limitata.
- Dispongono di un sistema che consente la visualizzazione di uno screen di applicazione per volta.

Occorre bilanciare la massima usabilità da parte dell'utente con il minimo consumo elettrico, al fine di garantire una sufficiente durata delle batterie.

3.5 Windows Mobile

Il primo sistema operativo per palmari sviluppato da Microsoft (Windows CE 1.0) risale al 1996 ed è stato progettato per processori PowerPC, MIPS e Intel [32]. La soluzione forniva agli sviluppatori Windows un ambiente di sviluppo simile a quello standard, che presentava una scelta di chiamate al sistema operativo ricavate da quelle WIN-32. Windows CE è un sistema operativo multithread, con un'interfaccia utente grafica opzionale ed è multipiattaforma. Con Windows Mobile 6.5, rilasciato nel maggio del 2009, il sistema operativo Microsoft per dispositivi mobili offre pieno supporto a tutte le funzionalità richieste per gli attuali smartphone. L'utente Windows, con l'utilizzo di Windows Mobile, si trova ad operare in un ambiente simile a quello dei comuni desktop PC Windows. La versione 6 di Windows Mobile si basa su Windows CE 5. Windows CE 5 [67] è la piattaforma di sviluppo per il sistema operativo di Microsoft. La flessibilità e la modularità della piattaforma ha consentito lo sviluppo di differenti versioni per differenti dispositivi e processori. La Figura 3.5, illustra l'architettura della piattaforma Windows CE.

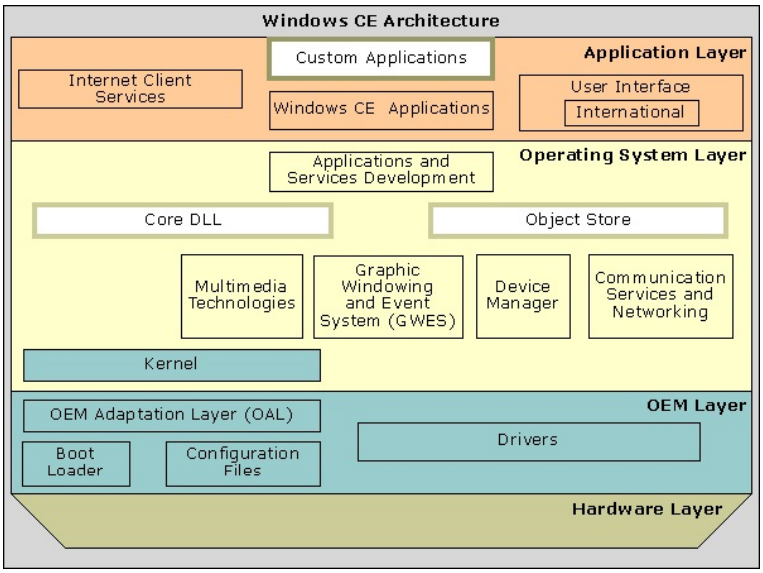

Figura 3.5: Architettura di Windows CE [67]

Windows CE 5 implementa le funzionalità del core in processi separati PSL (Process Server Libraries); il file system, il windows manager grafico (gwes) e i drivers sono eseguiti in user space. Lo sviluppo delle applicazioni è facilitato da Windows Mobile SDK, un tool di sviluppo che fornisce strumenti grafici per l'editing, per la compilazione e per il testing del codice. Tra gli strumenti di sviluppo forniti da Microsoft ci sono anche emulatori per un discreto numero di dispositivi mobili: è dunque possibile sviluppare codice anche qualora non si abbia accesso ad alcuni dispositivi fisici.

Windows Phone 7 è la nuova versione del sistema operativo per smartphone di Microsoft, che lo ha presentato al Mobile World Congress il 15 febbraio 2010. È completamente differente da tutte le precedenti versioni di Windows Mobile; supporta il multitouch, gli schermi capacitivi, ha una nuova interfaccia grafica molto simile a quella di Zune HD, e riunisce in una sola piattaforma i contenuti di Xbox LIVE e Zune. Inoltre gestisce gli account di social network quali Facebook e Twitter, e possiede una nuova versione di Internet Explorer basata su Windows Internet Explorer 7 con alcuni elementi della versione 8. Questa versione di Windows Phone conterrà una edizione Mobile di Office 2010, con Word, Excel, Powerpoint, OneNote e Sharepoint.

3.6 Symbian OS

Symbian OS è un sistema operativo per dispositivi mobili, prodotto da Symbian Foundation [68]. Nasce closed source, ma di recente si è avviato il processo che renderà open source, con licenza EPL, l'intero codice del sistema operativo. Symbian Ltd. nasce nel 1998 a Londra dalla cooperazione tra Nokia, Motorola, Ericsson e Psion, con l'intento di sviluppare un sistema operativo ed una piattaforma software che potessero adattarsi a dispositivi mobili come palmari e smartphone. L'anno seguente,il gran potenziale del progetto, è stato riconosciuto anche dalla Panasonic, che ha acquistato quote della neonata società. Il sistema operativo Symbian nasce dal progetto di Epoc, realizzato dalla Psion nel 1989 anno in cui si apprestava a mettere

sul mercato il primo modello di palmare mai prodotto. Il primo telefonino dotato di sistema operativo Symbian è stato l'R380 realizzato dalla Ericsson nel 2000. La versione deriva direttamente da Epoc release 5. Negli ultimi anni Symbian Ltd. si è affermata come leader mondiale nello sviluppo di sistemi operativi per dispositivi mobili. Ad oggi Symbian equipaggia il maggior numero di smarthphone in commercio. Dal 2000 le differenti release hanno costantemente aggiunto nuove funzionalità. Sono stati introdotti, nel tempo, supporto alla tecnologia Bluetooth ed alla tecnologia IrDA, agli standard EDGE e 3G, ai servizi offerti dal protocollo IPv6, al VoIP ed alle reti Wi-Fi. Da Maggio 2009 è disponibile la nuova release Symbian 2 che equipaggia il Nokia N97.

Le versioni con codice aperto di Symbian OS sono:

- *Symbian 1*, essendo la prima release uscita nell'ottobre del 2008, costituisce la base per la piattaforma. Incorpora Symbian OS e S60 5th Edition (che si basa su Symbian OS 9.4) e quindi non è stato reso disponibile in open source.

- *Symbian 2* è stato rilasciato a titolo gratuito il 1° giugno del 2010. Alcune porzioni sono concesse in licenza EPL, ma la maggior parte del codice sorgente è sotto la licenza proprietaria SFL e disponibile solo ai membri della Symbian Foundation.

- *Symbian 3*, rilasciato a settembre 2010 ha introdotto nuove caratteristiche come un nuovo 2D e 3D architettura grafica, miglioramenti dell'interfaccia utente e il supporto per display esterno tramite HDMI.

  - *Symbian Anna* rilasciato da Nokia nel mese di aprile 2011, include miglioramenti come ad esempio un nuovo browser, una tastiera virtuale con orientamento verticale, nuove icone e in tempo reale lo scorrimento homescreen.

  - *Symbian Belle* rilasciata il 24 agosto 2011, aggiunge una barra di notifica, più profonda integrazione della comunicazione di campo

vicino, a forma libera ri-considerevole widgets homescreen, e sei schermi madre invece dei precedenti tre.

La Figura 3.6 illustra la struttura generica del sistema operativo Symbian.

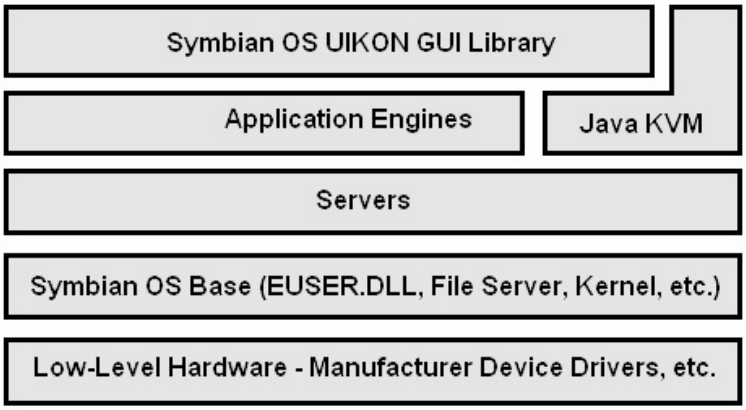

Figura 3.6: Struttura del sistema operativo Symbian [68]

La figura mostra come il kernel, il file server, i driver del dispositivo e la sezione relativa alla gestione della memoria siano localizzati al livello inferiore della struttura. In realtà, a livello di maggior dettaglio, Symbian presenta una struttura a microkernel, ovvero, il kernel gestisce direttamente solo una parte minima e strettamente necessaria dei servizi e ciò garantisce elevata robustezza, affidabilità ed efficienza. Altri servizi, quali networking, telefonia e gestione del filesystem sono collocati, all'interno della struttura del sistema operativo, a livelli superiori. Le caratteristiche del sistema operativo Symbian possono essere riassunte nei seguenti punti:

- performance: il sistema è progettato per minimizzare i consumi elettrici e per lavorare con memorie di bassa capacità;
- multitasking: le applicazioni software devono poter essere eseguite simultaneamente;
- standard: l'uso di tecnologie standard è uno dei principi cardine su cui si basa il sistema operativo Symbian;

- object oriented: architettura software orientata agli oggetti;
- memory management optimized: gestione ottimizzata della memoria;
- runtime memory requirements are minimized: i progettisti compiono lo sforzo di minimizzare la quantità di memoria necessaria al funzionamento del sistema;
- security: sono implementati meccanismi di sicurezza per garantire le comunicazioni e l'integrità dei dati;
- supporto applicativo per l'international environment con la presenza del set di caratteri Unicode;
- un'elevata varietà di API per consentire la reusable component per il software applicativo;

Il linguaggio nativo di Symbian è il C++ e ci sono molteplici piattaforme basate su Symbian OS che forniscono l'SDK per gli sviluppatori. È opportuno osservare che la curva di apprendimento per lo sviluppo di software per dispositivi mobili Symbian è molto ripida a causa di paradigmi di programmazione legati ancora agli standard dei dispositivi degli anni '90. Altri linguaggi con cui è possibile sviluppare software applicativo per dispositivi mobili dotati di sistema operativo Symbian sono Java Micro Edition e Python.

3.7 Differenze tra i sistemi operativi

In questa sezione si affronteranno i confronti tra i sistemi operativi descritti precedentemente [22]. Prima di confrontare questi sistemi operativi introduciamo una terminologia di base che include le seguenti domande:

- Che cos'è un sistema operativo?

  Un sistema operativo è un'unità organizzativa all'interno di un sistema informatico. È l'interfaccia tra le applicazioni e l'hardware. La sua

funzione primaria è l'amministrazione delle risorse operative a disposizione.

- Che cos'è un sistema mobile?

  Un sistema mobile è un sistema informatico che non è vincolato ad un determinato posto. È possibile spostarlo oppure portarlo in giro come ad esempio un telefono cellulare, un palmare, ecc. Anche se ci sono molte somiglianze tra fisso e un sistema operativo mobile, ci sono anche distinzioni chiare riguardo alla mobilità. Un esempio per un'applicazione in cui un normale sistema operativo non è in grado di essere utilizzato è il controllo di ABS[13] in un auto. Un sistema operativo come Windows XP che non è sufficientemente stabile da garantire il funzionamento del sistema ABS per un lungo periodo, non può essere utilizzato. Questo esempio fa notare quali attributi sono importanti per un sistema mobile di un qualsiasi dispositivo: Il sistema deve essere stabile e a prova di fallimento.

3.7.1 Classificazione dei sistemi operativi per i dispositivi mobili

I dispositivi mobili hanno cambiato il loro profilo drammaticamente negli ultimi anni. I cellulari avanzati di oggigiorno integrano pienamente le funzionalità dei personal digital assistant (PDA) con quelle di un telefono cellulare tradizionale. Esamineremo i fattori critici per i sistemi operativi che operano nel mercato e che li differenziano gli uni dagli altri.

La classificazione dei sistemi operativi deve considerare il mercato in cui loro vengono utilizzati. Il mercato per i dispositivi mobili avanzati è difficile da paragonare ad altri mercati come quello di PC in cui vengono utilizzati anche i sistemi operativi. Ovviamente, i bisogni degli utenti e le loro esigenze

[13] Il Sistema anti bloccaggio, dall'acronimo inglese ABS (Antilock Braking System) è un sistema di sicurezza che evita il bloccaggio delle ruote dei veicoli garantendone la guidabilità durante le frenate.

sono diverse. Symbian, produttore del sistema operativo di telefonia mobile Symbian OS [17], ritiene che il mercato della telefonia mobile ha cinque caratteristiche fondamentali che lo rendono unico:

1. Mobilità: i telefoni cellulari sono piccoli e mobili
2. Universali: i telefoni cellulari sono onnipresenti - prendono di mira un mercato grande di consumatori, utenti aziendali e professionali
3. Collegamento: i telefoni cellulari sono connessi occasionalmente - possono essere collegati alla rete wireless, localmente con altri dispositivi, oppure da soli.
4. Innovazione: i produttori devono differenziare i loro prodotti al fine di innovare e competere in un mercato in rapida evoluzione.
5. Aperti: la piattaforma deve essere aperta per abilitare la tecnologia a fornitori software per lo sviluppo di applicazioni di terze parti, tecnologie e servizi.

Queste cinque caratteristiche del mercato della telefonia mobile sottolineano la differenza da altri mercati dove questi sistemi operativi vengono utilizzati. Per avere successo in questo mercato, un prodotto deve rispondere a queste caratteristiche senza limitare le sue funzionalità.

3.7.2 Confronto tra i sistemi operativi

- **Android OS**
  - Piattaforma aperta.
  - Può compilare firmware personalizzati.
  - È un buon framework, esteso su ogni nuovo firmware.
  - Supporta il multitasking.
  - Lo sviluppo dell'SDK è gratuito.

  - Facile da debaggare, è possibile inviare i log agli sviluppatori.
  - Supporta Java e la macchina virtuale Dalvik è ottimizzata per i dispositivi mobili.
  - Offre la possibilità di programmare in C usando il dev kit nativo "NDK".
  - Può eseguire linguaggi di scripting come LUA, Perl, Python, ecc.
  - È possibile installare applicazioni di terze parti da sdcard, diversi siti, ecc.
  - Le applicazioni possono sovrascrivere tutto: le interfaccie e-mail, l'invio di SMS, tastiere personalizzate, ecc.
  - Supporta i widget.
  - Può pubblicare immediatamante applicazioni sul mercato Android con una quota d'iscrizione.
  - Supporta Adobe Flash.
  - Fornisce GSM, Bluetooth, EDGE (tecnica per aumentare la velocità di trasmissione dei dati in GSM), 3D e WiFi.

- **iOS**
  - Piattaforma chiusa.
  - Multitasking
  - Il pacchetto developer tools dell'Xcode può essere incluso in ogni Mac in modo che ognuno possa creare applicazioni per iPhone e iPad.
  - Il linguaggio di programmazione è Objective C.
  - L'utente non ha accesso alla sdcard - l'utente può fare la sincronizzazione solo via internet o LAN.
  - Le applicazioni di terze parti possono essere intallate dal negozio Apple store. Per testare le applicazioni, gli sviluppatori possono usare la pubblicazione ad hoc.

  - La pubblicazione sull'Apple store è un processo lungo e a volte faticoso.
  - Non c'è supporto ufficiale di Adobe Flash.
  - Fornisce GSM, Bluetooth, EDGE (tecnica per aumentare la velocità di trasmissione dei dati in GSM), 3D e WiFi.

- **Symbian OS**
  - Piattaforma aperta.
  - Dispone funzionalità di multithreading, multitasking e protezione di memoria.
  - La CPU viene disabilitata quando non ci sono eventi attivi (questo assicura alle batterie una durata maggiore).
  - Sono disponibili un discreto numero di programmi, sia gratuiti che a pagamento rendendo il prodotto espandibile e personalizzabile.
  - È possibile visionare e (in alcuni casi) modificare direttamente dal telefono documenti di Word, Excel, PowerPoint, PDF e Outlook Express.
  - Supporta Adobe Flash Lite.
  - Supporta GSM, Bluetooth, Infrarossi e WiFi.

- **Blackberry OS**
  - Piattaforma chiusa.
  - L'ambiente di programmazione è Java e J2ME.
  - Il sistema operativo supporta il WAP 1.2.
  - I sviluppatori di terze parti sono in grado di scrivere applicazioni utilizzando le classi API disponibili per Blackberry OS.
  - Le applicazioni che fanno uso di determinate funzionalità devono essere firmate digitalmente e possono essere aggiornate direttamente dal telefono tramite l'applicazione AppWorld.

  - Non fornisce nessun supporto per Adobe Flash Player.
  - Fornisce GSM, Bluetooth, EDGE (tecnica per aumentare la velocità di trasmissione dei dati in GSM) e WiFi.

- **Windows Mobile**
  - Piattaforma chiusa.
  - Sistema operativo compatto basato sulle API Win32 di Microsoft.
  - L'ambiente di programmazione è Silverweight e XNA.
  - Supporta il multitouch, gli schermi capacitivi.
  - Gestisce gli account di social network quali Facebook e Twitter e possiede una versione di Internet Explorer basata su Windows Internet Explorer 7.
  - Vengono integrate alcune applicazioni della suite Microsoft Office in versione adattata a Pocket PC (Pocket Word, Pocket Excel e Pocket PowerPoint).
  - Non supporta il multitasking per le applicazioni di terze parti.
  - Le applicazioni di terze parti possono essere installate solo dal MS marketplace.
  - Non fornisce nessun supporto per Adobe Flash Player.
  - Supporta GSM, Bluetooth, Infrarossi e WiFi.

Capitolo 4

Approcci per lo sviluppo di applicazioni mobili

Lo sviluppo delle tecnologie wireless e l'enorme diffusione dei dispositivi mobili sempre più potenti e facilmente trasportabili ha spinto una gran parte del mercato nella direzione del mobile business. Cresce costantemente il numero delle aziende che ricorrono alle applicazioni su piattaforme mobili per rimanere sempre più vicini ai loro clienti. In questo capitolo si illustreranno alcuni approcci per lo sviluppo di applicazioni mobili.

4.1 Applicazioni web vs. Applicazioni native

Esistono due approcci fondamentali per far interagire l'utente di un dispositivo mobile con un'applicazione. Uno è rappresentato dall'interazione con una pagina web, tramite la navigazione in un network o la navigazione in pagine offine. L'altro approccio è la realizzazione di un applicativo mirato e specifico per quel tipo di dispositivo o sistema operativo. Nei due casi rispettivamente si parla di Applicazione Web e di Applicazione Nativa; vediamo di seguito le differenze principali e come i due approcci sono in relazione fra loro.

- Un'applicazione Web (web-app) [53] è un'applicazione accessibile via web mediante un network, come ad esempio una intranet o attraverso la Rete Internet. Questo modello applicativo è divenuto piuttosto popolare alla fine degli anni novanta, in considerazione della possibilità per un client generico di accedere a funzioni applicative, utilizzando come terminale normali web browser. Infatti l'opportunità di aggiornare ed evolvere a costo ridotto il proprio applicativo, senza essere costretti a distribuire numerosi aggiornamenti ai propri clienti attraverso supporti fisici, ha reso la soluzione piuttosto popolare per molti produttori software. Nel caso di uno smartphone è possibile accedere tramite una connessione GSM o Wireless. In generale, si parla di applicazione Web quando la funzione svolta dalla pagina è più che la semplice consultazione. Il suo contenuto è in genere dinamico e interattivo. Ciò che può essere definito una web-app sono software come webmail, e-commerce, web forum, blog, giochi online e altro. Troviamo applicazioni web ovunque. Esempi comuni sono quelle applicazioni che ci permettono di cercare sul web, come il motore di ricerca Google; di collaborare a progetti, come SourceForge[1]; di acquistare prodotti da un'asta, come avviene su eBay[2]. Inoltre sono nati molti siti web che fungono da directory di script, spesso gratuiti (Opensource), come Hotscripts.com [54]. Ma questo è solo un esempio poiché usando un qualsiasi motore di ricerca è facile scoprire un'infinità di applicazioni web in tutte le lingue e paesi. La finestra che consente all'utente l'interazione con queste applicazioni è il browser. In genere questo tipo di software è realizzato impiegando dei linguaggi di programmazione

[1] *SourceForge* è una piattaforma e un sito web che fornisce gli strumenti per portare avanti un progetto di sviluppo software in modo collaborativo.

[2] *eBay* è un sito di aste on-line fondato il 6 settembre 1995 da Pierre Omidyar. È una piattaforma (marketplace) che offre ai propri utenti la possibilità di vendere e comprare oggetti sia nuovi che usati, in qualsiasi momento, da qualunque postazione Internet e con diverse modalità, incluse le vendite a prezzo fisso e a prezzo dinamico, comunemente definite come "aste online".

che lasciano la computazione e la gestione del comportamento della pagina al Server. Il codice della pagina in questo caso è compilato lato-server, al browser viene fornita la pagina web senza il codice di programmazione utilizzato al suo interno. Questo rende la gestione della pagina sicura in quanto l'utente non verrà a sapere come vengono gestiti i dati. L'alternativa per realizzare una web application è quello di appoggiarsi a codice Javascript. Questo codice sarà all'interno della pagina e sarà compito del web Browser interpretarlo e gestirlo. Quello che consente di portare le applicazioni web dal web al mobile sono i diversi framework.

- Un'applicazione mobile nativa è [55] un software realizzato ad hoc per una o più piattaforme e contengono normalmente una grande quantità di dati (immagini e testi). Ciò comporta l'utilizzo del linguaggio di programmazione adatto, l'installazione di una SDK3, la configurazione di eventuali piattaforme di sviluppo legate al sistema target. Per alcune piattaforme proprietarie è necessario utilizzare hardware adeguati per compilare le applicazioni. I vantaggi derivanti da un approccio nativo sono legati a:

  - un incremento delle prestazioni (velocizza e semplifica la fruizione dei dati (perchè una parte di essi risiede nel dispositivo mobile));
  - una precisione superiore nella creazione dell'interfaccia utente;
  - un maggiore controllo nella gestione degli eventi;
  - possibilità di interfacciamento con tutte le possibilità hardware del dispositivo.

  A fronte di un maggiore impegno di progettazione e codifica esiste la possibilità di progettare l'applicazione in ogni dettaglio in modo da rendere il suo aspetto unico.

  Come vedremo emergeranno pregi e difetti dei due approcci, con riferimento alle rispettive piattaforme.

- **Pro e contro tra Applicazioni Web e Applicazioni Native**

  Applicazioni differenti hanno esigenze diverse. Alcune applicazioni funzionano meglio utilizzando le tecnologie web piuttosto che altre. Conoscere i pro e i contro di ciascun approccio aiuterà a prendere la giusta decisione su quale approccio sia migliore per ciascun caso specifico.

- Alcuni *pro* dello sviluppo di **applicazioni native** sono:

  - Facilmente reperibile, perchè si trova solo nei negozi di applicazioni del produttore e da esso è certificata e garantita.
  - Efficaci ambienti di sviluppo.
  - La possibilità di accedere a tutte le caratteristiche hardware del dispositivo.
  - Non dipende dall'accesso alla rete
  - Permette di monetizzare più facilmente i contenuti perchè, ad esempio, è più difficile estrarne i contenuti e redistribuirli, scoraggiando la pirateria.
  - Sono reperibili: ossia il fatto di trovare un'applicazione in un ecosistema garantito come quello dei produttori garantisce visibilità al produttore e tutela il consumatore.
  - Sfruttano appieno le particolarità del telefono: oggi i telefoni sono dotati di GPS, giroscopio, oscillometro, ecc. Molte applicazioni native non usano queste funzionalità ma non è da escludere che lo facciano un domani.
  - Sono remunerative, vale a dire che attraverso una applicazione nativa si accede all'ecosistema del produttore di telefoni e anche ai metodi di pagamento da esso usati.

- I *contro* dello sviluppo di **applicazioni native** sono:

  - Si deve sviluppare su di una piattaforma specifica, per esempio un Mac utilizzando il linguaggio Objective C.
  - Non è possibile rilasciare correzioni di un bug in modo tempestivo.
  - Il ciclo di sviluppo è lento e quello di testing è vincolato dai limiti dell'App Store.

- I *pro* dello sviluppo di **applicazioni web** sono:
  - Gli sviluppatori web possono utilizzare i loro strumenti di authoring usuali.
  - Si può utilizzare il design attuale e le competenze di sviluppo già acquisite.
  - Le applicazioni web possono essere eseguite su qualsiasi dispositivo che disponga di un browser web.
  - Si possono correggere i bug in tempo reale.
  - Il ciclo di sviluppo è più veloce.

- Alcuni *contro* dello sviluppo di **applicazioni web** sono:
  - Non si può accedere a tutte le caratteristiche hardware del telefono.
  - Si deve sviluppare un proprio metodo di pagamento se si vuole mettere in vendita l'applicazione.
  - Può essere difficile ottenere effetti sofisticati nell'interfaccia utente.

4.2 Scegliere l'approccio migliore

Qual è il migliore approccio?

La natura sempre online dei dispositivi mobili fornisce un ambiente in cui la linea di separazione tra un'applicazione web e un'applicazione nativa diventa molto sfocata. Ci sono alcune funzionalità poco note dei dispositivi

mobili che consentono di utilizzare un'applicazione web, se si vuole, anche in modalità offline [56]. Inoltre, diversi progetti di terze parti, tra i quali il più noto è PhoneGap, stanno sviluppando attivamente alcune soluzioni che permettono agli sviluppatori web di trasformare un'applicazione web in un'applicazione nativa per le piattaforme mobili.
Questa è una miscela perfetta. Si può scrivere nel linguaggio nativo, rilasciare un prodotto come applicazione web pura (per i dispositivi mobili che dispongono di un browser moderno) senza passare; per esempio, attraverso Apple e utilizzare lo stesso codice per creare una versione nativa avanzata dell'applicazione che può accedere all'hardware del dispositivo ed essere potenzialmente venduta nell'App Store. E se Apple la rifiuta non ci sono problemi poiché sono sempre in grado di lavorare sulla versione nativa, mentre i clienti utilizzano l'applicazione web.
Assieme agli sviluppi dell'hardware è determinante la sempre maggiore spinta di internet a portare l'utente ad avere un'esperienza sempre più coinvolgente nel navigare. Questa interazione con il web avviene non solo grazie al dispositivo hardware stesso; ma grazie anche ai web browser. Le case produttrici di browser svolgono un grande ruolo nel mercato di internet. Cercano di rendere l'esperienza di navigazione la migliore possibile per i loro utenti, in termini di velocità, facilità di utilizzo, compatibilità con il maggior numero di pagine; il tutto volto a guadagnare una maggiore fetta di mercato. Queste due motivazioni hanno spinto le associazioni che progettano standard a concentrare gli sforzi per creare una base che metta insieme l'esigenza dell'utente e quella delle grandi compagnie. Le compagnie produttrici di browser chiedono uno standard, gli utenti chiedono una maggiore velocità e interattività con il web.

4.3 Linguaggi di programmazione per le applicazioni web

Le tre principali tecnologie che si utilizzano per costruire applicazioni web sono HTML, CSS e JavaScript. Questi tre linguaggi contribuiscono a

creare la struttura della pagina web (HTML), di personalizzarne il suo aspetto (CSS), e di renderla dinamica e interattiva mediante il codice Javascript. In seguito si illustra in breve in cosa consiste ciascuno dei tre.

4.3.1 HTML

HTML stà per HyperText Markup Language, è il linguaggio predominante delle pagine web [49]. Per descrivere le pagine web viene usato un sistema di tag. I tag sono costituiti da parentesi angolate (come `<html>`), il browser è programmato per riconoscere i tag e la loro funzione, una volta riconosciuti non li mostra nella pagina finale ma li utilizza per costruire la pagina. É possibile inserire immagini e oggetti e creare moduli (d'ora in poi *form*) interattivi. Esistono tag per strutturare il documento con link, intestazioni, liste, paragrafi e altri elementi. É possibile inserire all'interno del codice della pagina altri linguaggi attivi, ad esempio si può incorporare del codice Javascript o utilizzare altri linguaggi più server-side per svolgere compiti più complessi come PHP[3] o Java [71]. Le specifiche del linguaggio HTML sono state pubblicate dal World Wide Web Consortium (W3C) [57].

4.3.2 CSS

Il Cascading Style Sheets (CSS) è un linguaggio che definisce lo stile, la formattazione e l'aspetto di un documento scritto in un linguaggio di markup [50]. È più comunemente usato assieme alle pagine web scritte in HTML o XHTML, ma può anche essere applicato ad ogni tipo di documento XML. Il linguaggio definisce i comportamenti visivi di una pagina web. Ad ogni elemento della pagina viene associata una serie di comandi che modificano l'aspetto dell'elemento stesso, e il suo comportamento all'interno della pagina. L'intero codice va a creare uno script che viene associato alla pagina stessa

[3] *PHP*, originariamente acronimo di Personal Home Page, ed ora visto come un preprocessore di ipertesti è un linguaggio di scripting interpretato, con licenza open source e libera (ma incompatibile con la GPL), originariamente concepito per la programmazione Web ovvero la realizzazione di pagine web dinamiche.

e definisce l'aspetto degli elementi in base ad alcune azioni dell'utente (click di alcuni elementi o altro). Definisce il posizionamento, la colorazione, il comportamento del testo, degli elementi, di immagini. Mentre l'autore di un documento tipicamente associa il proprio documento ad uno specifico CSS, i lettori possono utilizzare un foglio di stile proprio e sovrascrivere quello che il proprietario ha specificato. Le specifiche del CSS sono aggiornate dal W3C. L'introduzione del CSS si è resa necessaria per separare i contenuti delle pagine web dalla formattazione delle stesse, in modo tale da permettere una programmazione più chiara e di facile utilizzo, sia per gli autori che per gli utenti. Tra le novità più rilevanti dei CSS esiste l'introduzione delle funzionalità per la gestione di:

- bordi, con la possibilità di inserire angoli arrotondati e ombre;
- sfondi, con la possibilità di definire sfondi multipli, dimensioni e origini degli stessi;
- colori, definibili tramite parametri HSL (Hue, Saturation, Lightness) o RGB (Red, Green, Blue), a loro volta combinabili con il grado di trasparenza e opacita;
- effetti sul testo, ad esempio l'ombreggiatura.

4.3.3 Javascript

JavaScript è un linguaggio di scripting orientato agli oggetti comunemente usato nei siti web. Fu sviluppato da Brendan Eich della Netscape Communications con il nome di *Mocha* e successivamente di *LiveScript*, ma in seguito è stato rinominato *"JavaScript"* ed è stato formalizzato con una sintassi più vicina a quella del linguaggio Java di Sun Microsystems [51].

La caratteristica principale di JavaScript è quella di essere un linguaggio interpretato: Il codice quindi non viene compilato bensì c'è un interprete (in JavaScript lato client esso è incluso nel browser che si sta utilizzando) che esegue riga per riga, a tempo di esecuzione, quanto trascritto nello script.

JavaScript presenta le caratteristiche di un normale linguaggio interpretato (e di conseguenza i suoi vantaggi e svantaggi) con una sintassi analoga a quella di un linguaggio compilato (essa è relativamente simile a quella del C, del C++ e del Java), quindi con la possibilità di utilizzare funzionalità tipiche dei linguaggi di programmazione ad alto livello (strutture di controllo, cicli, etc.) e con in più anche la potenzialità di definire strutture più complesse, vicine a quelle adottate nei normali linguaggi object oriented (creazione di prototipi, istanziazione di oggetti, costruttori). JavaScript è un linguaggio debolmente tipizzato, ovvero il tipo delle variabili può non essere assegnato in fase di dichiarazione e le variabili stesse vengono convertite in maniera automatica dall'interprete.

4.3.4 Il plugin jQTouch

Le applicazioni web sono sicuramente la scelta più immediata per il gran numero di programmatori web esistenti, i quali, proprio per la natura di questo tipo di applicazioni, sono già pronti per portare le loro idee su diversi dispositivi mobili. Si introduce jQTouch [32], un plugin per JQuery[4] per lo sviluppo web su iPhone e iPodTouch. jQTouch è uno strumento OpenSource che aiuta a sviluppare applicazioni web per dispositivi mobili, in primis per l'iPhone. Con questa applicazione si possono creare delle semplici animazioni native, effettuare il preload delle immagini ed effettuare svariate operazioni di setup interne.

La caratteristicha più interesante di jQTouch è che permette di effettuare rapidamente pagine HTML che sembrano vere applicazioni native per l'iPhone. jQTouch consente di sviluppare rapidamente applicazioni che sfruttano i pattern dell'interfaccia utente sfruttando le competenze degli sviluppatori JavaScript. Grazie alle sue API molto semplici, jQTouch sta guadagnando popolarità nel mondo dei dispositivi mobili. jQTouch è una libreria

[4]jQuery è un framework basato su javascript che facilita non di poco lo sviluppo web offrendo una serie di strumenti per la gestione di animazioni, eventi, AJAX e quant'altro con l'uso di codice solido, semplice e soprattutto già pronto.

di codice sorgente che include gli standard Javascript e CSS. Per utilizzare jQTouch, è necessario che si strutturi il codice HTML in un modo specifico e che si seguano alcuni pattern specifici. Introduciamo due nozioni importanti che sono *"applicazione"* e *"schermo"* definendo il loro significato.

- *Schermo:* ciò che l'utente vede da una pagina all'altra. Ogni schermo si presume essere un elemento `<div>`[5] che è il figlio del corpo di HTML.
- *Applicazione:* la pagina HTML che include jQTouch, JavaScript e CSS, cosí come tutti gli schermi (alcuni dei quali possono essere caricati dinamicamente).

L'inizio di una nuova applicazione che utilizza jQTouch è semplice; tuttavia, la modifica delle applicazioni già esistenti è difficile perchè l'applicazione deve lavorare all'interno dei vincoli di jQTouch. Alcuni di questi vincoli sono:

- Non si può mai lasciare una pagina singola dell'applicazione.
- Gli URL devono avere percorsi completi.
- Ogni schermo non è una pagina web piena, invece, è un elemento "`<div>`" che è un figlio immediato del corpo dell'applicazione.
- Assicurarsi di non utilizzare gli "`<div> id =`", se non per identificare gli schermi.

Un aspetto interessante di jQTouch, è che questo plugin si può utilizzare sia tramite integrazione all'interno di applicazioni mobili web-based che di framework multipiattaforma (tipo Rhodes e PhonGap). Alcuni esempi di framework multipiattaforma riguardano l'inizializzazione e l'utilizzo di schermate, la loro modifica dinamica con AJAX, strumenti come le navigation

[5]Quando si vogliono riunire insieme più elementi all'interno di un contenitore strutturale unico, al fine di specificarne caratteristiche comuni si può usare l'elemento generico $< div >$ (ad es. quando si desidera definire l'allineamento rispetto ai margini della pagina di più elementi contemporaneamente).

bar, ecc. A questo punto, per la realizzazione di pagine web per lo smartphone iPhone, possiamo tenere in considerazione alcuni strumenti utili per velocizzare il lavoro di sviluppo, come ad esempio l'utilizzo del framework multipiattaforma PhoneGap e librerie JavaScript. Infine si sceglie di lavorare con jQtouch per la creazione di un semplice sito web ottimizzato per il dispositivo mobile iPhone.

Nel capitolo cinque, verrà mostrato l'integrazione del plugin jQTouch con il framework multipiattaforma PhoneGap.

4.4 Linguaggi di programmazione per le applicazioni native

I linguaggi di programmazione utilizzati dalle applicazioni native sono Objectice C, C, C++ e Java.

4.4.1 Objective C

Objective C è un linguaggio di programmazione orientato agli oggetti sviluppato alla metà degli anni ottanta presso la Stepstone Corporation, la cui diffusione è principalmente legata al framework OpenStep di NeXT ed al suo successore Cocoa, presente nei sistemi operativi Mac OS X e iOS di Apple (quindi nei dispositivi portatili iPod touch, iPhone, iPad ed applicazioni relative) [59]. Come lo stesso nome suggerisce, questo linguaggio è un'estensione ad oggetti del linguaggio di programmazione C: esso mantiene la completa compatibilità con quest'ultimo, conservando tutte le caratteristiche ed aggiungendone di nuove. Tra l'altro, anche a causa di questo, Objective C non è dotato di forte tipizzazione (caratteristica che invece esibiscono, ad esempio, sia C++ che Java). C è un sottoinsieme stretto del linguaggio in questione: ne consegue che è possibile compilare un qualsiasi programma scritto in C con un compilatore Objective C. La gran parte della sintassi è derivata da quella del predecessore, mentre quella relativa alle

caratteristiche object-oriented è stata creata per ottenere la comunicazione a scambio di messaggi. Essendo un linguaggio orientato agli oggetti l'Objective C fornisce tutte quelle istruzioni che permettono di implementare e definire una applicazione secondo il paradigma orientato agli oggetti.

4.4.2 C

Il C è un linguaggio di programmazione ad alto livello [69]. Tuttavia, poiché esso mantiene evidenti relazioni semantiche con il linguaggio macchina e l'assembly, risulta molto meno astratto di linguaggi anche affini (appartenenti allo stesso paradigma di programmazione), come per esempio il Pascal. Il C è un linguaggio molto efficiente, e si è imposto come linguaggio di riferimento per la realizzazione di software di sistema su gran parte delle piattaforme hardware moderne. La standardizzazione del linguaggio (da parte dell'ANSI prima e dell'ISO poi) garantisce la portabilità dei programmi scritti in C su qualsiasi piattaforma. Oltre che per il software di sistema, il C è stato a lungo il linguaggio dominante in tutta una serie di altri domini applicativi caratterizzati da forte enfasi sull'efficienza. Esempi tipici sono le telecomunicazioni, il controllo di processi industriali e il software real-time. Oggi il predominio del C in questi contesti è in parte diminuito a seguito dell'avvento di competitor significativi, primo fra tutti il C++; tuttavia, il tempo in cui il C si potrà considerare obsoleto appare ancora molto lontano.

4.4.3 C++

Il C++ è un linguaggio di programmazione orientato agli oggetti, con tipizzazione statica [70]. È stato sviluppato da Bjarne Stroustrup ai Bell Labs nel 1983 come un miglioramento del linguaggio C. I miglioramenti principali sono: l'introduzione del paradigma di programmazione a oggetti, funzioni virtuali, overloading degli operatori, ereditarietà multipla, template e gestione delle eccezioni.

Il C++ è molto diffuso e apprezzato, ma raramente è usato al massimo delle sue potenzialità: la semantica del C++ è molto ricca di dettagli e sfumature che condizionano il comportamento del codice, e che molto spesso i compilatori implementano in maniera scorretta o incompleta: molte delle caratteristiche dello standard ISO del linguaggio non sono ancora implementate nei compilatori attuali, anche se la situazione sta migliorando lentamente. La grande ricchezza semantica del C++, insieme alle librerie che lo accompagnano, lo rende un linguaggio estremamente espressivo e potente, ma che richiede molto tempo per venire appreso e padroneggiato completamente. Inoltre a causa della variabilità del comportamento dei compilatori nel maneggiare le funzioni più avanzate del linguaggio, i programmatori di C++ che scelgono di farne uso si rivolgono ad un'architettura (processore, sistema operativo e compilatore) particolare sacrificando la portabilità su altre piattaforme.

4.4.4 Java

Java è un linguaggio di programmazione orientato agli oggetti, creato da James Gosling e altri ingegneri di Sun Microsystems [71]. Java è stato creato a partire da ricerche effettuate alla Stanford University agli inizi degli anni Novanta. Nel 1992 nasce il linguaggio Oak, prodotto da Sun Microsystems. Tale nome fu successivamente cambiato in Java a causa di un problema di copyright. Per facilitare il passaggio a Java ai programmatori, legati in particolare a linguaggi come il C++, la sintassi di base (strutture di controllo, operatori e così via) è stata mantenuta pressoché identica a quella del C++; tuttavia, non sono state introdotte caratteristiche ritenute fonti di una complessità non necessaria a livello di linguaggio e che favoriscono l'introduzione di determinati bug durante la programmazione, come l'aritmetica dei puntatori, l'ereditarietà multipla delle classi, e l'istruzione goto. I programmi scritti in linguaggio Java, compilati in bytecode, sono destinati all'esecuzione sulla piattaforma Java, ovvero saranno lanciati su una Java Virtual Machine e, a tempo di esecuzione, avranno accesso alle API della libreria standard.

Ciò fornisce un livello di astrazione che permette alle applicazioni di essere interamente indipendenti dal sistema su cui esse saranno eseguite. Java è un linguaggio orientato agli oggetti, indipendente dalla piattaforma, contiene strumenti e librerie per il networking ed è progettato per eseguire codice da sorgenti remote in modo sicuro.

Capitolo 5

Framework di sviluppo multipiattaforma

In questo capitolo si introdurranno quattro framework di sviluppo multipiattaforma: Rhodes, PhoneGap, Titanium Mobile e Sencha Touch. Questi framework consentono di creare applicazioni per iOS, Android, BlackBerry, Windows Phone e Symbian.

5.1 Rhodes

Rhodes è un framework Ruby[1] open source creato per costruire rapidamente applicazioni native per tutti i principali smartphone (iOS, Android, RIM Blackberry, Windows Mobile e Symbian) [28]. Come linguaggi ausiliari Rhodes utilizza anche HTML, CSS e Javascript. Esso permette di costruire vere applicazioni native, e non applicazioni web, che permettono di sfruttare

[1] *Ruby* on Rails, spesso chiamato RoR o semplicemente Rails, è un framework open source per applicazioni web scritto in Ruby da David Heinemeier Hansson per conto della 37signals la cui architettura è fortemente ispirata al paradigma Model-View-Controller (MVC). I suoi obiettivi sono la semplicità e la possibilità di sviluppare applicazioni di concreto interesse con meno codice rispetto ad altri framework. Il tutto con necessità di configurazione minimale.

le funzionalià del dispositivo come: GPS[2], contatti e calendario, macchina fotografica, la mappatura narrativa, codice a barre, cattura della firma, Bluetooth e Near Field Communications[3]. Rhodes è l'unico framework nel suo genere a supportare l'MVC (Model View Controller) permettendo di separare

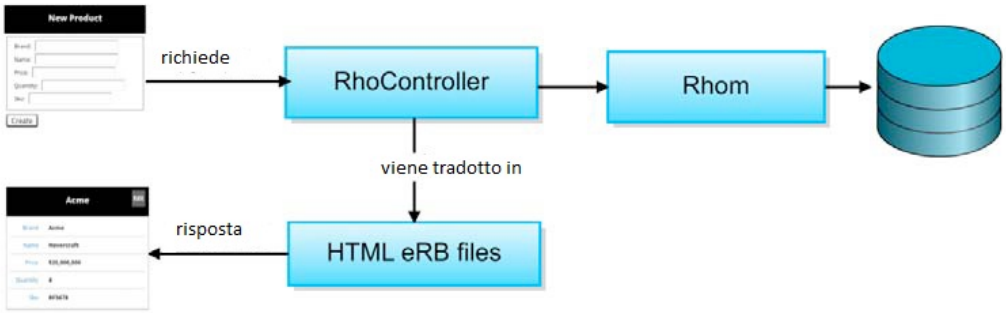

Figura 5.1: Model-View-Controller [28]

quindi la logica dell'applicazione dalla grafica. Questo aspetto ha costituito un punto di significante importanza nella scelta del framework. Sono inoltre disponibili due importanti servizi esclusivi di questo framework: il primo è un servizio di host chiamato RhoHub [29], che effettua la costruzione del prodotto senza imporre agli sviluppatori l'obbligo di attrezzare la propria piattaforma di lavoro per tutti i dispositivi mobile di cui intendono effettuare il deploy; il secondo è la connettività a RhoSync[4] [30], cioè un metodo veloce per permettere a qualsiasi applicazione smartphone di connettersi, recuperare, analizzare e gestire database delle loro applicazioni sfruttando le API degli smartphone moderni.

È semplice sviluppare Rhodes perchè c'è poco codice da scrivere.

[2]Acronimo di Global Positioning System: sistema di 24 satelliti costruiti dal Dipartimento della Difesa USA per il posizionamento e di libero accesso a chiunque sia dotato di un apposito ricevitore satellitare.

[3]*Near Field Communication* (NFC) è una tecnologia che fornisce connettività wireless bidirezionale a corto raggio. È stata sviluppata congiuntamente da Philips e Sony.

[4]*RhoSync* è la componente server di Rhodes e affiancato a Rhodes, permette di facilitare lo sviluppo quando è necessario far persistere i dati dell'applicazione.

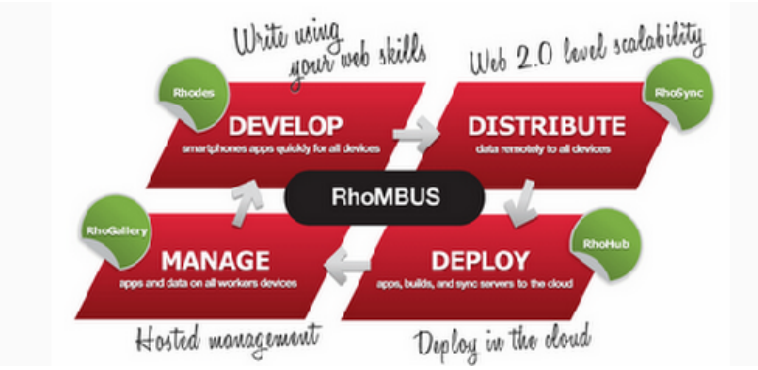

Figura 5.2: Rhodes, RhoSync, RhoHub, RhoGallery [44]

Si ha a disposizione anche un Object Relational Manager (ORM) chiamato Rhom, che sincronizza dati remoti usando RhoSync [44].

Di seguito si elencano le caratteristiche prinicipali di Rhodes che sono:

1. Supporto multipiattaforma (iOS, Android, Blackberry, ecc.).
2. Basato su Ruby.
3. Rhosync server personalizzabile.
4. Uso di tecnologie web comuni per la presentazione (HTML, CSS, JavaScript).
5. Open Source.

5.1.1 Funzionalità di Rhodes

Le funzionalità che Rhodes mette a disposizione degli sviluppatori sono descritte nella Figura 5.3 [45].

Come si può notare dalla tabella (Figura 5.3), la quasi totalità delle funzioni è già implementata per tutte le piattaforme di maggior rilievo. Viene fatta eccezione per Windows Phone 7, che è una piattaforma più recente rispetto alle altre, e per Symbian il cui supporto è stato introdotto solo con la versione 3.1.0 di Rhodes, rilasciata nel mese di agosto 2011. Si può comunque notare che le funzioni sono in via di sviluppo.

Capability	iOS	Windows Mobile	Windows Phone	BlackBerry	Android	RhoSimulator	Symbian
GeoLocation	0.3	0.3	TBD	0.3	1.0	3.5	TBD
PIM Contacts	0.3	0.3	TBD	0.3	1.0	3.5	3.1
PIM Calendar	2.2	2.2	TBD	2.2	2.2	3.5	TBD
Camera	1.0	1.0	TBD	1.0	1.0	3.0	TBD
Barcode	2.1	2.1	TBD	2.1	2.1	4.0	TBD
Date/Time picker	1.2.2	2.0	TBD	1.2	1.2	3.0	TBD
Menu	1.2.2	2.0	3.0	1.2	1.5	3.0	3.1
Toolbar	1.2.2	2.3	3.0	n/a	1.5	3.0	3.1
Tab Bar	1.2.2	3.5	3.0	n/a	1.5	3.1	3.1
Nav Bar	2.0	3.5	TBD	n/a	2.0	4.0	TBD
Signature Capture	2.1	3.0	TBD	3.5	2.1	4.0	TBD
Audio/Video capture	4.0	4.0	TBD	4.0	4.0	4.0	TBD
Bluetooth	2.2	2.2	TBD	2.2	2.2	4.0	TBD
NFC	TBD	TBD	TBD	TBD	3.0	4.0	TBD
Push	1.2	4.0	TBD	1.2	2.2	3.5	TBD
Screen rotation	2.1	3.5	TBD	2.0	2.1	3.5	TBD
Native Maps	1.4	3.5	TBD	1.4	1.5	3.5	TBD
Alerts/Audio File Playback	1.2	1.5	TBD	1.2	1.2	3.1	TBD
Ringtones	3.5	1.5	TBD	1.5	1.5	3.5	TBD

Figura 5.3: Funzionalità di Rhodes [45]

Altre caratteristiche di Rhodes vengono descritte nella Figura 5.4.

5.1.2 Struttura base delle applicazioni

Le applicazioni create di base da Rhodes sono strutturate in poche semplici cartelle molto intuitive ed ordinate. Sono presenti infatti le seguiti suddivisioni [46]:

- cartella “app”: contiene effettivamente tutta la parte logica e grafica che servono alla creazione dell’applicazione. Al suo interno sono presenti delle ulteriori sottocartelle che dipendono dall’applicazione in se in quanto esse vengono create nel momento in cui viene aggiunto un Model. All’interno di quest’ultime si trova, per ognuna, il controller, il model e tutte le viste necessarie per comporre quel modulo.

	Rhodes	Legacy Providers	Me-Too Frameworks
	Framework Features		
Supports ALL major smartphones	Yes	No	No
Model View Controller	Yes	No	No
App Language	Standard	Proprietary	Standard
App Store Accepts	Yes	No	Yes
Barcode/Signature Capture/Bluetooth	Yes	Yes	No
HTML5 Support	Yes	No	Yes
Hosted Offering	Yes	No	No
	Offline Data Synchronization		
Basic Sync	Yes	Yes	No
Metadata Layer	Yes	No	No
Uses Native Push APIs	Yes	No	No
Hosted SaaS Option	Yes	No	No
	Development Environment OS Support		
Windows	Yes	Yes	Yes (no iPhone)
Mac	Yes	Yes	Yes
Linux	Yes	No	Yes (no iPhone)

Figura 5.4: Caratteristiche di Rhodes [45]

- cartella "bin": il contenuto principale è composto da tutte le librerie ed estensioni che Rhodes utilizza per permettere la corretta implementazione delle sue funzionalità. Il resto del contenuto è di importanza minore;

- cartella "icon": come dice il nome stesso, questa cartella ha il solo scopo di contenere le icone utilizzate;

- cartella "public": contiene i file che si occupano principalmente di fornire un aspetto grafico, come i file CSS e le immagini, e i file Javascript, propri di Rhodes oppure quelli di jquery, utilizzati nel progetto.

- file "build.yml": specifica tutti i parametri necessari nella fase di build, in particolare vengono inserite anche le estensioni utilizzate;

- file "Rakefile": è il file di make, specifio del linguaggio Ruby, che permette la build automatica dell'applicazione;

- file "rhoconfig.txt": contiene tutti gli input di configurazione propri di Rhodes;

- file "rholog.txt": questo file contiene una copia, permanente, del contenuto della console presente su Eclipse nella fase di compilazione ed esecuzione dell'applicazione.

Architettura a run-time

I file di sviluppo di Rhodes vengono compilati in un eseguibile nativo che è installato sul dispositivo oppure viene eseguito un simulatore oppure interfacce web sul sito **rhohub.com** [47].

Dal momento che le applicazioni di Rhodes sono applicazioni binarie, possono essere presentate e distribuite attraverso l'iTunes App Store di Apple, Blackberry World, il mercato Android, e altri canali di distribuzione. Per costruire un dispositivo, è in genere necessario firmare per quei programmi di sviluppo e acquisire chiavi di crittografia richieste per firmare le applicazioni, anche se non verrà scritto niente sulle piattaforme dell'SDK. Sulle piattaforme dove la lingua principale di sviluppo è Java, come Blackberry, le applicazioni di Rhodes vengono precompilate in Java bytecode per poi essere eseguiti. Sulle piattaforme iPhone, Android, Windows Mobile e Symbian, le applicazioni di Rhodes vengono compilate in Ruby 1.9 bytecode. In queste piattaforme, Rhodes include un esecutore Ruby che esegue il bytecode sul dispositivo. Per connettere le applicazioni di Rhodes con le applicazioni web, è possibile utilizzare RhoSync oppure connettersi direttamente. Il collegamente è possibile direttamente tramite JavaScript, usando il net/ http library, oppure l'ottimizzazione di Rho: AsyncHttp.

Anche se Ruby è un linguaggio interpretato, usando Rhodes, non è possibile eseguire codice Ruby a runtime utilizzando, ad esempio, la stringa

"eval"[5]. Questa capacità è stata intenzionalmente rimossa dall'interprete Ruby per conformarsi con la regola dell'iPhone App Store che dice:

"Un'applicazione non può in sè installare oppure lanciare un altro codice eseguibile, includendo senza limitazioni attraverso l'uso di un architettura plug-in, chiamando altri framework, altri API oppure altro. Nessun codice interpretato può essere scaricato e utilizzato in un'applicazione tranne per il codice che viene interpretato e eseguito da le API di Apple e interpreti integrati."

www.rhomobile.com/rhodes/iphone-app-store-rules-and-guidelines-on-use-of-frameworks [31]

Rhodes è un'applicazione completamente nativa e incorpora i browser nel dispositivo. Dato questo fatto, ci sono delle implicazioni per il markup, CSS, e JavaScript che possono essere supportati in ogni piattaforma. Alcuni dispositivi, come l'iPhone e Android, hanno dei browser con multi caratteristiche altri invece, come Blackberry no. Questo significa che non si può scrivere HTML e CSS che sfruttano le caratteristiche di uno specifico browser su un dispositivo e aspettarsi che funzioni su altri dispositivi con web browser meno capaci. Le applicazioni di Rhodes, invece, vengono sviluppate in modo simile alle applicazioni web, che vengono eseguite localmente come applicazioni native e non da remoto come le applicazioni web. Sia l'elaborazione che l'accesso al database vengono gestite localmente.

Il database - Rhom

Rhom è un mini database per Rhodes [32]. Fornisce un'interfaccia di alto livello per rendere l'utilizzo del database locale più potente e semplice. Stiamo parlando del database SQLite che è presente su tutte le piattaforme tranne BlackBerry che utilizza il database HSQLDB.

Rhom supporta attualmente due tipi di modelli: **Property Bag** e **Fixed Schema**.

[5]La stringa *eval* trasforma la stringa racchiusa tra parentesi tonde in codice JavaScript.

Property Bag [48]

Con un modello "Property Bag", tutti i dati vengono memorizzati in una singola tabella usando il pattern oggetto-attributo-valore riferito anche come il modello entità-attributo-valore.

I vantaggi del **Property Bag** sono:

- Semplice da usare, non necessita la specifica degli attributi.
- La migrazione dei dati non è necessaria.
- Gli attributi possono essere aggiunti o rimossi senza modificare lo schema del database.

Gli svantaggi del **Property Bag** sono:

- Per alcune applicazioni, le dimensioni del database possono essere significativamente più grandi dello schema fisso. Questo perché ogni attributo è indicizzato per la ricerca veloce.
- Il processo di sincronizzazione può essere leggermente più lento perché gli inserimenti vengono fatti al livello dei attributi.

In un modello **Property Bag**, Rhom ragruppa gli oggetti in `source id` e `object id`. Il seguente esempio illustra questo concetto:

```
Source ID: 1, Model Name: Account
+-----------+----------+--------------+----------------------+
| source_id | attrib   | object       | value                |
+-----------+----------+--------------+------- --------------+
|         1 | name     | 48f39f63741b | A.G. Parr PLC 37862  |
|         1 | industry | 48f39f63741b | Entertainment        |
|         1 | name     | 48f39f230529 | Jones Group          |
|         1 | industry | 48f39f230529 | Sales                |
+-----------+----------+--------------+----------------------+
```

Rhom esporrà una classe *Account* con due attributi: `name` e `industry`.

```
account = Account.find('48f39f63741b')
account.name
  #=> "A.G. Parr PLC 37862"

account.industry
  #=> "Entertainment"
```

Fixed Schema [48]

Con un modello di schema fisso, ogni modello ha una tabella di database separata e ogni attributo esiste come una colonna nella tabella. In questo senso, i modelli di schema fisso sono simili alle tabelle relazionali.

I vantaggi di **Fixed Schema** sono:

- Le dimensioni del database sono più piccole, gli indici possono essere specificati solo su attributi specifici
- Il processo di sincronizzazione può essere eseguito più velocemente perchè interi oggetti vengono inseriti in un momento.

I svantaggi di **Fixed Schema** sono:

- Le modifiche dello schema devono essere gestiti con la migrazione dei dati
- Le prestazioni del database possono essere lenti se non si specificano gli indici corretti.

L'obiettivo principale di Rhom è quello di fornire un modello di interfaccia semplice e intuitiva per un'applicazione di Rhodes.

Differenze tra Rhodes e Rails

- Rhodes è ispirato da Ruby ma è significamente più piccolo e semplice.
- Non ci sono directory separate per modelli, controller e le viste. Ogni modello è nella propria directory.

- Non è possibile eseguire le applicazioni di Rhodes in modo interattivo usando gli script/console. È necessario compilare il codice e installarlo nel simulatore per eseguirlo.

- Molte differenze da Rails rendono più facile il funzionamento su dispositivi mobili con memoria limitata. Rhodes è più leggero perché offre funzioni di base necessarie.

RhoSync

Si parla della componente server del loro pacchetto, ossia RhoSync: affiancato a Rhodes, permette di facilitare lo sviluppo quando è necessario far persistere i dati dell'applicazione e non ci si vuole impegnare nelle politiche di gestione diretta di un database. Per RhoSync è prevista sia l'esistenza di un database (leggasi "servizio") remoto, sia le primitive per accedervi, le operazioni di ricezione (anche in modalità di sincronizzazione) e di inserimento (aggiornamento ed eliminazione) e l'autenticazione per gli utenti.

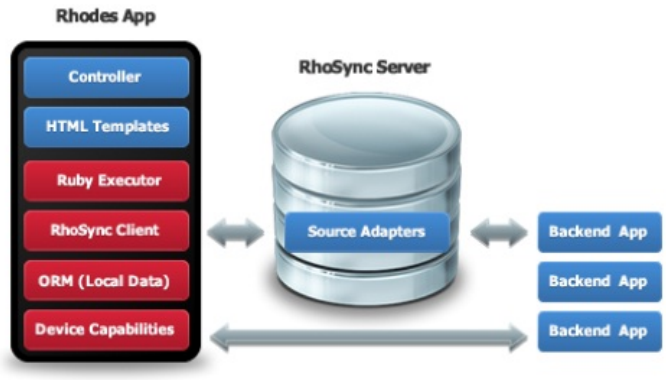

Figura 5.5: Rhosync [30]

5.2 PhoneGap

Phonegap è un progetto Open Source della Nitobi Software [33], un'azienda che crea applicazioni per dispositivi mobili e applicazioni web da più di

dieci anni. Consiste in un insieme di librerie statiche che permettono di sviluppare velocemente ed efficacemente applicazioni per dispositivi mobili di diverse famiglie. L'idea fondamentale di questo progetto è cercare di realizzare lo slogan "Write once, port everywhere". Phonegap si propone di focalizzare gli sforzi degli sviluppatori sull'applicazione piuttosto che perdere tempo ad adattarla ad ogni piattaforma. Per fare ciò un'applicazione realizzata con Phonegap richiede solo la conoscenza di HTML, CSS e Javascript. Il porting verso le varie piattaforme viene fatto installando gli ambienti di sviluppo relativi e compilando la webapp realizzata[6]. I requisiti sono quindi di installare le SDK, dove richiesto dalla piattaforma, e gli strumenti per consentire la compilazione delle applicazioni.

5.2.1 Vantaggi di PhoneGap

Sviluppare un'applicazione per tutte le maggiori piattaforme richiede la conoscenza dei linguaggi delle stesse, l'installazione di IDE[7] per lo sviluppo, di personale in grado di seguire lo sviluppo. Una pianificazione per l'ingegnerizzazione del software è d'obbligo per garantire la realizzazione di un prodotto uniforme malgrado le eterogeneità dei dispositivi. La realizzazione di un'applicazione su varie piattaforme richiede quindi tempo e denaro, per reclutare il personale, formarlo se necessario, e preparare le piattaforme di sviluppo per ogni piattaforma [34]. Rivolgersi al mercato in questo modo implica l'impiego di importanti risorse che potrebbero essere volte allo sviluppo dell'applicazione stessa o ad una fase di test più accurata. A causa delle consistenti risorse da dover mettere in gioco ciò che alcune aziende fanno è sviluppare la propria applicazione solo per le piattaforme più utilizzate, o concentrare gli sforzi sulle piattaforme con meno restrizioni. Ciò ovvia-

[6] A seconda della piattaforma e della versione di Phonegap utilizzata, può essere necessario adattare o ridurre l'impiego di codice javascript.

[7] *Integrated Development Enviroment o Ambiente di Sviluppo Integrato*; normalmente consiste in un editor di codice sorgente, un compilatore e/o un interprete, un tool di building automatico, e un debugger.

mente porterebbe a ridurre il potenziale mercato dell'applicazione oltre che impedire automaticamente agli utenti delle altre piattaforme di utilizzarla. Con Phonegap lo sviluppatore può dedicarsi maggiormente allo sviluppo della sua applicazione web senza pensare agli intralci o impedimenti delle varie piattaforme. Chiaramente le possibilità dell'applicazione saranno limitate a quelle di una applicazione web[8].

Alcuni aspetti chiave da valutare per decidere se sfruttare il framework PhoneGap per il proprio progetto sono:

- sebbene tutte le piattaforme si stiano allineando agli standard di HTML5, alcune permettono altre funzionalità che vanno oltre quelle offerte da Phonegap. Occorre valutare quali sono le esigenze e tenere conto della possibilità di utilizzare queste possibilità.

- La fase di sviluppo, malgrado non richieda particolari attenzioni per la stesura della/e pagina/e HTML, necessita di qualche attenzione nello sviluppo della parte Javascript. Ogni piattaforma ha il suo browser nativo e gestisce il codice in maniera diversa.

- A prescindere dal sistema operativo della piattaforma obiettivo, è d'obbligo una fase rigorosa di test per verificare il funzionamento.

- Come prospettive future è auspicabile che ogni piattaforma migliori il proprio browser nativo. Migliorando il rendering delle pagine web si migliorano le applicazioni realizzate in Phonegap.

- Qualora si disponesse di un'applicazione web funzionante, magari già sviluppata sfruttando HTML5, il porting verso le piattaforme mobili diventa molto agevole.

- Come nota finale si può aggiungere che le piattaforme coperte da PhoneGap sono al momento la quasi totalità del mercato.

[8]Sono utilizzabili anche i comandi javascript che mette a disposizione il browser nativo.

Figura 5.6: Sistemi operativi supportati dalla piattaforma Phonegap

5.2.2 Architettura di PhoneGap

Phonegap fà da ponte tra il sistema operativo e l'applicazione web realizzata dallo sviluppatore (Figura 5.7) [32]. Si intuisce che occorre che dall'applicazione web esista una modalità standard per invocare le API native in maniera indipendente dal tipo di piattaforma sottostante. PhoneGap è infatti un framework che permette ad un'applicazione web di invocare le API native mediante funzioni JavaScript di cui è possibile vedere gli esempi forniti sul sito di riferimento.

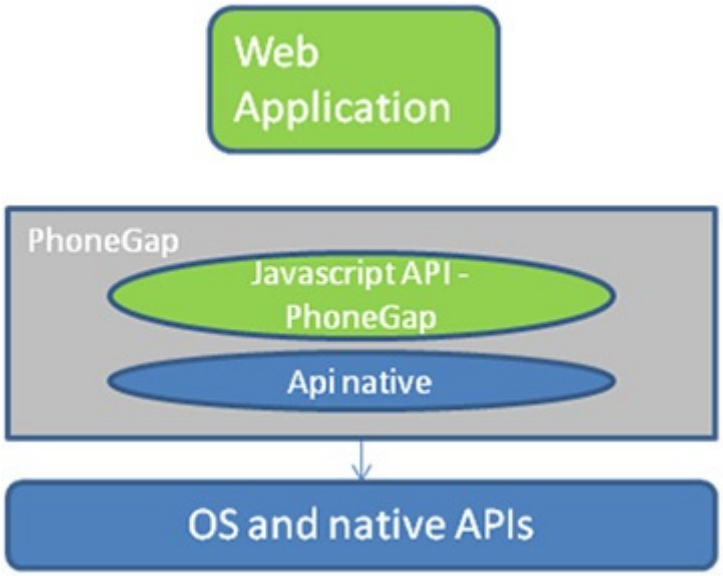

Figura 5.7: Architettura di Phonegap [32]

Il vantaggio nell'utilizzo di questo framework è che esso fornisce l'aggancio tra la piattaforma nativa e l'applicazione web, di modo che uno sviluppatore

che non ha conoscenze specifiche sui linguaggi nativi, possa dedicarsi allo sviluppo dell'applicazione web con tecnologie standard. La caratteristica fondamentale di PhoneGap sta nel fatto che, una volta realizzata l'applicazione per il mobile con questo framework, l'applicazione si adatta a qualunque piattaforma supportata. PhoneGap userà il linguaggio nativo della piattaforma per accedere alle risorse hardware e software in modo da aggiungere le funzionalità di base al motore JavaScript e renderle così facilmente utilizzabili dall'applicazione come fossero tradizionali metodi di libreria. In base alla tipologia di piattaforma con la quale dovrà interfacciarsi, l'implementazione di aggancio sarà di conseguenza sviluppata in Objective C per iPhone, in Java per Android, e così via; e tale implementazione è fornita dallo stesso framework. In pratica esiste un runtime basato su WebKit4 in cui vengono iniettate le componenti statiche. Il risultato sarà un pacchetto composto di due elementi principali con differenti responsabilità che però cooperano tra loro per fornire delle funzioni a valore aggiunto. Nel caso specifico il runtime si occupa di dialogare direttamente con il dispositivo e le parti statiche offrono l'interfaccia verso l'utente. L'uso di JavaScript e di Ajax[9] consente poi di dare vita alle applicazioni che possono avere comportamenti complessi e comunicazioni verso endpoint remoti realizzando di fatto un'applicazione completa in cui il client è un elemento fondamentale in linea con le moderne applicazioni Web 2.0 (e in contrapposizione a quanto accadeva solo pochi anni fa per le applicazioni web la cui interfaccia era completamente elaborata server side).

La Figura 5.8 mostra come è strutturata l'architettura di Phonegap.

[9] *AJAX*, acronimo di Asynchronous JavaScript and XML, è una tecnica di sviluppo per la realizzazione di applicazioni web interattive (Rich Internet Application). Lo sviluppo di applicazioni HTML con AJAX si basa su uno scambio di dati in background fra web browser e server, che consente l'aggiornamento dinamico di una pagina web senza esplicito ricaricamento da parte dell'utente. AJAX è asincrono nel senso che i dati extra sono richiesti al server e caricati in background senza interferire con il comportamento della pagina esistente. Normalmente le funzioni richiamate sono scritte con il linguaggio JavaScript.

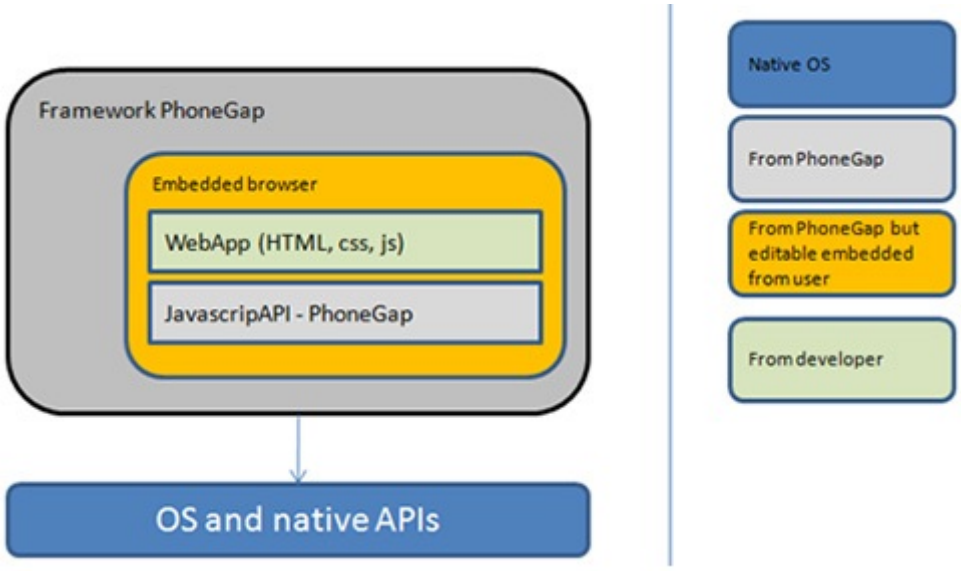

Figura 5.8: Architettura di una applicazione realizzata con Phonegap [32]

Partendo dal basso verso l'alto, si può notare che la parte in blu è quella del sistema operativo della piattaforma nativa e come tale viene fornita. Al livello immediatamente sopra troviamo il framework PhoneGap (in grigio) e anch'esso ci viene fornito insieme alle API JavaScript. È curioso notare il rettangolo in arancione che incapsula l'applicazione web e le API JavaScript. Questo è stato fatto per mettere in evidenza che l'oggetto browser può essere aperto all'interno dell'applicazione che si sta sviluppando e dunque dentro PhoneGap. Per comportamento di default il browser viene aperto esternamente all'applicazione. Di conseguenza ad ogni richiesta di una pagina web si ha l'apertura del web browser. L'apertura di una pagina nel browser embedded nasconde la barra degli indirizzi, i bottoni forniti dal browser di default. In questo modo si oscura relativamente il fatto che si tratti di una pagina web, vantaggio indiscutibile nel rendere l'esperienza più simile a quella di un'applicazione nativa. La scelta di aprirlo o meno all'interno dell'applicazione sta allo sviluppatore e nel momento in cui decidesse di aprirlo all'interno dell'applicazione deve apportare delle semplici modifiche al codice di aggancio nativo. Infine, in verde è evidenziata la parte a carico dello sviluppatore che

è l'applicazione web, realizzata all'interno di PhoneGap. Per cui, il pacchetto applicativo destinato alla distribuzione conterrà file statici come HTML, JavaScript, CSS e le classi native di aggancio. Il pacchetto dovrà essere opportunamente firmato prima di essere distribuito sullo store ufficiale delle piattaforme di riferimento.

5.2.3 Creare un'applicazione con PhoneGap

In questa sezione verrà mostrato come creare un'applicazione con PhoneGap e come estendere le sue funzionalità, concentrando il lavoro sulla piattaforma Android.

Un esempio più esteso viene presentato nell'Appendice A per la piattaforma Symbian.

Si inizia impostando l'ambiente di lavoro. Prima di tutto è necessario scaricare PhoneGap, dopodichè, una volta creato il progetto, è necessario aggiungere la libreria esterna dedicata ad Android. Proviamo ora a creare un **"Hello World"** di prova. Si copia la cartella `xml`, contenente la lista dei plugin attivi, presente nel file zip di Phonegap, nella cartella di risorse `res`. Si crea quindi una cartella chiamata `www`, come sottocartella della directory `assets` del nostro progetto, e si copia la libreria `phonegap-1.0.0.js` all'interno di essa. Si crea una semplice pagina HTML con scritto **"Hello World!"** impostando anche uno stile adeguato. Questa cartella rappresenterà la root della nostra applicazione.

Si crea dunque un nuovo progetto Android con una nuova activity `HelloGapActivity` importando `com.phonegap.*`; estendendo da DroidGap. Infine sarà sufficente richiamare:

```
super.loadUrl("file:///android_asset/www/index.html");
```

Da questo momento si può dedicare l'attenzione escusivamente alla parte HTML. Nell'header della pagina includiamo `phonegap-1.0.0.js` e un semplice script di prova che utilizza PhoneGap per accedere alle informazioni del dispositivo:

```
<script type="text/javascript" charset="utf-8">
    var deviceInfo = function() {
        document.getElementById("platform").innerHTML = device.platform;
        document.getElementById("version").innerHTML = device.version;
        document.getElementById("uuid").innerHTML = device.uuid;
        document.getElementById("name").innerHTML = device.name;
        document.getElementById("width").innerHTML = screen.width;
        document.getElementById("height").innerHTML = screen.height;
        document.getElementById("colorDepth").innerHTML = screen.colorDepth;
    };

    function init() {
        document.addEventListener("deviceready", deviceInfo, true);
    };
</script>
```

Nella funzione `init()`, richiamata al load del body, si aggiunge un listener all'evento `deviceready` che chiamerà `deviceInfo()` appena il dispositivo sarà pronto. Questa funzione non fa altro che accedere alle informazioni del device sul quale sta girando l'applicazione e sul relatvo schermo. Le informazioni sono contenute in due apposite strutture screen e device. Viene lanciato l'applicazione e si ha un risultato simile alla Figura 5.9:

Figura 5.9: Hello World con PhoneGap

Questo piccolo esempio, oltre ad introdurre PhoneGap, mostra tutta la semplicità di questo strumento: accedere alle funzioni del dispositivo è semplice e si baserà spesso su chiamate asincrone e relative definizioni di callback

in caso di successo e di insuccesso, alle quali verrano passati dei parametri che potremo riutilizzare.

Lo sviluppo

Lo sviluppo di una vera e propria applicazione verrà costruita interamente con HTML, CSS e Javascript. L'applicazione in questione, ha lo scopo di scattare una foto e selezionare una cornice da sovrapporre ad essa.

Infine tale foto sarà salvata nella sdcard del dispositivo. Si vedrà come jQuery potrà esserci d'aiuto per migliorare l'esperienza utente nella nostra applicazione.

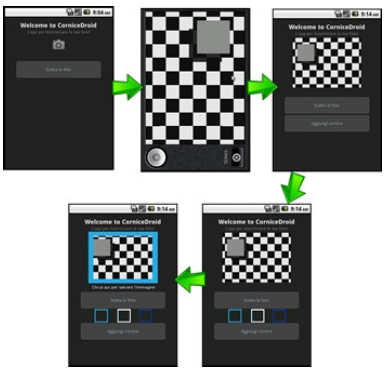

Figura 5.10: Il ciclo di vita della nostra applicazione in 5 screenshots

Impostiamo nella pagina di index il layout, includendo un apposito CSS, le librerie jQuery, Phonegap e tutte le altre librerie utilizzate di contorno, ad esempio *Pixastic* per l'editing delle immagini e *Slide* per creare appunto l'effetto slide. Si hanno quindi due pulsanti, quello per scattare la foto e quello per visualizzare la galleria delle cornici che inizialmente risulterà invisibile. Il primo pulsante richiamerà la funzione `show_pic` che tramite PhoneGap si collegherà alla nostra fotocamera, attraverso questo codice:

```
navigator.camera.getPicture(dump_pic, fail, {
    quality : 50, destinationType: Camera.DestinationType.FILE_URI
});
```

PhoneGap mette a disposizione l'oggetto *Camera* per eseguire tutte le operazioni con la fotocamera del nostro dispositivo. Basta richiamare il metodo `getPicture` passandogli due callback, la prima in caso di successo, la seconda in caso di fallimento, e una lista di parametri, la qualità dell'immagine e il tipo di risorsa restituita alla callback di successo.

`Camera.DestinationType.FILE_URI` specifica l'URI del file di destinazione, in alternativa potremmo farci restituire i dati e applicarli al campo `src` del canvas. Si definisce la funzione `callback` di successo.

```
function dump_pic(file) {
    resizeImg(file);
}
```

Come parametro in ingresso riceve l'URI di cui sopra, che viene passata alla routine di `resizeImg` per ridimensionare l'immagine altrimenti troppo grande. La routine di `fail` si limita invece a loggare il messaggio di errore e lanciare un alert con tale messaggio.

```
function fail(msg) {
    console.log(msg.code);
    alert(msg);
}
```

In questo caso si è utilizzato la funzione javascript `alert`, ma è possibile ricorrere ad oggetti PhoneGap per avere una finestra più strutturata con titolo e testo per il pulsante, come nel seguente esempio:

```
function fail(msg) {
    console.log(msg.code);
    try{
        navigator.notification.alert
        (msg, alertCallback, "Titolo", "Testo del pulsante");
    }
    catch (e) {
        alert(msg);
    }
}
```

Il blocco `try catch` permette di poter eseguire questa funzione anche su un normale browser che non dispone di PhoneGap. Infine per loggare qualsiasi messaggio basta ricorrere all'oggetto console e richiamare il metodo log.

Torniamo al nostro programma. Nella funzione `resizeImg` si rende visibile il pulsante delle cornici e si mostra nella pagina la nostra foto appena scattata e ridimensionata. Come mostrato nella pagina index, tra il pulsante per scattare una foto e quello per visualizzare le cornici, abbiamo inserito 3 miniature delle cornici racchiuse in un tag `<ul>`: questo trucco ci permette di far comparire le miniature con l'effetto slide al click del pulsante.

Ognuna delle miniature ha associata la funzione blending al click, procedura che, sempre sfruttando pixastic, permetterà di sovrapporre la cornice scelta all'immagine appena fotografata, passando come parametro l'immagine intera che si trova nella cartella `cornici`. Come ultima funzionalità vogliamo avere la possibilità di salvare la nostra foto con cornice annessa, ma l'unica possibilità per farlo è quella di creare un proprio plugin per estendere PhoneGap.

PhoneGap è un strumento di sviluppo per il mondo mobile, la sua estendibilità e semplicità, la possibilità di utilizzare linguaggi come Javascript, HTML e CSS, dovrebbe permettere un aumento del numero di sviluppatori in questo nuovo settore.

5.2.4 Integrazione di jQTouch con il framework PhoneGap

Per usare le caratteristiche di jQTouch in un'applicazione Phonegap, si devono copiare le cartelle `jQTouch/` e `themes/` nella cartella *www* dell'applicazione PhoneGap. Nel file `index.html` si sostituisce il codice CSS e JavaScript nella sezione HEAD come di seguito mostrato:

```
<link rel="stylesheet" href="jqtouch/jqtouch.min.css"
type="text/css" media="screen"
```

```
title="no title" charset="utf-8">
<link rel="stylesheet" href="themes/apple/theme.min.css"
type="text/css"
media="screen" title="no title" charset="utf-8">
<script src="jqtouch/jquery.1.3.2.min.js"
type="text/javascript" charset=
"utf-8"></script>
<script src="jqtouch/jqtouch.min.js"
type="text/javascript" charset="utf-8"></script>
<script>
var jQT = $.jQTouch();
</script>
```

La prima operazione da compiere per realizzare un sito Web destinato alla navigazione su iPhone e basato su jQtouch, sarà naturalmente quella di scaricare il plugin dal sito ufficiale del progetto [61]; l'archivio compresso che lo contiene è circa 42 MB, fortunatamente però queste dimensioni sono dovute in particolare alla presenza di alcune demo, l'estensione in sé richiede invece pochi Kb di spazio. Una volta eseguito il download, bisognerà scompattare l'archivio nella cartella destinata ad ospitare il sito Web, per cui si potrà per esempio creare una directory denominata *"mobile"* all'interno della root del Web server, in questo modo la cartella potrà essere utilizzata anche per la definizione del percorso ad un sotto dominio (ad esempio *"mobile.sito.it"*).

Fondamentalmente l'estensione fornisce le seguenti risorse:

- il plugin (*jqtouch.js*), presente nella cartella `/jqtouch`; in essa sono stati salvati anche jQuery e il file CSS di jQtouch;
- la cartella `/themes` in cui sono presenti i fogli di stile e le immagini dei temi messi a disposizione nativamente dall'estensione;
- delle estensioni destinate all'integrazione di funzionalità addizionali (cartella `/extensions`);

- una cartella denominata `demos` che contiene alcuni esempi pratici sull'utilizzo del plugin;
- un esempio di file `.htaccess` che potrà essere personalizzato sulla base delle diverse esigenze `sample.htaccess`.

L'obiettivo dell'esempio proposto sarà quello di creare un sito web associando alla navigazione tra le pagine un effetto di transizione a scorrimento in senso orizzontale; a questo scopo basterà creare un unico file, che potrà essere chiamato `index.html`, destinato all'intero contenuto, i link saranno quindi degli ancoraggi tra i diversi elementi presenti.

Creato il file, sarà possibile procedere con la digitazione del codice di pagina:

```
<!doctype html>
<html>
<head>
<meta charset="UTF-8" />
<title>Il mio primo sito Web per iPhone</title>
<style type="text/css" media="screen"
>@import "jqtouch/jqtouch.css";</style>
<style type="text/css" media="screen"
>@import "themes/apple/theme.css";</style>
<style type="text/css" media="screen">
    ul.edgetoedge li a{
    background: url(themes/apple/img/chevron.png)
    right center no-repeat,
    -webkit-gradient(linear, 0% 0%, 0% 100%,
    from(#fff5d8), to(#ffeab1));
    }
</style>
```

La dichiarazione del `doctype` sarà quella di HTML5, non ancora considerata uno standard dal W3C ma supportata dai browser più diffusi. La

codifica Unicode utilizzata sarà invece UTF-8, che permetterà di rappresentare praticamente tutti i caratteri. Verranno poi importati alcuni fogli di stile, il primo sarà quello relativo al plugin stesso, il secondo sarà invece associato al tema scelto, che, nel caso specifico dell'esempio proposto, è chiamato ‘‘Apple’’. Il codice presenta anche una regola CSS personalizzata, essa consentirà di associare ai link/punti elenco uno sfondo con sfumatura e l'immagine di una freccia che evidenzi la presenza di un collegamento. Il codice javascript invece sarà:

```
<script src="jqtouch/jquery-1.4.2.min.js" type="text/javascript"
charset="utf-8"></script>
<script src="jqtouch/jqtouch.js" type="application/x-javascript"
charset="utf-8"></script>
<script type="text/javascript" charset="utf-8">
    var jQT = new $.jQTouch({
        icon: 'jqtouch.png',
        addGlossToIcon: false,
        startupScreen: 'jqt_startup.png',
        statusBar: 'black'
    });
</script>
```

dove

- *icon*: consente di definire un icona per un'applicazione;
- *addGlossToIcon* è una proprietà è accetta due possibili opzioni: true, verrà quindi aggiunto l'effetto gloss (brillantezza) all'icona, e false, nel caso non si desideri questo effetto;
- *startupScreen*: indica il percorso assoluto o relativo all'immagine per la schermata home;
- *statusBar*: associa un colore come attributo di sitle alla barra di stato, sono accettati i valori default, black-translucent e black;

la parte successiva riguarderà la home page del sito web:

```
</head>
<body>
<div id="home">
    <div class="toolbar">
        <h1>iPhone Website</h1>
    </div>
    <ul class="edgetoedge">
        <li><a href="#home">Home</a></li>
        <li><a href="#whoami">Chi sono</a></li>
        <li><a href="#contacts">Contatti</a></li>
        <li><a href="#copyleft">Copyleft</a></li>
    </ul>
</div>
```

Come si noterà anche analizzando le parti successive, la pagina avrà un struttura "a blocchi", questo perché il dispositivo verifica l'inclusione di un contenuto all'interno di un `<div>`, di un elenco o di un paragrafo e procede con l'espansione del blocco a tutto schermo, una struttura come quella proposta faciliterà quindi l'utilizzatore nelle operazioni di ingrandimento.

Il secondo blocco riguarderà invece la pagina `#whoami`, l'unica differenza con quella precedente sarà il fatto che i punti in elenco non saranno anche dei link ma dei semplici contenuti testuali. Questa configurazione permetterà di visualizzare gli elementi con un alto stile rispetto all'immagine precedente. Ecco il codice:

```
<div id="chisono">
<div class="toolbar">
<a href="#" class="back">back</a>
<h1>Chi sono</h1>
</div>
<div class="info">Main </div>
```

```
<ul class="edgetoedge">
<li>Fist</li>
<li>Second</li>
<li>Third</li>
</ul>
</div>
```

Il blocco di chiusura sarà costituito da una semplice pagina informativa e verrà seguito dal markup di chiusura.

```
<div id="copyleft">
<div class="toolbar">
<a href="#" class="back">back</a>
<h1>Copyleft</h1>
</div>
<div class="info">All</div>
</div>
</body>
</html>
```

5.3 Titanium Mobile

Le tecnologie web sono una delle soluzioni più usate negli ultimi anni per creare applicazioni in quanto:

- Le applicazioni sono facili da sviluppare e mantenere.
- Il web offre la possibilità di sviluppare applicazioni cross-platform.

Partendo da queste considerazioni, Titanium nasce per fornire agli sviluppatori un framework open source per costruire applicazioni desktop o mobile native usando tecnologie web come HTML, CSS, JavaScript, Ruby e PHP. La piattaforma Titanium è stata introdotta da Appcelerator Inc. [32] nel Dicembre 2008, ma il supporto per lo sviluppo di applicazioni mobile è stato

aggiunto nel Giugno 2009, mentre il supporto all'iPad soltanto in Aprile 2010. La versione attuale è la 1.7.2 del 21 Luglio 2011. Per ottenere la piattaforma Titanium Developer è necessario registrarsi presso Appcelerator, sempre allo stesso indirizzo è possibile trovare tutte le risorse o le informazioni di cui si ha bisogno, utilizzando la documentazione Titanium, il forum o il blog ufficiale.

5.3.1 Architettura di Titanium Mobile

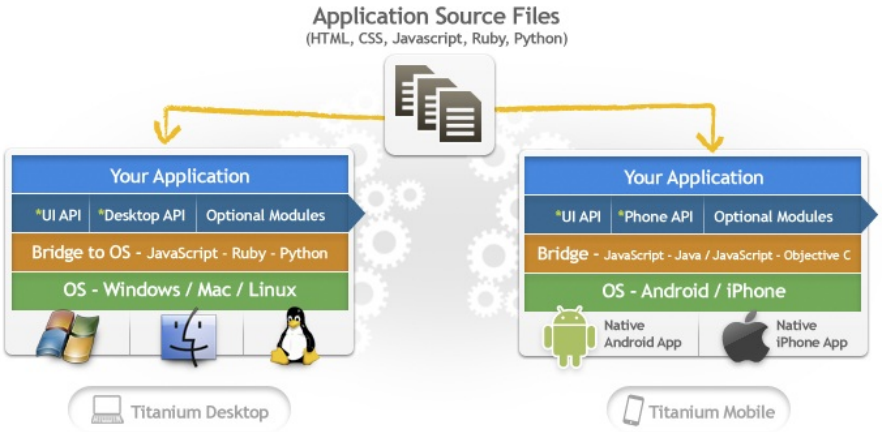

Figura 5.11: Architettura Titanium Platform [35]

La piattaforma Titanium è composta da **Titanium Desktop** e da **Titanium Mobile** [35]:

- **Titanium Desktop** permette di creare applicazioni Desktop native a partire da Ruby, Python, JavaScript, CSS e HTML e il codice generato è portabile e quindi compatibile con Linux, Windows o Mac OS X. Rispetto alle applicazioni create con Adobe AIR, le applicazioni create con Titanium Desktop hanno migliori performance, occupano meno del 10% della memoria delle applicazioni AIR e garantiscono il pieno accesso a tutte le risorse del sistema operativo[10]. Le API Desktop per-

[10]Descrizione Titanium Desktop e confronto con Adobe Air: http://www.appcelerator.com/products/titanium-desktop-application-development

mettono inoltre di generare una UI simile a quella del sistema operativo scelto, sia esso Mac OS X o Windows XP, o Vista.

- **Titanium Mobile** permette di creare applicazioni native per iOS o Android a partire da JavaScript, HTML e CSS, il codice creato viene tradotto dal compilatore in Objective-C (per iOS) o in Java (per Android), Titanium supporta completamente le interfacce di Android e iOS e fornisce pieno supporto ai servizi come Twitter, Facebook, Yahoo, nonchè funzionalità del dispositivo mobile come l'accelerometro, il rilevamento della posizione (geolocation) e le mappe.

Ci sono tanti modi di creare applicazioni, alcune di esse, specialmente per dispositivi mobili, sono in realtà pagine HTML e JavaScript che sono poi fruite sul dispositivo tramite l'avvio di un browser. Con Titanium Developer invece le applicazioni create sono native e questo porta numerosi vantaggi, ad esempio l'estensibilità del codice, l'utilizzo di interfacce native, la velocità di caricamento di una applicazione nativa (2 secondi) rispetto al caricamento di un'applicazione fruibile tramite browser (20 secondi) e analogamente nella transizione fra finestre differenti. Secondo Appcelerator in questo modo i costi di sviluppo vengono ridotti dell'80%[11].

La piattaforma Titanium mette inoltre a disposizione degli sviluppatori Titanium Analytics, una serie di servizi Cloud per testare, distribuire e analizzare le applicazioni create. Inoltre Titanium è un framework open source e quindi si può contribuire a migliorarlo.

All'avvio di Titanium Developer viene chiesto di inserire le credenziali, poi è possibile scegliere se importare un progetto già esistente oppure crearne uno di nuovo (Figura 5.12), scegliendo fra un progetto Desktop, Mobile o iPad. Per ogni progetto bisogna inserire alcuni dati quali nome dell'applicazione e Id, sito internet dello sviluppatore e scegliere la versione di Titanium SDK fra quelle scaricate: gli sviluppatori di Appcelerator lavorano per risolvere i bug

[11]Descrizione Titatium Mobile: http://www.appcelerator.com/products/ titanium-mobile-application-development/

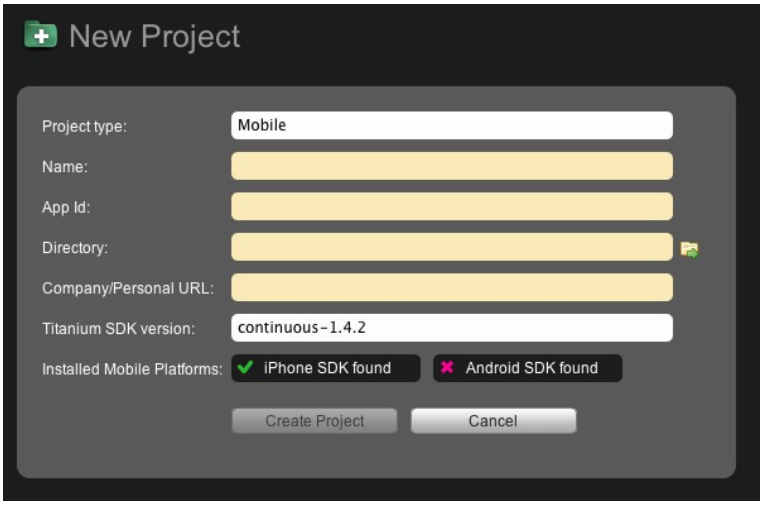

Figura 5.12: Creazione nuovo progetto con Titanium Mobile [35]

segnalati[12], e per rilasciare nuove funzionalità, aggiornando continuamente l'SDK, le versioni beta si possono trovare e importare dal sito di Appcelerator Network [38], mentre i rilasci ufficiali vengono comunicati attraverso il blog.

L'applicazione presenta un menù con tre opzioni:

- Dashboard: è una sorta di schermata di benvenuto, contenente tutti i link a blog, documentazione, forum.
- Edit: permette di modificare i dati inseriti precedentemente.
- Test e Package: permette di compilare il codice e creare un pacchetto pronto per la commercializzazione o il test sul dispositivo.

5.3.2 Panoramica di Titanium Mobile

Come già detto precedentemente Titanium Developer compila il codice scritto in Javascript, HTML e CSS generando l'applicazione in Objective-C o in Java in base alle nostre preferenze. Prima di compilare è necessario aver

[12]Pagina ufficiale Appcelerator di ticketing per la piattaforma Titanium: https://appcelerator.lighthouseapp.com/dashboard

precedentemente installato iOS SDK e/o Android SDK, che vengono automaticamente localizzati da Titanium (in caso contrario compare in messaggio di errore).

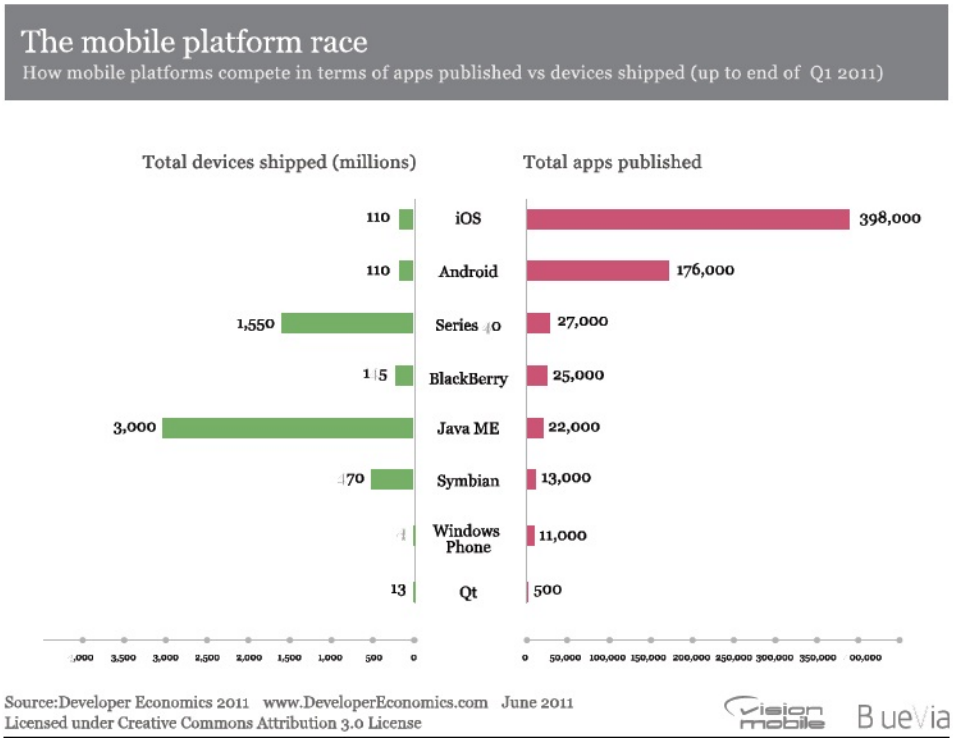

Figura 5.13: La competizione delle piattaforme mobili [36]

La bontà della scelta di Titanium di puntare su applicazioni per iOS e Android (e in futuro anche per BlackBerry) è confermata dai dati presentati alla World Mobile Developer Economics 2011 [36], in Figura 5.13 e in Figura 5.14, dove si nota che le applicazioni per Android e iOS sono quelle più numerose presenti nei vari Store e che gli sviluppatori si rivolgono prevalentemente a queste due piattaforme per le loro applicazioni.

Le API Titanium Mobile sono formate da moduli e oggetti che corrispondono ai framework dei vari layer di iOS, ad esempio il framework UiKit è rappresentato nelle API Mobile dal modulo Titanium.UI, il layer media ha il corrispondente nel modulo Titanium.Media, cioè tutto quello che è ottenibile in iOS con Objective-C lo è anche scrivendo il codice in Javascript e CSS,

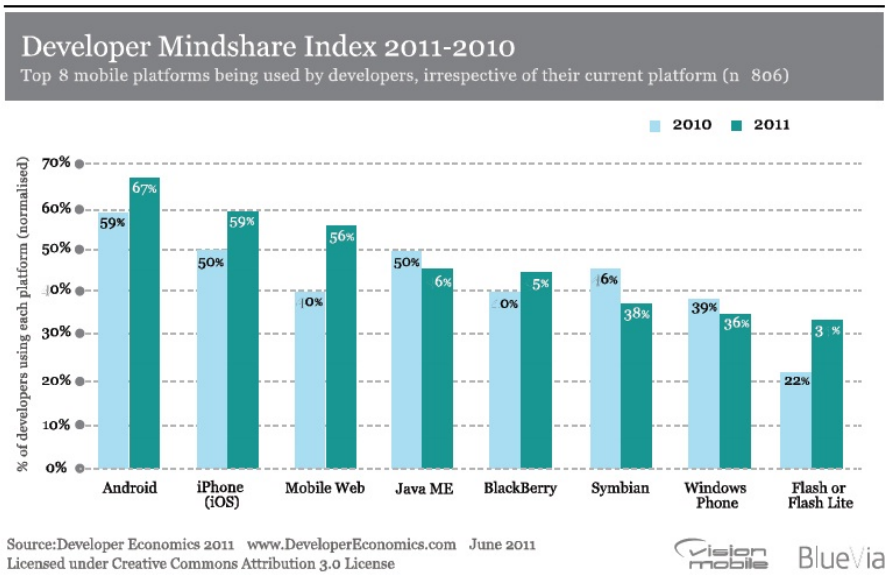

Figura 5.14: Piattaforme usate dagli sviluppatori [36]

sarà poi Titanium Developer a creare il codice nativo a partire da quello scritto da noi. Nella versione 1.7.2 [37] le API Titanium Mobile contengono 35 moduli, 95 oggetti, 915 metodi e 2880 proprietà. Ogni oggetto ha delle proprietà particolari e dei metodi che sono invocabili su di esso, il modulo in testa alla piramide e dai cui discendono gli altri è Titanium Module. L'obiettivo di Titanium è quello di permettere agli sviluppatori di creare un'applicazione riusabile e cross-platform, alcuni moduli però sono specifici per iOS (Titanium.UI.iOS, Titanium.UI.iPad, Titanium.UI.iPhone), altri per Android (Titanium.UI.Android e Titanium.UI.Android.OptionMenu), ma questi riguardano principalmente la UI, i widget disponibili e alcune caratteristiche particolari dei vari sistemi operativi, ma la maggior parte dei moduli permettono di ottenere le stesse funzionalità nei diversi sistemi operativi.

5.3.3 Creare un'applicazione con Titanium Mobile

In questa sezione verrà mostrato come creare un'applicazione con Titanium Mobile sia sul simulatore iPhone e sia su quello Android.

L'esempio **"HelloWorld"** permette di prendere confidenza con la piattaforma e le API e prevede di creare un'applicazione basata su un menù con due opzioni: in base alla scelta fra le due si apre una finestra con una scritta differente. La cartella **"HelloWorld"**, e qualsiasi progetto nuovo, ha il seguente contenuto:

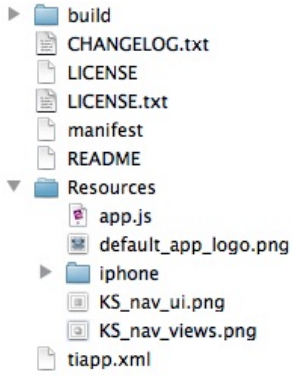

Figura 5.15: Contenuto cartella "Hello World" con Titanium Mobile

- La cartella `build` contiene il codice prodotto dal compilatore.
- `Changelog.txt`: in questo file è possibile inserire un diario delle modifiche apportate al codice.
- `LICENSE`: il file contiene la licenza Appcelerator e non viene incluso nell'applicazione.
- `README`: descrive il progetto, non incluso nell'applicazione.
- `License.txt`: descrive la forma di protezione dei dati scelta dallo sviluppatore.
- `manifest`: contiene le informazioni inserite al momento di creazione del nuovo progetto, è usato dal compilatore.
- La cartella `Resources` è la cartella principale, essa contiene i file javascript creati, nonchè le immagini, video o audio utilizzati. Il file principale che equivale al main di Objective-C è il file app.js.

- `default-app-logo.png`: è un'immagine in formato PNG di dimensioni 55x55 px che rappresenta l'icona dell'applicazione.

- la cartella `iphone` contiene invece l'immagine `Default.png` (320x480 px) che rappresenta l'immagine che compare al caricamento dell'applicazione, detta "splash screen". Quando viene creato un nuovo progetto Titanium crea automaticamente questa struttura e inserisce i propri loghi, che possono essere rimpiazzati con altre immagini, che devono comunque essere delle dimensioni prefissate. Analogamente verrà creata anche la cartella "Android" per contenere le risorse specifiche per questo sistema operativo, se Titanium rileva che Android SDK è correttamente installato sul computer.

- `tiapp.xml`: questo file è un descrittore dell'applicazione, contiene dettagli relativi all'applicazione ed è fondamentale per fissare alcune proprietà specifiche dell'applicazione. È usato sia dal compilatore che a Runtime. Come struttura del file, `tiapp.xml` è il corrispondente del file `info.plist` (information property list), generato in Xcode, che è un un file di testo strutturato contenente informazioni essenziali dell'applicazione in esecuzione. Il testo è strutturato come XML, la cui radice è un dictionary formato da chiavi e valori sulla applicazione e la configurazione.

Il file più importante è il file `app.js`, da esso Titanium genera il codice nativo dell'applicazione. `App.js` è un file javascript, all'interno di esso vengono creati gli oggetti dell'applicazione e definite le funzioni utilizzate. In questo file, e in generale in qualsiasi altro file javascript, è possibile creare un riferimento (proprietà "URL") o includere altri file Javascript utilizzando Titanium.include, del tutto analogo alla direttiva `#include` del linguaggio C. Dividendo il codice in più file è possibile ottenere una modularizzazione del codice, creando per esempio un file per ogni funzionalità dell'applicazione. Questo aumenta la chiarezza e favorisce il riutilizzo del codice, rispetto ad avere un unico file contenente tutte le definizioni e le funzioni. Quando si

crea riferimento ad un altro file Javascript in Titanium ci si riferisce ad un nuovo "sotto-contesto", eseguito in un thread specifico per esso rispetto al contesto generale del file `app.js`, e con un proprio namespace delle variabili.

Il codice dell'applicazione **"HelloWorld"** è il seguente:

```
// create tab group
var tabGroup = Titanium.UI.createTabGroup();

//
// create base UI tab and root window
//
var win1 = Titanium.UI.createWindow({
    title: 'Tab 1',
    backgroundcolor:'#fff'
});
var tab1 = Titanium.UI.createTab({
    icon: 'KS_nav_views.png',
    title:'Tab 1',
    window:win1
});
var label1 = Titanium.UI.createLabel({
    color: '#999',
    text:'Hello World',
    font:{fontSize:20,fontFamily:'Helvetica Neue'},
    textAlign:'center',
    width:'auto'
});

win1.add(label1);

tabGroup.addTab(tab1);
```

```
//open tab group
tabGroup.open();
```

In questo esempio con i vari metodi `Titanium.UI.create` vengono creati i vari oggetti, che poi vengono aggiunti al contenitore principale, in questo caso `TabGroup`. Per creare i vari widget si fa riferimento al modulo `Titanium.UI`, ogni oggetto ha delle proprietà che ricordano le comuni proprietà CSS come color, title, text, width, font, chiaramente adattate per Titanium. Il posizionamento si ottiene con top, bottom, left e right. Inoltre è possibile utilizzare le funzioni JavaScript `SetTimeout()` e `SetInterval()` per ritardare o eseguire più volte nel tempo un determinato comportamento. Scritto il codice si può avviare la compilazione dell'applicazione scegliendo se il target è l'iPhone o Android: Titanium rileverà in automatico la versione corrente di iOS SDK e Android SDK installati sul computer, e se non ci sono errori l'applicazione verrà direttamente lanciata sul corrispondente simulatore, in caso contrario verrà data comunicazione sulla console dei vari errori o warning trovati nel codice. Dalla stessa finestra è inoltre possibile compilare l'applicazione e installarla direttamente sul dispositivo collegato, un'operazione identica a quella eseguita in Xcode con il comando `Build and Run`. A compilazione ultimata, nella cartella del progetto viene creato il Package dell'applicazione iOS così come verrebbe creato da Xcode.

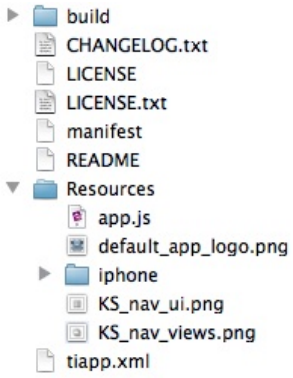

Figura 5.16: Cartella "Hello World" con Titanium Mobile

Nella Figura 5.17 viene mostrata l'applicazione "HelloWorld" sul simulatore iPhone, sulla sinistra, e sul simulatore Android, ottenuta a partire dal medesimo codice.

Figura 5.17: "HelloWorld" sul simulatore iPhone

Figura 5.18: "HelloWorld" sul simulatore Android

5.4 Sencha Touch

Il gruppo Sencha ha rilasciato nella seconda metà del 2010 Sencha Touch [32], un progetto Open Source che permette di sviluppare interfacce web per tablet che assomigliano a delle vere e proprie applicazioni native. Sencha Touch utilizza gli standard HTML5, CSS3 e Javascript, di conseguenza non sarà necessario installare alcuna applicazione aggiuntiva sui dispositivi che supportano questi standard. Con HTML5, le applicazioni Sencha Touch possono essere utilizzate anche offline, e grazie alla geolocazione è possibile integrare dati geografici nei propri progetti. L'uso di CSS3 permette di non avere praticamente nessuna immagine nelle librerie dei componenti: gli stili, i bordi, i gradienti, le ombre, le transizioni, i menu e i pulsanti sono tutti componenti creati con CSS puro. Tramite Javascript vengono gestite una buona parte delle interazioni con l'utente finale. Inoltre il framework permette l'utilizzo di eventi multitouch complessi, interazioni viste finora solamente in applicazioni native. Si tratta di una soluzione concepita per essere multipiattaforma e quindi contemporaneamente compatibile con iOS e con Android, senza escludere la possibilità che in futuro possano essere supportati altri sistemi operativi. Di conseguenza la visualizzazione di tutti questi componenti è indipendente dalla risoluzione dello schermo e la fluidità dell'applicazione dipende dalle caratteristiche tecniche del dispositivo. Ci sono tuttavia alcune differenze minori tra i vari sistemi operativi, ma nulla di rilevante: ad esempio, le transizioni 3D con CSS non sono supportate da Android.

5.4.1 Caratteristiche di Sencha Touch

Alcune caratteristiche di Sencha Touch sono:

- *Flessibilità*

  La progettazione di un tema per un'applicazione Sencha Touch è particolarmente semplice e non richiede particolari abilità da designer. Utilizzando la sintassi CSS, gli sviluppatori possono modificare facilmente

le variabili del tema e alterare completamente il modo in cui si presentano le loro pubblicazioni. Sencha Touch utilizza anche la tecnologia SASS (Syntactically Awesome StyleSheets), che è un'estensione di CSS3 che aggiunge regole annidate, variabili e nuovi tipi di ereditarietà. Grazie a CSS3, quasi ogni aspetto del design, come ad esempio gli angoli arrotondati, le sfumature e le ombre; può essere aggiunto, modificato o rimosso con una riga di codice. Sostituire dei testi con delle icone è senza dubbio un accorgimento utile per un'applicazione che deve girare su dispositivi con schermi dalle dimensioni ridotte. Proprio per questo motivo Sencha Touch include un set di oltre 300 icone di alta risoluzione pronte per essere utilizzate all'interno delle barre degli strumenti.

- *Utilizzo delle applicazioni indipendenti dalla risoluzione*

  Uno dei problemi nello sviluppo di applicazioni multipiattaforma per i dispositivi multimediali è proprio la difficoltà nell'adattarle alle risoluzioni diverse dei vari dispositivi. Sencha Touch utilizza un metodo innovativo per eliminare questo problema, consentendo agli sviluppatori di modificare la scala globale delle loro interfacce e lasciando al framework l'incarico di adattare i vari componenti utilizzati alle caratteristiche tecniche dei diversi display. In questo modo ad esempio, i pulsanti sono sempre abbastanza grandi da toccare, indipendentemente dallo schermo nel quale sono visualizzati.

- *Animazioni*

  Sencha Touch è dotato di un robusto sistema di animazioni, che ne permette l'utilizzo in modo facile e flessibile tra le varie schermate, tramite semplici righe di codice. Le animazioni disponibili, come ad esempio le transizioni in sfumatura o tridimensionali (queste ultime solo per iOS), sono accompagnate da un vasto set di opzioni di personalizzazione, come la direzione e la velocità.

Per creare un'applicazione è stato necessario scaricare il framework Sencha Touch, scaricabile gratuitamente dal sito ufficiale. Dopo aver scaricato l'archivio ed averne estratto il contenuto in una locazione accessibile dai dispositivi mobili (ad esempio un server), sarà sufficiente creare un file HTML che conterrà i riferimenti al framework stesso e alla nostra applicazione per poter cominciare lo sviluppo della web application. Il codice per impostare i riferimenti avrà una struttura molto simile alla seguente:

```
<html>
<head>
<title>Titolo della pagina<\title>
<link rel= "stylesheet"
href="sencha-touch-1.1.0/resources/css/sencha-touch.css"
type="text/css">
<script src="sencha-touch-1.1.0/sencha-touch.js"
type="text/javascript">
</script>
<script src="ApplicazioneWeb.js"
type="text/javascript"></script>
<\head>
<body><\body>
<\html>
```

Dal codice si può notare che i riferimenti impostati sono tre:

`sencha-touch.css:` è il CSS contenente tutti gli stili necessari alla rappresentazione e all'impaginazione dei vari componenti Sencha Touch, come ad esempio pannelli, pulsanti, form e liste. Questo foglio di stile contiene inoltre colori e font utilizzati dal framework e può essere anche modificato per personalizzare graficamente la propria applicazione.

`sencha-touch.js:` questo file contiene il cuore di Sencha Touch. Senza la sua inclusione, non sarebbe possibile avviare l'applicazione web. Va comunque sottolineato che questo script contiene una versione minimale del

framework, la quale non contiene la documentazione e non agevola il debug in caso di errori. In fase di sviluppo e quindi consigliato impostare il riferimento a `sencha-touch-debug.js`, mentre in fase di pubblicazione il riferimento potrà essere impostato nuovamente a `sencha-touch.js`.

`applicazioneWeb.js:` il nome e ovviamente impostato a titolo d'esempio. Questo riferimento punterà al codice Javascript dell'applicazione web creata.

Creazione dell'applicazione web

Per la creazione dell'applicazione web, il file Javascript dell'applicazione web verrà strutturato come segue:

```
Ext.setup({
onReady: function(){
//
//Qui andrà inserito il codice dell'applicazione
//
}
});
```

La funzione `Ext.setup` si occupa di fornire alla pagina web le configurazioni necessarie. Al suo interno e definita la funzione `onReady`, la quale rappresenta il vero punto di ingresso di ogni applicazione Sencha Touch, proprio come il *"Page_Load"* di una pagina ASP.NET o la funzione *"Main"* di un'applicazione C. È quindi necessario che tutti i componenti del Sencha Touch, inclusi nell'applicazione, siano creati all'interno della funzione `onReady`. Nel caso in cui un componente sia all'interno della funzione `Ext.setup`, ma al di fuori della `onReady`, verrà generato un errore Javascript all'interno della console del browser, dovuto al tentativo di utilizzare un componente Sencha Touch ancor prima che il framework stesso sia stato completamente caricato.

Una volta che si è creato la propria pubblicazione, questa sarà subito utilizzabile accedendo da un browser di un dispositivo portatile alla pagina `index.html`. In alternativa, in base alle capacità dello sviluppatore,

sarà ovviamente possibile incapsulare la pubblicazione all'interno di un applicazione Android e/o Apple che potrà essere inserita rispettivamente nel Market e/o nell'AppStore.

Sencha Touch, essendo un progetto Open Source GPLv3, offre un framework ricco e in continuo aggiornamento, con API completamente documentate [60]. Le funzionalità disponibili sono davvero numerose e combinandole tra loro è possibile ottenere risultati sorprendenti per un'applicazione web. Nella prossima sottosezione verranno illustrati alcuni esempi di programmazione introduttivi al fine di sottolineare la semplicità di utilizzo del framework.

Sencha Touch offre diverse opportunità nell'ambito della produzione delle applicazioni web per i dispositivi touchscreen. Visto che il framework è Open Source, è decisamente un vantaggio per lo sviluppatore che, oltre a utilizzare gratuitamente lo strumento, avrà anche l'opportunità di condividere idee e di partecipare al miglioramento del prodotto. Dal punto di vista tecnico, tutti i componenti di Sencha Touch rispondono perfettamente e rapidamente agli input su tablet. Nel caso di applicazioni particolarmente complesse l'hardware del dispositivo fa da collo di bottiglia sulle prestazioni del programma. Questo rende Sencha Touch una valida soluzione per la realizzazione di diversi progetti. La varietà del framework pone ben pochi limiti nello sviluppo, nel caso della realizzazione di un nuovo progetto utilizzando Sencha Touch, il lavoro può risultare molto impegnativo: la scrittura di codice non permette di visualizzare in tempo reale che cosa si sta modificando, di conseguenza un'operazione comune come lo spostamento dei vari elementi della pagina risulta più complessa del classico drag-and-drop di Adobe InDesign.

Va inoltre sottolineato che sono comunque necessarie capacità basilari di programmazione. Lo sviluppatore, difatti, non ha a disposizione interfacce grafiche per impaginare la pubblicazione, ma dovrà esclusivamente scrivere righe di codice. Inoltre dopo aver creato un'applicazione web, a differenza di altri strumenti software, sarà responsabilità dello sviluppatore individuare un modo per distribuirlo e renderlo utilizzabile offline. Una delle idee più

indicate potrebbe essere quella di accorparlo in un'applicazione nativa, per poi poterla distribuire e vendere le proprie applicazioni nei mercati online.

5.4.2 Utilizzo del framework Sencha Touch

In Sencha Touch, la definizione degli oggetti avviene tramite il formato JSON (JavaScript Object Notation), notazione particolarmente semplice da utilizzare. La definizione di un oggetto avviene mediante un insieme di coppie del tipo proprietà: valore separate da virgole, il tutto contenuto all'interno di parentesi graffe. Sul sito ufficiale del framework sono disponibili alcune web application dimostrative che permettono di comprendere appieno tutte le potenzialita di Sencha Touch. In particolare **Sencha Touch Kitchen Sink Demo** e una vera e propria raccolta di tutti i componenti interattivi che possono essere inseriti nelle proprie pubblicazioni. Il codice per la creazione dei componenti puo essere riassunto nelle seguenti fasi:

1. la creazione dell'oggetto,
2. la valorizzazione dei relativi parametri,
3. l'eventuale inserimento di altri oggetti al suo interno.

Eventuali accorgimenti aggiuntivi per alcuni elementi sono riportati nelle sezioni corrispondenti. Nell'applicazione dimostrativa sono presenti diverse categorie, che sono state analizzate con l'utilizzo di un iPad e di uno smartphone Android di fascia medio-bassa.

- **User Interface**

  All'interno della categoria **User Interface** sono esemplificati gli elementi tipici di un'interfaccia utente.

  - *Buttons:* vengono visualizzati 9 pulsanti dalle diverse caratteristiche, disegnati esclusivamente dal codice CSS3 del framework. Ognuno e stato definito come item valorizzando semplicemente le proprieta ui (tipo di oggetto) e text (testo contenuto).

- *Forms:* è presente un classico modulo di inserimento dati dal design particolarmente piacevole, suddiviso in tre tab. Sono presenti "campi testo", "combo box", "sliders" e "checkflag".
- *List:* l'esempio propone un elenco di persone, simile ad una rubrica, che possono anche essere suddivise per gruppo in base all'iniziale.
- *Nested List:* si tratta di liste annidate in cui si giunge all'elemento desiderato passando per step intermedi che affinano la selezione. L'esempio permette di selezionare una casa automobilistica passando prima per la selezione di un continente e di uno stato.
- *Icons:* in questa sezione vengono mostrate tutte le icone del framework che possono essere applicate su tab bar e barre degli strumenti. Le icone dell'esempio sono tutte allineate orizzontalmente e si estendono al di fuori della schermata. Per visualizzarle tutte sarà sufficiente trascinare il dito su una delle due barre in modo da spostarla da destra verso sinistra o viceversa.
- *Toolbar:* le applicazioni web sviluppate con Sencha Touch, proprio come veri software per PC, supportano anche la gestione di barre degli strumenti con l'aggiunta di pulsanti e notifiche.
- *Carousel:* un carousel è un pannello esteso al di fuori dell'area osservabile dall'utente. Muovendo rapidamente il dito in una direzione (swipe) è possibile spostare il carousel di uno step, visualizzando un'altra sezione precedentemente non visibile. La combinazione di due o più di questi oggetti all'interno di una pagina offre un design particolarmente sofisticato, ma allo stesso tempo molto semplice da realizzare.
- *Tabs e Bottom Tabs:* grazie a Sencha Touch è anche possibile utilizzare dei tab per distinguere le varie schermate della propria applicazione. Un aspetto interessante di questi oggetti è

che possono essere ordinati dall'utente semplicemente toccandoli e trascinandoli nella posizione prescelta.

- *Maps:* in Sencha Touch è anche possibile utilizzare le mappe di Google che, in combinazione con la geolocalizzazione offerta dall'HTML5, permettono ad esempio di creare applicazioni il cui comportamento varia in base alla località in cui ci si trova.
- *Overlays:* gli overlays sono tutti quei componenti che si sovrappongono alla schermata che si sta visualizzando. Il framework permette di inserire:
  * Action Sheet: sezioni a comparsa dal basso che comprendono più pulsanti.
  * Overlay: pannelli a comparsa dai quali si esce toccando un punto all'esterno di essi.
  * Alert: messaggi che richiedono la pressione del tasto ok per essere chiusi.
  * Prompt: popup in cui va riempito un campo.
  * Confirm: classica richiesta per confermare un'azione.

- **Animation**

  La seconda sezione dell'applicazione del "Sencha Touch Kitchen Sink Demo" mostra quali animazioni possono essere inserite tra una pagina e l'altra. Nel dettaglio, le transizioni disponibili sono:

  - *Slide:* la schermata scorre verso una direzione e lascia spazio alla successiva.
  - *Slide (cover):* simile all'animazione precedente, ma la nuova schermata scorre in una direzione e si sovrappone alla schermata precedente.
  - *Slide (reveal)* anche questa simile alle precedenti, ma la schermata scorre in una direzione e svela la schermata successiva posizionata sotto.

  - *Pop:* classico effetto di tipo zoom con la schermata che compare dal centro e si estende.
  - *Fade:* effetto sfumatura.
  - *Flip:* effetto tridimensionale che fa ruotare la schermata.
  - *Cube:* effetto tridimensionale che passa alla schermata successiva come se fosse posizionata su un'altra faccia di un cubo.

- **Touch Events**

  Sencha Touch e in grado di riconoscere numerose tipologie di input da touchscreen. Questa sezione della demo stampa quali tra i seguenti input ha riconosciuto: Touchstart, Touchmove, Touchend, Scrollstart, Scroll, Scrollend, Single tap, Tap, Double tap, Taphold, Swipe, Pinch.

Capitolo 6

Conclusioni e Sviluppi Futuri

I dispositivi mobili, al giorno d'oggi, hanno un ruolo sempre più importante tanto nelle aziende quanto nella nostra vita privata, permettendoci di compiere operazioni e svolgere dei compiti che, fino a qualche anno fa, erano eseguibili solo attraverso un normale PC. Non è raro oggi lavorare direttamente su un dispositivo mobile (ad esempio: smartphone) sfruttando una connessione GPRS o UMTS, riducendo i tempi di lavoro e, di conseguenza, aumentando potenzialmente il volume d'affari.

Con il passare del tempo, l'evoluzione tecnologica che ha accompagnato lo sviluppo dei normali PC, ha coinvolto anche i dispositivi mobili. Evoluzione che li ha trasformati da semplici organizer "da tasca" a veri e propri terminali ricchi di funzionalità, discreta potenza di calcolo ma soprattutto di connettività. Quest'ultima caratteristica li ha resi estremamente versatili soprattutto per applicazione di tipo aziendale e di produttività personale.

Leggere le ultime notizie, consultare la propria casella di posta elettronica, verificare il proprio conto in banca o l'andamento del mercato, condividere dei contenuti o solo il proprio stato d'animo su social network; sono tutte comodità a cui gli utenti di smartphone sono poco disposti a rinunciare. In Italia più che in altri paesi, si è disposti a investire su uno smartphone piuttosto che su un cellulare più economico. Secondo una ricerca di Ipsos MediaCT [72] gli utenti di smartphone in Italia sono 20 milioni, il 52% in

più rispetto all'anno 2010. E sono cresciuti del 224% le ricerche effettuate dal dispositivo mobile.

Prima dell'successo degli smartphone il mercato delle applicazioni era gestito esclusivamente con applicazioni Java. Da semplici dispositivi con sistemi operativi dedicati e con poche risorse, si è passati a piattaforme hardware complesse basate su sistemi operativi pensati per girare in ambito desktop.

I sistemi operativi che attualmente sono presenti sul mercato dei dispositivi mobili sono: iOS, BlackBerry, Android, Symbian e Windows Phone. La presenza di piattaforme diverse comporta un'inevitabile problema di portabilità del software. L'assenza di un modello unico che possa soddisfare i requisiti di adattabilità dell'applicazione e dei servizi forniti tramite essa allo specifico utente che li utilizza e, la portabilità di un'applicazione sulle principali piattaforme mobili obbliga lo sviluppo dell'applicazione in molteplici piattaforme.

Un'applicazione non è portabile su più piattaforme a causa della dipendenza dal sistema operativo proprietario, delle dimensioni dello schermo o del tipo di browser utilizzato. Ciascuno dei principali produttori ha ritenuto opportuno realizzare le proprie applicazioni in maniera nativa, utilizzando il linguaggio di programmazione della loro piattaforma, definendo anche un processo di distribuzione dell'applicazione mediante store. In questo modo, si sono creati degli obblighi sulla conoscenza di specifici linguaggi di programmazione come Objective C per la piattaforma iPhone e l'impossibilità nell'utilizzo di un'applicazione creata ad esempio per iPhone da parte di un utente che possiede invece un HTC. Tutto questo ha un forte impatto soprattutto nei riguardi di un'azienda che ad esempio vuole distribuire su più piattaforme l'applicazione, con l'obiettivo di svilupparla una sola volta senza ricorrere a linguaggi proprietari ma a standard web, contenendo i tempi e i costi di realizzazione. Questo comporta non solo l'aumento dei tempi di realizzazione, ma anche l'aumento dei costi.

Così una possibile soluzione a questo problema è il framework per lo

sviluppo multipiattaforma. Ovviamente è difficile pensare a un'applicazione ludica di questo tipo: come è noto i giochi più movimentati richiedono una grande potenza di calcolo per le operazioni di rendering e quindi non si può pensare di appesantire ulteriormente il loro tempo di esecuzione, ma per le restanti tipologie di applicazioni la cross-platform può essere la soluzione a questa grave problematica.

Quindi, è preferibile specializzarsi nello sviluppo su un framework multipiattaforma, che sulle singole piattaforme (che richiedono ambienti di sviluppo, linguaggi di base e metodologie spesso molto differenti tra loro), superando sia i divari tecnici tra piattaforme, sia soprattutto quelli tra dispositivi differenti su cui la stessa piattaforma gira; solo Apple fa da eccezione, in quanto vende esclusivamente in maniera combinata l'hardware e il software.

Nella tesi si è prestato attenzione ai vari framework per lo sviluppo multipiattaforma come: Rhodes, PhoneGap e Titanium Mobile. Questi framework creano applicazioni per iOS, Android (Titanium Mobile), BlackBerry e Symbian (PhoneGap e Rhodes) e particolarità molto interessante di questi framework è il fatto che, in alcuni di loro, le applicazioni mobili sono create partendo da applicazioni web (HTML, CSS e JS).

L'utilizzo di questi framework fornisce codice già ottimizzato per i diversi dispositivi mobili e per le diverse piattaforme. Altri framework utilizzati e apprezzati per lo sviluppo mobile sono: JoHTML5 Mobile App, Sencha Touch, iPFaces, Open Mobile ecc. [73].

Ad una prima occhiata si può subito intuire la grande "convenienza" dell'utilizzo di questi strumenti davvero validi. Ad esempio la versione mobile del noto framework JavaScript, jQuery, offre supporto per qualsiasi piattaforma, da iOS ad Android passando per Windows Phone e Symbian. Questo framework mette a disposizione layout ottimizzati, widget/UI, temi e un set completo di feature, API e componenti, tutto quello che serve per sviluppare siti e/o applicazioni multipiattaforma.

Negli ultimi tempi si parla molto delle tecnologie del futuro, dei nuovi strumenti per lo sviluppo di siti e applicazioni web, dell'utilizzo di vari frame-

work multipiattaforma e della direzione in cui il web sta andando. Tutti sembrano concordare su un punto: *"il futuro del web è mobile"*. Il mobile ha acquisito e acquisterà un'importanza estremamente rilevante grazie alla grande diffusione e la crescita dei dispositivi mobili.

Appendice A

Creazione di una applicazione con PhoneGap per Symbian

In questa appendice si presenta, come esempio, una applicazione sviluppabile con PhoneGap per Symbian.

Phonegap per la piattaforma Symbian si appoggia a "Nokia's Web Runtime" che rappresenta il motore del Web Browser per Symbian S60 (usato come esempio). Il rendering del contenuto delle pagine è dato dalla tecnologia WebKit. La piattaforma usata è Symbian S60 5th Edition. Questo significa che il cellulare Nokia 5530[1] (Figura A.1) con sistema operativo Symbian può utilizzare un'applicazione realizzata in questo modo.

Figura A.1: Nokia 5530

[1] *Nokia 5530*, http://it.wikipedia.org/wiki/Nokia_5530_XpressMusic

Lo sviluppo per cellulari Nokia con sistema operativo Symbian è piuttosto semplice; è sufficiente utilizzare quanto fornito dai sorgenti PhoneGap. Prima di iniziare lo sviluppo è consigliabile dotarsi di un ambiente dove lo script fornito assieme a Phonegap gestisce la pacchettizzazione. Qualora si lavorasse su Windows è necessario dotarsi di Cygwin (scaricabile dal sito: http://www.cygwin.com/), durante l'installazione dello stesso occorre selezionare il pacchetto (usando il comando `make`) e il pacchetto di compressione zip in modo da poter utilizzare lo script dalla shell Cygwin (Figura A.2).

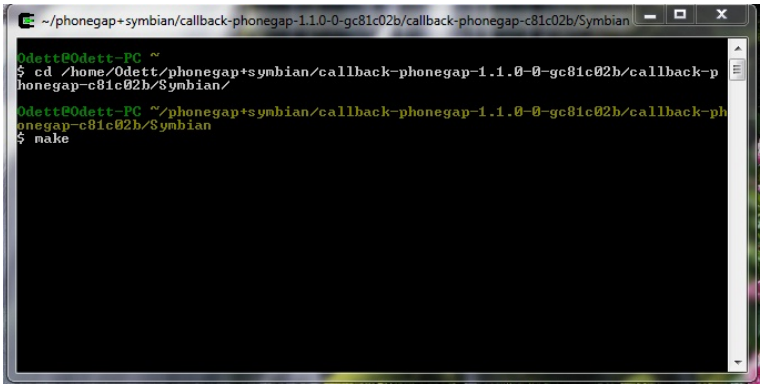

Figura A.2: Terminale Cygwin su Windows 7

Inoltre, è necessario scaricare PhoneGap ed estrarre i contenuti in una cartella apposita e seguire i passi:

- una volta individuata la cartella relativa al sistema Symbian (/phonegap-symbian.wrt) l'applicazione andrà a rimpiazzare il file `index.html` situato nella sottocartella `www`.
- Assicurarsi che il file `phonegap.js` sia linkato nell'intestazione della pagina.
- Rimpiazzare l'icona di default `Icon.png` con un'icona a piacere.
- Eventualmente modificare il file `info.plist`.

- Sviluppare la propria applicazione attorno al fine HTML, le API Phonegap saranno accessibili solo da questa pagina.
- Raggiungere mediante shell Cygwin o Unix la cartella `phonegap-symbian.wrt` e lanciare il comando `make`, lo script produrrà `phonegap-symbian.wrt/pp.wgz`.

Il file `index.html` preso come esempio è un'applicazione che testa la conoscenza del framework PhoneGap.

```
<html>
<head>
<script>
window.onload = function() {
document.getElementById("amount").focus();
document.getElementById("clear").addEventListener('click',
function(event){document.forms[0].reset();}, false);
document.getElementById("split_form").addEventListener
('submit', function(event){
try {
var result = document.getElementById("amount").value *
document.getElementById("gratuity").value /
document.getElementById("num_diners").value;
document.getElementById('result').value=
"$"+result.toFixed(2);
} catch(err)
{
txt="There was an error on this page.\n\n";
txt+="Error description:\n\n" + err.message + "\n\n";
txt+="Click OK to continue.\n\n";
alert(txt);
}
return false;
```

```
}, false);
};
</script>
</head>
<body>
<div id="index">
<h1>Esempio Framework PhoneGap - Piattaforma Symbian</h1>
<form action="#" id="split_form">
<p><label>Nome</label><input type="text" name="amount"
id="amount"></p>
<p><label>Cognome</label><input type="text" name="num_diners"
id="num_diners" value="1"></p>
<p>
<label>Comprensione del Framework PhoneGap</label>
<select id="gratuity" name="gratuity">
<option value="1.0">0%</option>
<option value="1.10">10%</option>
<option value="1.15" selected="1.15">15%</option>
<option value="1.18">18%</option>
<option value="1.20">20%</option>
</select>
</p>
<p><input type="text" name="result" id="result"></p>
<p><input type="submit" value="Calc"></p>
<p><a href="#" id="clear">Clear</a></p>
</form>
</div>
</body>
</html>
```

L'apertura di questo file con un qualsiasi browser viene presentato in questo modo:

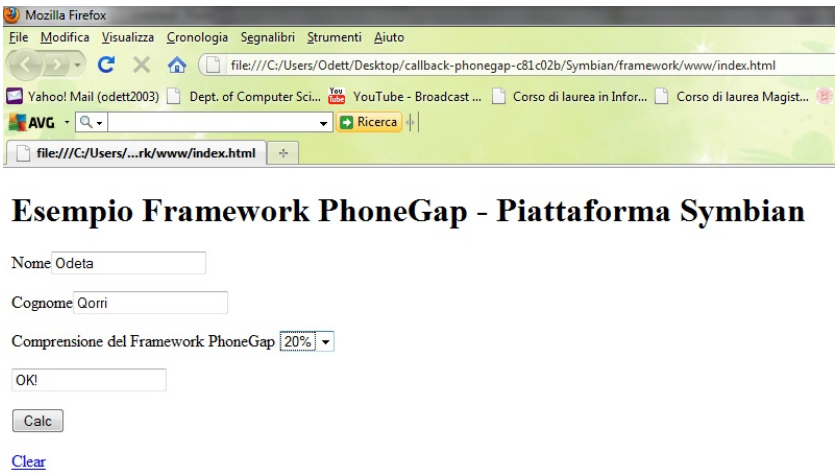

Figura A.3: Apertura del file index.html con Mozilla Thunderbird

Il file `app.wgz`, può essere trasmesso a un cellulare via bluetooth o via cavo usb. Per l'installazione è sufficente aprirla tramite il file manager dal telefono stesso (Nokia 5530). Per una fase di testing prima di arrivare all'installazione è possibile ricorrere a due alternative.

- Una possibilità è quella di installare il simulatore della versione di Symbian che ci interessa testare. Purtroppo per fare ciò è necessario installare Eclipse IDE[2].
- L'installazione di Symbian S60 5th edition richiede attorno a 1GB di spazio su disco. Qualora non si disponga di un cellulare con la stessa versione del sistema operativo è consigliato di fare una prova dell'applicazione sul simulatore.

Caricare l'applicazione sul simulatore

Per caricare il file con l'applicativo sul simulatore è sufficiente copiare il file nella cartella:

```
<cartella installazione SDK>\epoc32\winscw\c\Data\Others
```

[2] *Eclipse IDE* è scaricabile dal sito: http://www.eclipse.org/downloads/

L'emulatore è disponibile seguendo il percorso:

`S60 Developer Tools >[S60 versione piattaforma] >[versione SDK] >Emulatore`

dal menù start di Windows. A seconda della potenza del calcolatore, puó richiedere anche qualche minuto. Una volta avviato entrare nel menù e aprire:

Per il simulatore S60 5th Edition SDK, aprire Applications. Alcuni passi da seguire per installare la nostra applicazione sono i seguenti:

Figura A.4: Selezionare il paese d'origine per Symbian S60 5th edition

Figura A.5: Installazione del file phonegap.wrt

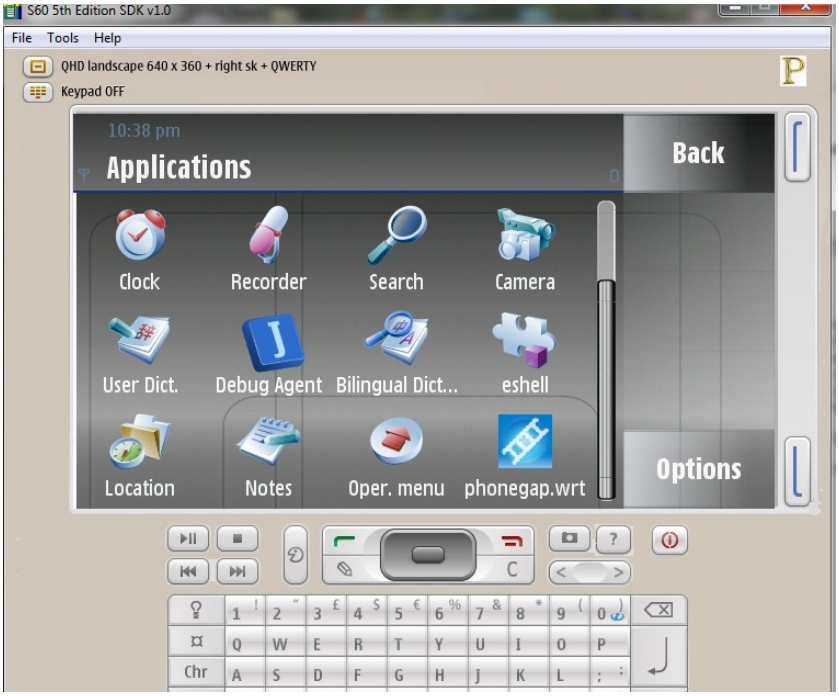

Figura A.6: L'applicazione phonegap.wrt sul Menu delle applicazioni

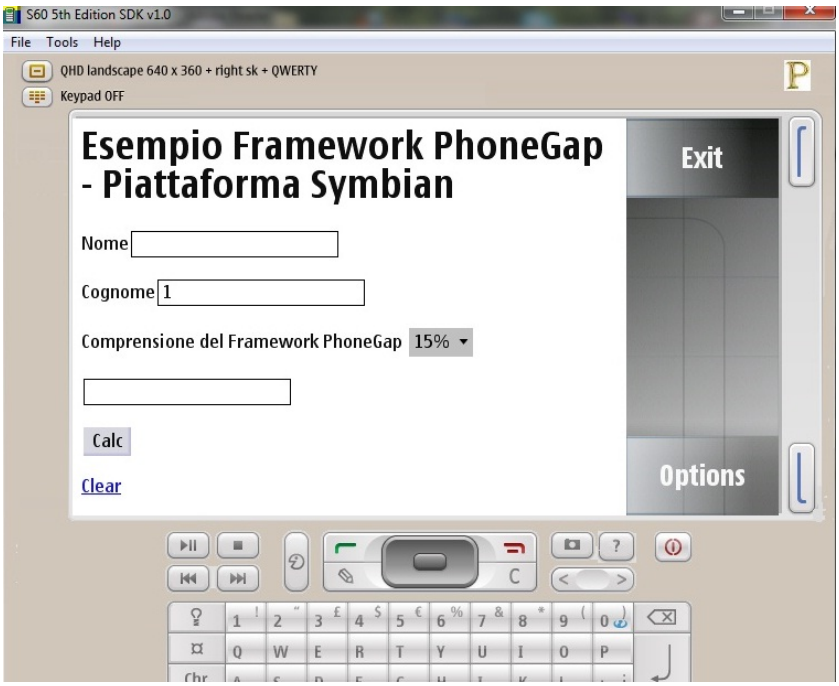

Figura A.7: Apertura dell'applicazione phonegap.wrt

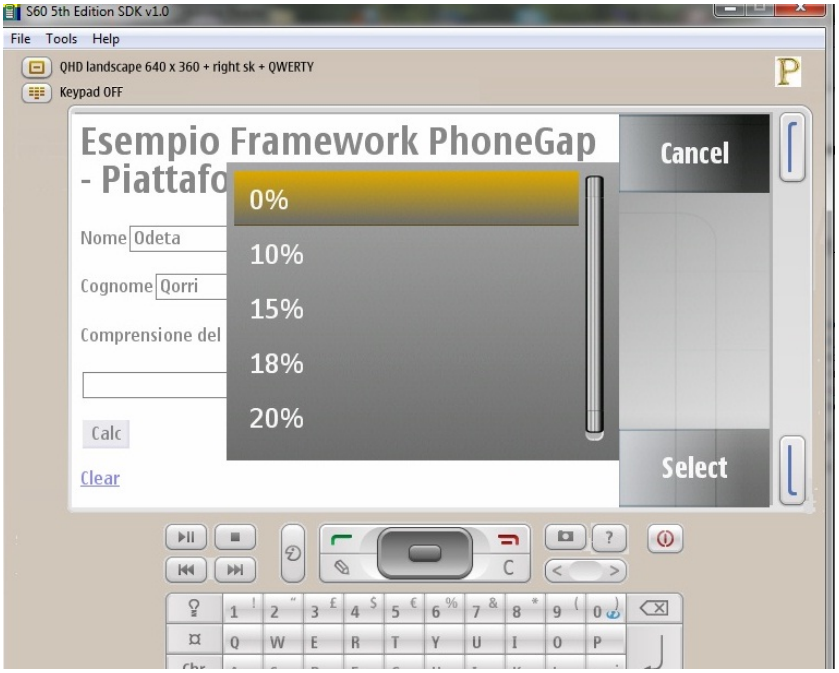

Figura A.8: Modifica dei dati dell'applicazione sul simulatore

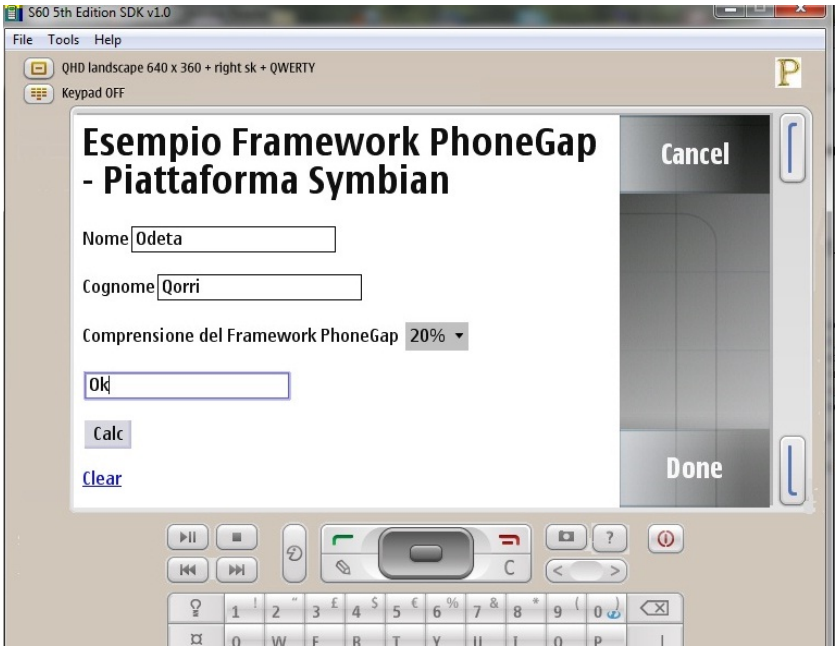

Figura A.9: Modifica dei dati dell'applicazione sul simulatore

Bibliografia

[1] Adelstein F., S.K.S. Gupta, G.G.Richard III, L.Schwiebert, Fundamentals of Mobile and Pervasive Computing, McGraw-Hill Companies. Inc, ISBN 0-07-141237-9, 2005

[2] EMC Corporation, http://www.emc.com/utilities/globalsiteselect.jhtml?checked=true, 01/09/2011

[3] TIA (Telecommunications Industry Association) - Advancing Global Communications, http://www.tiaonline.org/, 01/09/2011

[4] Pei Zheng, Lionel Ni, Smartphone & next generation mobile computing, Elsevier Inc, ISBN 13: 978-0-12-088560-2, 2006

[5] Simon Cellular Phone, http://research.microsoft.com/en-us/um/people/bibuxton/buxtoncollection/detail.aspx?id=40, 03/09/2011

[6] Nokia Communicator, http://en.wikipedia.org/wiki/Nokia_Communicator, 03/09/2011

[7] Elinor Mills, Intervista con Andy Rubin da CNET NEWS.com "From Danger's realm come Android's makers", http://www.zdnetasia.com/insight/communications/0,39044835,62034932,00.htm, 28/11/2007

[8] Google, What is Android,
http://code.google.com/android/what-is-android.html, 03/09/2011

[9] Tannenbaum E.S., Herder J.N., Bos H., "Can we make operating systems reliable and secure", 2006

[10] Darren Murph,"UN Report: 6 in 10 People Worldwide Use Cell Phones", http://www.switched.com/2009/03/03/un-report-6-in-10-people-worldwide-use-cell-phones/,03/03/2009

[11] Findlay Shearer, "Power management in mobile devices",
http://www.eetimes.com/design/power-management-design/4016299/Power-management-in-mobile-devices-A-view-of-energy-Conservation,
04/07/2008

[12] Gartner Inc., "Gartner Says 428 Million Mobile Communication Devices Sold Worldwide in First Quarter 2011, a 19 Percent Increase Year-on-Year", http://www.gartner.com/it/page.jsp?id=1689814,
10/05/19

[13] comScore Inc.,"comScore Reports January 2011 U.S. Mobile Subscriber Market Share",
http://www.comscore.com/Press_Events/Press_Releases/2011/3/ comScore_Reports_January_2011_U.S._Mobile_Subscriber_Market_Share,
07/03/2011

[14] Pen Computing Magazine, "Why did Apple kill the Newton?",
http://pencomputing.com/frames/newton_obituary.html, 27/02/1998

[15] Wikipedia, "Apple Newton",
http://it.wikipedia.org/wiki/Apple_Newton, 20/09/2011

[16] LivingMac Tracker,"L'età della Pietra degli iDevices: Apple Newton",
http://www.biteyourapple.net/index.php/2011/07/leta-della-pietra-

degli-idevices-apple-newton/,
18/07/2011

[17] Harrison R., Shackman M., "Symbian OS C++ for Mobile Phones Volume 3", John Wiley & Sons. Ltd, ISBN 0-470-85611-4, 2007

[18] PalmOne Inc. Treo smartphones.
http://www.palmone.com/us/products/smartphones/, 22/09/2011

[19] Peersman G., Cvetkovic S., Griffiths P.,"The Global System for Mobile Communications Short Message Service", IEEE Personal Communications Magazine, 2000

[20] Vaughan-Nichols S.J., "OSs battle in the smart-phone market", IEEE Computer, 2003

[21] Snol L., "More smartphones than desktop PCs by 2011",
http://www.pcworld.com/article/171380/
more_smartphones_than_desktop%_pcs_by_2011.html, 2009

[22] Falaki H., Mahajan R., Kandula S., "Diversity in Smartphone Usage", ACM, 2010

[23] Scrimieri A., "Apple è l'azienda più valutata al mondo, diventando troppo potente per il Dow Jones",
http://www.ipaditalia.com/apple-e-lazienda-piu-valutata-al-mondo-diventando-troppo-potente-per-il-dow-jones-69343.html,20/09/2011

[24] Hellstrand P., "Optimized iPhone Real-Time Rendering", Master's Thesis in Computing Science at Umea University Sweeden,
11/01/2011,

[25] Speckmann B., "The Android mobile platform", Master's Thesis in Computing Science, University of Michigan, 2008

[26] Square Trade Inc.,"Smart Phone reliability: Apple iPhones with fewest failures, and major Android manufacturers not far behind", http://www.squaretrade.com/pages/cell-phone-comparison-study-nov-10,2010

[27] comScore Inc.,"iPhone and Android Traffic by Connection Type", http://www.comscoredatamine.com/2011/06/iphone-and-android-traffic-by-connection-type/, 2011

[28] Rhodes, http://rhomobile.com/products/rhodes/, 25/09/2011

[29] Rhohub, https://app.rhohub.com/, 25/09/2011

[30] Rhosync, https://app.rhohub.com/, 25/09/2011

[31] Rhomobile bog, www.rhomobile.com/rhodes/iphone-app-store-rules-and-guidelines-on-use-of-frameworks, 29/05/2009

[32] Allen S., Graupera V., Lundrigan L., "Pro Smartphone Cross-Platform Development", Apress, ISBN-13: 978-1-4302-2869-1, 2010

[33] Stark J., "Sviluppare applicazioni per iPhone", O'Reilly & Associates Inc., ISBN: 978-88-481-7506-7, 2010

[34] Phonegap, http://www.phonegap.com/, 26/09/2011

[35] Codeboxed, "Titanium Desktop vs. Titanium Mobile", http://www.codeboxed.com/2011/03/titanium-desktop-vs-titanium-mobile/, 2011

[36] Vision Mobile, "Developer Economics 2011", http://www.visionmobile.com/rsc/researchreports/VisionMobile-Developer_Economics_2011.pdf, 2011

[37] Appcelerator Network, "Titanium Mobile 1.7.2", http://developer.appcelerator.com/apidoc/mobile/, 2011

[38] Appcelerator Network, "Versioni beta Titanium Mobile", http://builds.appcelerator.com.s3.amazonaws.com/index.html, 25/09/2011

[39] Blackberry 7, http://en.wikipedia.org/wiki/BlackBerry_OS, 01/10/2011

[40] BlackberryItalia.it, "La storia dei palmari Blackberry: Un racconto di successo", http://www.blackberryitalia.it/articoli/431/la-storia-dei-palmari-blackberry-un-racconto-di-successo, 09/10/2007

[41] Bochicchio M.S., "BlackBerry 7, nuovo sistema operativo della RIM", http://www.fullpress.info/Sistemi-operativi/BlackBerry-7-nuovo-sistema-operativo-della-RIM/7-42349-1.html, 17/08/2011

[42] WIRED MAGAZINE, "The Untold Story: How the iPhone Blew Up the Wireless Industry", http://www.wired.com/gadgets/wireless/magazine/16-02/ff_iphone?currentPage=2, 01/09/2008

[43] apple.com, "iPhone 4S", http://www.apple.com/it/iphone/, 05/10/2011

[44] OSnews, "Rhodes: Mobile App Development Framework", http://www.osnews.com/story/21578/Rhodes_Mobile_App_Development_Framework, 28/05/2009

[45] freakzion.com, "Mobile Programming FrameWorks", http://www.freakzion.com/2011/04/mobile-programming-frameworks-

rhodes.html,
25/04/2011

[46] rhomobile.com, "Build Rhodes Application",
http://docs.rhomobile.com/rhodes/build, 04/10/2011

[47] rhohub.com, https://app.rhohub.com/, 25/09/2011

[48] rhomobile.com, "Using the Local Database with Rhom",
http://docs.rhomobile.com/rhodes/rhom, 19/01/2011

[49] Kennedy B., Musciano C., "HTML: The Definitive Guide", Second edition, O'Reilly Media, 1997

[50] Meyer E., "Erik Meyer on CSS", First edition, New Riders Press, 2002.

[51] Flanagan D., "Javascript: The Definitive Guide", Second edition, O'Reilly Media, 1997

[52] iWebKit Demo, http://snippetspace.com/iwebkit/demo/, 20/09/2011

[53] Definizione Applicazione web,
http://it.wikipedia.org/wiki/Applicazione_Web, 02/10/2011

[54] Hostscript, http://www.hotscripts.com/, 02/10/2011

[55] Definizione Applicazione nativa,
http://www.mobiquity.it/2010/08/30/73/, 02/10/2011

[56] Stark J.,"Building iPhone Apps with HTML, CSS e JavaScript",
O'REILLY, ISBN 978-0-596-80578-4, 2010

[57] World Wide Web Consortium, http://www.w3.org/, 10/10/2011

[58] HTML5, http://dev.w3.org/html5/spec/, 10/10/2011

[59] Objective C, http://it.wikipedia.org/wiki/Objective_C, 22/10/2011

[60] Documentazione SenchaTouch, http://dev.sencha.com/deploy/touch/docs/, 10/10/2011

[61] jQTouch, http://www.jqtouch.com/, 10/10/2011

[62] Newton MessagePad 2000, http://www.everymac.com/systems/apple/ /messagepad/stats/newton_mp_2000.html, 20/10/2011

[63] Newton EMate 300, http://it.wikipedia.org/wiki/EMate_300, 20/10/2011

[64] Blackberry 8707g, http://smartphone-celluler.blogspot.com/2010/01/blackberry-8707g.html, 20/10/2011

[65] Proiezioni di Gartner, http://www.gartner.com/technology/home.jsp, 20/10/2011

[66] iPhone, http://it.wikipedia.org/wiki/IPhone, 20/10/2011

[67] AddLogic, www.addlogic.se, 20/10/2011

[68] Symbian OS, http://it.wikipedia.org/wiki/Symbian_OS, 20/10/2011

[69] Il linguaggio C, http://it.wikipedia.org/wiki/C_(linguaggio), 22/10/2011

[70] Il linguaggio C++, http://it.wikipedia.org/wiki/C%2B%2B, 22/10/2011

[71] Il linguaggio Java, http://it.wikipedia.org/wiki/Java_(linguaggio), 22/10/2011

[72] Ipsos Media CT, http://www.ipsos.com/mediact/, 22/10/2011

[73] List of Best Open Source Mobile Application Framework, http://www.tobacamp.com/articles/list-of-best-open-source-mobile-application-framework/, 25/10/2010

Ringraziamenti

Desidero ringraziare, il Prof. Antonio Messina per avermi dato l'opportunità di svolgere questa tesi con lui, per il sostegno e l'aiuto datomi nel corso di tutta la scrittura della tesi e per le preziose indicazioni con le quali mi ha guidato durante questo percorso universitario.

Il maggiore ringraziamento va ai miei genitori Perikli Qorri e Zhaneta Qorri per la loro collaborazione, per aver creduto sempre in me, per la fiducia e la pazienza che hanno dimostrato durante questi anni, e a mia sorella Aurora Qorri per avermi consigliato e motivato durante tutto il percorso universitario.

Un grande ringraziamento va anche a Alessio Riccardo per essermi stato vicino ed avermi supportata in tutti i momenti difficili della mia carriera universitaria e per avermi costantemente incoraggiata.

Un caloroso ringraziamento va ai miei amici che mi sono stati vicini durante questi anni: in particolare un grazie speciale a Brikena Llaci, Alfred Dhoga, Adela Kureta, Maria Grazia Contini e Sergio Greco per il sostegno che ancora adesso continuano a darmi.

Desidero inoltre ringraziare le mie carissime colleghe della Segreteria Didattica, e in particolare Serena Alessandrini e Francesca Mazzaglia, per essermi state vicine sia nei momenti difficili, sia nei momenti felici: siete state per me più vere amiche che semplice colleghe.