

ALMA MATER STUDIORUM · UNIVERSITÀ DI BOLOGNA

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

**GUIDA VIRTUALE AL MUSEO DELLA
MUSICA:
UNO STUDIO DI FATTIBILITÀ**

**Relatore:
Chiar.mo Prof.
Fabio Panzieri**

**Presentata da:
Matteo Fagiolino**

**Sessione III
Anno Accademico 2010-2011**

*Dedico questa tesi a mio nonno Roberto.
Una grande persona che ora non c'è più
e che vorrei potesse essere fiera di me ...*

Indice

1	Introduzione	1
1.1	L'istituto museale	1
1.2	I Musei Musicali	3
1.3	Il Museo internazionale e Biblioteca della musica di Bologna	3
1.3.1	Il museo	3
1.3.2	La biblioteca	4
1.4	Il Museo virtuale	5
1.5	Struttura della Tesi	6
2	La Guida Virtuale al Museo della Musica	7
2.1	Analisi dei requisiti software	7
2.1.1	Accessibilità e Usabilità	8
2.1.2	Requisiti della modalità d'uso in remoto	8
2.1.3	Requisiti della modalità d'uso in loco	9
2.2	Analisi delle piattaforme hardware	10
2.3	Considerazioni	11
3	I Dispositivi Mobili	13
3.1	Gli Smartphone	13
3.2	I Tablet PC	14
3.3	I principali sistemi operativi per dispositivi mobili	16
3.3.1	iOS	17
3.3.2	Android	19
3.3.3	BlackBerry OS	19

3.3.4	Windows Phone 7	20
3.3.5	Symbian	21
3.4	Le caratteristiche dei principali sistemi operativi per dispositivi mobili	21
4	Guida Virtuale al Museo della Musica per sistemi iOS	23
4.1	Possibili scelte progettuali	23
4.2	Applicazione Nativa per iOS	24
4.2.1	Linguaggio di sviluppo: Objective-C	25
4.2.2	Ambiente di sviluppo: Xcode	26
4.2.3	Pro e Contro di una applicazione nativa per iOS	27
4.3	WebApp per iOS	28
4.3.1	Il motore WebKit	29
4.3.2	Tecnologie di sviluppo: HTML5, CSS3, JavaScript	30
4.3.3	Pro e Contro di una webapp per iOS	33
5	La WebApp	35
5.1	L'architettura di una applicazione web	35
5.2	L'interfaccia grafica	38
5.2.1	Look and Feel di una applicazione nativa	38
5.3	I contenuti multimediali	42
5.3.1	La galleria immagini	42
5.3.2	Video e audio	43
5.4	L'audioguida	44
5.5	La geolocalizzazione	45
5.6	L'interazione con i servizi nativi di iOS	47
5.7	Memorizzazione dei dati in una webapp	48
5.8	La modalità offline	50
6	La Realtà Aumentata	51
6.1	L'applicazione ibrida	51
6.1.1	Il Framework PhoneGap	52

6.2	La piattaforma Layar	52
6.2.1	Layar Vision	53
	Conclusioni e sviluppi futuri	55
	Bibliografia	57

Elenco delle figure

1.1	Alcune immagini del Museo della Musica di Bologna	4
2.1	La Realtà Aumentata nei musei	10
3.1	Alcuni smartphone disponibili in commercio	15
3.2	Alcuni tablet disponibili in commercio	17
3.3	Elenco di gesture multi-touch	18
4.1	I Sistemi Operativi mobile più usati per internet	24
4.2	La piattaforma di sviluppo Apple, Xcode 4	26
4.3	Alcune applicazione web su Safari Mobile	29
4.4	Il logo di HTML5	32
5.1	Il modello Three Tier per applicazioni web	36
5.2	Alcune modifiche dell'applicazione web per iOS	40
5.3	Esempio di interfaccia con JQTouch	41
5.4	Esempio galleria immagini con PhotoSwipe	43
5.5	Esempio del lettore JPlayer	45
5.6	Esempio servizio di geolocalizzazione	46
5.7	Esempio servizio Contatti	48
6.1	Architettura della piattaforma Layar	53

Elenco delle tabelle

3.1	Principali sistemi operativi per dispositivi mobili	22
-----	---	----

Capitolo 1

Introduzione

Scopo di questa tesi è di valutare le tecnologie diffuse ai giorni nostri che possano consentire la realizzazione di un software per l'esplorazione virtuale dei musei della musica. Lo studio si sofferma inizialmente sulla definizione di istituto museale, per capire precisamente i possibili requisiti e servizi che deve offrire un'applicazione ad essi dedicata. Con un'analisi di questi requisiti vengono poi valutati i dispositivi attualmente disponibili in commercio, tenendo in considerazione le caratteristiche hardware e software, oltre a valutare la diffusione in mercato di tali piattaforme. Buona parte dello studio è dedicato alle scelte progettuali possibili per sviluppare il software, indicando per entrambi gli strumenti e gli ambienti di sviluppo e analizzando vantaggi e svantaggi dei diversi approcci. L'orientamento verso una delle scelte progettuali è poi seguita da un approfondimento sulle tecnologie da adottare per l'implementazione delle funzionalità valutate in sede di analisi dei requisiti, fornendo alcune documentazioni e dimostrazioni di fattibilità.

1.1 L'istituto museale

Prima di iniziare questo percorso relativo allo studio di fattibilità di una Guida Virtuale al Museo della Musica è opportuno descrivere in dettaglio cosa sia un istituto museale, in modo da comprendere e valutare durante

la fase di progettazione quali possano essere gli obiettivi, i vantaggi e gli svantaggi nello sviluppo di un'applicazione legata ad un ente culturale.

Il museo è una raccolta, pubblica o privata, di oggetti relativi ad uno o più settori della cultura, della scienza e della tecnica. Come definito dal Codice dei beni culturali e del paesaggio¹, per quanto riguarda il territorio italiano:

“il museo è una struttura permanente che acquisisce, cataloga, conserva, ordina ed espone beni culturali per finalità di educazione e di studio”

Le tipologie di musei relativi ai settori di interesse culturale, storico e scientifico sono varie. Di seguito è presente un elenco delle principali categorie di musei, le quali possono comprendere al loro interno ulteriori tipologie di interesse più specifiche:

- Museo archeologico

- Museo d'arte

- Museo storico

- Museo scientifico

- Museo di scienze naturali

- Museo virtuale

¹Il Codice dei beni culturali e del paesaggio è il principale riferimento normativo italiano che attribuisce al Ministero per i Beni e le Attività Culturali il compito di tutelare, conservare e valorizzare il patrimonio culturale dell'Italia.

Emanato con decreto legislativo del 22 gennaio 2004, n. 42, e modificato poi dal d. lgs. 24 marzo 2006, n. 156, dal d. lgs. 24 marzo 2006, n. 157, dal d. lgs. 26 marzo 2008, n. 62, e dal d. lgs. 26 marzo 2008, n. 63

1.2 I Musei Musicali

I musei musicali fanno parte della categoria di musei storico-artistici e sono strutture adibite alla raccolta ed esposizione di opere, strumenti e documentazioni relative all'arte della musica. Tra i vari istituti museali di questa tipologia ve ne sono alcuni specializzati nella collezione di strumenti musicali, altri nella raccolta ed archiviazione di materiali riguardanti uno o più artisti, altri ancora nella documentazione del patrimonio musicale di specifici territori. Un esempio di musei musicali possono essere il Musical Museum di Londra con la sua ricca collezione di strumenti musicali meccanici ed automatici, il Beethoven-Haus di Bonn dedicato al compositore Ludwig van Beethoven e il Museo internazionale e Biblioteca della musica di Bologna.

1.3 Il Museo internazionale e Biblioteca della musica di Bologna

Per svolgere lo studio di fattibilità di una Guida Virtuale ai musei della musica abbiamo preso in considerazione principalmente il Museo internazionale di Bologna. In questa sezione viene descritta brevemente la storia e le caratteristiche del museo con un approfondimento più dettagliato dei contenuti e dei servizi che l'istituto offre ai visitatori.

Il Museo internazionale e Biblioteca della musica di Bologna nacque nel 1959 per custodire le collezioni di beni musicali del Comune di Bologna. Inizialmente denominato Civico Museo Bibliografico Musicale, viene cambiato il suo nome nel 2004 in onore dell'inaugurazione della nuova sede museale presso il centro storico di Bologna, situata all'interno di Palazzo Sanguinetti. L'istituto è suddiviso in due parti, il museo e la biblioteca.

1.3.1 Il museo

Il percorso museale di Palazzo Sanguinetti è composto da dieci sale che comprendono un'ampia selezione di volumi, strumenti e dipinti relativi al-

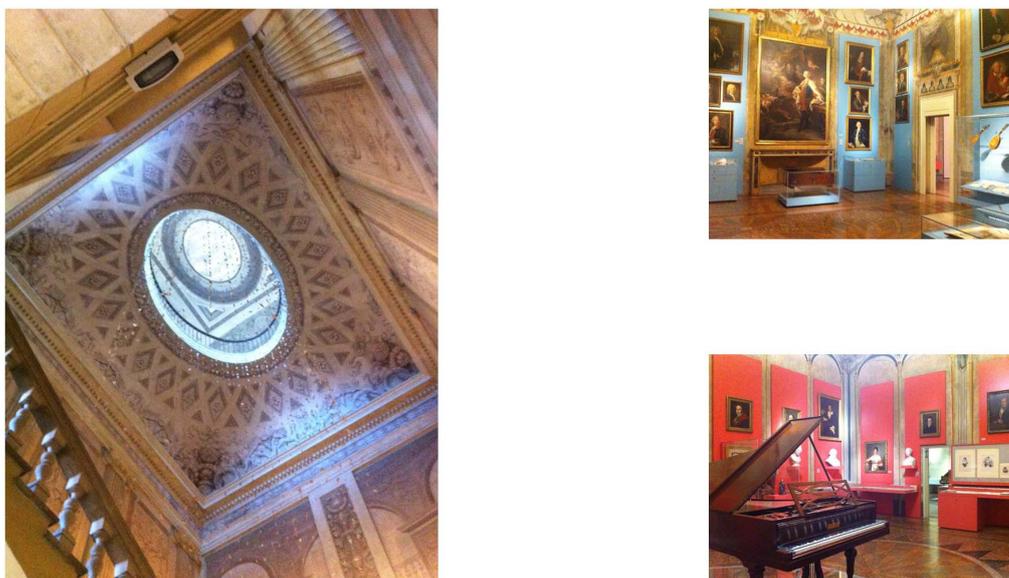


Figura 1.1: Alcune immagini del Museo della Musica di Bologna

la musica del periodo storico che va dal Quattrocento all'Ottocento facenti parte della ricca collezione di padre Giovanni Battista Martini. I servizi del museo offrono la possibilità di effettuare il percorso museale mediante l'illustrazione delle opere da parte di una visita guidata, previa prenotazione, oppure utilizzando un dispositivo a noleggio per l'audioguida. Tale dispositivo amplia notevolmente l'esperienza di esplorazione delle opere esposte mediante tracce audio che comprendono i commenti di buona parte del materiale esposto (identificabile tramite un apposito numero accanto alle esposizioni) arricchito dalla presenza di alcuni brani musicali riguardanti i più importanti compositori presenti all'interno del percorso.

1.3.2 La biblioteca

La biblioteca è situata nell'edificio di Piazza Rossini 2 (ex Convento di San Giacomo), in locali annessi al Conservatorio di musica G.B. Martini. Al suo interno è raccolta la collezione ereditata da Padre Martini, ampliata in seguito dal suo discepolo Stanislao Mattei e dal bibliotecario Gaetano Ga-

spari, il quale dal 1855 diresse la biblioteca riuscendo a ordinare e schedare tutto il materiale librario. La Biblioteca, così come da principio, è attualmente divisa in tre parti: la prima parte contiene opere teoriche e letterarie; la seconda opere e composizioni destinate a servire nella pratica dell'arte; la terza comprende infine la collezione dei libretti d'opera. Le schede manoscritte di Gaspari, ordinate in ordine alfabetico per autore, hanno permesso di creare il Catalogo della Biblioteca del Liceo Musicale di Bologna, denominato in seguito Catalogo Gaspari. I servizi della biblioteca comprendono la consultazione e il prestito del materiale, oltre al reperimento di informazioni bibliografiche e la riproduzione di microfilm, diapositive e fotografie.

1.4 Il Museo virtuale

Il museo virtuale ha diverse definizioni, ma principalmente è un modo per rendere fruibili e consultabili le opere all'interno di un museo in remoto o in loco utilizzando le nuove tecnologie. Le nuove tecnologie, applicate ai musei, o collegate ad essi, possono essere collegamenti video, audioguide o videoguide da utilizzare durante le visite e strumenti per la consultazione in remoto di guide multimediali interattive, visite simulate e archivi online. Per alcuni il museo virtuale è un semplice sito web, per altri è una visita simulata all'istituto mediante grafica in 3D o tramite la visualizzazione di foto panoramiche relative ai percorsi delle esposizioni. Gli esempi più esaustivi delle due tipologie di musei virtuali sopra esposte possono essere il Catalogo Gaspari On Line² per la consultazione via internet del patrimonio contenuto all'interno della Biblioteca della musica di Bologna e il Google Art Project³, l'ambizioso progetto della società di Mountain View che sfrutta la tecnologia utilizzata dal progetto Google Street View per mappare i percorsi espositivi di alcuni dei più famosi musei del mondo offrendo la possibilità di esplorare

²<http://badigit.comune.bologna.it/cmbm/scripts/gaspari/index.asp>

³<http://www.googleartproject.com/>

virtualmente i musei e visualizzare le opere digitalizzate con una qualità fotografica di 7 gigapixel.

1.5 Struttura della Tesi

Di seguito viene riportata in sintesi una descrizione dei capitoli che compongono lo studio di fattibilità per una Guida Virtuale al museo della musica.

Il Capitolo 2 si sofferma ad analizzare un'ipotetica Guida Virtuale esaminando i vari requisiti del software, le funzionalità d'uso ed i requisiti che devono essere soddisfatti dalla piattaforma hardware.

Nel Capitolo 3 viene fatta una panoramica dei dispositivi che possono soddisfare i requisiti hardware introdotti con il Capitolo 2, analizzando le diverse tipologie di piattaforme attualmente presenti in commercio e descrivendo i sistemi operativi più usati dagli utenti.

Il Capitolo 4 affronta un confronto tra le possibili scelte progettuali di una Guida Virtuale al museo della musica per sistema operativo iOS di Apple, descrivendo per entrambe gli strumenti e gli ambienti di sviluppo ed indicando vantaggi e svantaggi che comportano.

Il Capitolo 5 approfondisce una delle scelte progettuali valutate nel capitolo precedente, introducendo una descrizione delle possibili implementazioni che possono essere progettate per soddisfare i requisiti software analizzati nel Capitolo 2.

Il Capitolo 6 infine suggerisce un'ulteriore scelta progettuale necessaria per implementare la funzionalità di Realtà Aumentata, con relativa descrizione degli strumenti e delle piattaforme da utilizzare.

Capitolo 2

La Guida Virtuale al Museo della Musica

Abbiamo pensato a una guida virtuale come una sorta di portale che possa comprendere al suo interno la presenza di più musei della musica correlati tra loro, permettendo agli utenti la ricerca, la consultazione e la visita dei musei anche senza il bisogno di recarsi all'interno di essi, oltre ad essere utilizzabile anche come mezzo per arricchire l'esperienza esplorativa lungo il percorso museale presso gli istituti. Per capire come poter creare un'applicazione che racchiuda al suo interno le caratteristiche elencate sopra, andiamo ora ad analizzare i requisiti principali delle funzionalità software e della piattaforma hardware sui cui sarà in seguito basato lo studio relativo alla progettazione.

2.1 Analisi dei requisiti software

I requisiti fondamentali di un applicazione destinata al pubblico, e quindi incentrata sull'interazione uomo-macchina, sono l'accessibilità e l'usabilità del software. In seguito vedremo anche i diversi requisiti in base al contesto d'uso che dovrà essere in particolare relativo all'utilizzo dell'applicazione in remoto e in loco.

2.1.1 Accessibilità e Usabilità

Accessibilità

È la caratteristica di un dispositivo, di una risorsa o di un servizio di essere fruibile con facilità da qualsiasi tipologia di utenti. Consiste in particolare modo, da un lato, nella possibilità di accesso alle informazioni da parte di persone affette da disabilità temporanea o stabile, dall'altro nel garantire l'accesso ad una stessa risorsa anche da dispositivi con tecnologia datata o diverso sistema operativo.

Usabilità

È il grado di efficacia, efficienza e soddisfazione con le quali determinati utenti raggiungono determinati obiettivi in determinati contesti¹. L'efficacia è l'accuratezza e completezza con cui certi utenti possono raggiungere certi obiettivi in ambienti particolari. L'efficienza indica il costo in termini di risorse spese dall'utente (che possono essere economiche, fisiche e temporali) per riuscire a portare a termine un compito. La soddisfazione indica invece se l'esperienza dell'uso del prodotto sia stata positiva.

2.1.2 Requisiti della modalità d'uso in remoto

L'applicazione deve permettere agli utenti la possibilità di accedere alle informazioni relative ai musei senza che siano fisicamente presenti all'interno degli istituti stessi. Le caratteristiche del software utilizzabile in remoto devono comprendere l'accesso alla ricerca di musei, opere, artisti e documentazioni mediante l'uso di un database relazionale contenente le copie digitali dei reperti presenti all'interno dell'istituto museale. L'utente in questo modo può decidere se accedere alle informazioni di un unico museo o trovare materiale informativo riguardante un artista, un'opera o più contenuti legati a una singola risorsa. Per la consultazione di materiale multimediale il software

¹Definizione di Usabilità dell'International Standard Organization (ISO 9241-11:1998)

deve offrire la possibilità di accedere a gallerie immagini, visualizzare filmati e riprodurre brani o tracce audio. Il database deve anche comprendere i dati relativi al calendario degli eventi in programma presso gli istituti, gli orari di apertura e chiusura al pubblico e le informazioni riguardanti la posizione geografica degli enti museali (con riferimento agli indirizzi civici e ai dati di geolocalizzazione), oltre ai riferimenti per i contatti, come numeri di telefono e indirizzi email.

2.1.3 Requisiti della modalità d'uso in loco

L'applicazione deve, allo stesso tempo, fornire degli strumenti per il supporto durante una vera visita alle opere esposte dentro ai musei. In particolare i visitatori dovranno poter interagire con gli oggetti presenti lungo il percorso museale attraverso materiale multimediale aggiuntivo, come, per esempio, informazioni tramite l'ascolto di commenti audio, di brani musicali e la visualizzazione di documenti video. In definitiva, il software deve andare a sostituire le attuali audioguide presenti in ogni museo con un dispositivo di più comune uso per i visitatori, in modo da soddisfare maggiormente il grado di accessibilità alla piattaforma di uso e al software. La Guida Virtuale deve quindi permettere all'utente finale di usufruire degli strumenti di visita classici dei musei (per esempio con il riconoscimento delle tracce audio relative alle opere mediante il riconoscimento delle targhette esposte accanto ad esse). Tra le nuove tecnologie più in voga da utilizzare per le visite ai musei, vi sono al momento sperimentazioni dell'uso della Realtà Aumentata² per aumentare, appunto, l'interazione tra visitatori ed esposizioni. L'implementazione di questa funzionalità permette di rendere ancora più immersiva l'esplorazione degli scenari circostanti attraverso lo schermo dei dispositivi portatili o fissi, fornendo una rappresentazione virtuale di informazioni sotto

²Per realtà aumentata (in inglese *augmented reality*, abbreviato AR), o realtà mediata dall'elaboratore, si intende l'arricchimento della percezione sensoriale umana mediante informazioni, in genere manipolate e convogliate elettronicamente, che non sarebbero percepibili con i cinque sensi.

forma di immagini, video o oggetti in 3D altamente interattivi. La figura 2.1 mostra alcuni esempi dell'utilizzo della Realtà Aumentata all'interno dei musei per offrire una maggiore interazione tra le opere esposte e i visitatori.



Figura 2.1: La Realtà Aumentata nei musei

2.2 Analisi delle piattaforme hardware

Le piattaforme hardware sulle quali progettare una Guida Virtuale al Museo della Musica devono principalmente avere alcuni requisiti fondamentali. Di seguito viene proposto un elenco delle principali caratteristiche richieste dai dispositivi per la realizzazione delle funzionalità software sopra elencate.

Piattaforma di uso comune: perchè sia possibile la fruizione dell'applicazione a un maggior numero di utenti, è indispensabile orientare la progettazione della Guida Virtuale verso dispositivi di “massa”, ossia, appunto, le piattaforme hardware maggiormente utilizzate secondo le ricerche di mercato;

Connettività ad internet: il dispositivo deve permettere la connessione ad internet per consentire la consultazione e la ricerca di informazioni relative ai musei con richieste ad un database in remoto;

Utilizzo in mobilità: per soddisfare i requisiti della modalità d'uso durante una visita guidata, è necessario che il dispositivo sia portatile e di facile trasporto.

2.3 Considerazioni

I requisiti appena analizzati forniscono già un'indicazione sulla scelta dei dispositivi da poter utilizzare come piattaforme principali per una Guida Virtuale al museo della musica. In particolare le funzionalità dell'applicazione che abbiamo appena esaminato necessitano di un sistema hardware di facile trasporto e con capacità d'uso paragonabili ai normali calcolatori. Il prossimo capitolo approfondisce quindi lo studio di fattibilità analizzando alcune tipologie degli attuali dispositivi mobili attualmente disponibili in commercio.

Capitolo 3

I Dispositivi Mobili

Le piattaforme hardware che soddisfano le principali caratteristiche analizzate in precedenza sono prevalentemente alcune precise tipologie di dispositivi mobili. Un dispositivo mobile, o portatile, è un piccolo apparato di elaborazione di dati solitamente provvisto di un display con funzionalità touch screen o di una piccola tastiera con cui l'utente si può interfacciare per immettere dati in input e ricevere dati in output. I più recenti sistemi hardware mobili permettono di sfruttare le stesse prestazioni di calcolo dei computer convenzionali con la comodità di poterli utilizzare in mobilità. Le categorie che vanno per la maggiore tra i dispositivi portatili sono tutt'ora gli smartphone e i tablet PC.

3.1 Gli Smartphone

Gli smartphone sono dei dispositivi portatili che uniscono in un'unica piattaforma le funzionalità dei classici telefoni cellulari a quelle di gestione di dati personali. Possono infatti derivare dall'evoluzione di un telefono mobile con comuni funzioni di chiamata, invio e ricezione SMS e MMS e trasmissione dati di accesso alla rete internet, a cui si aggiungono le funzionalità dei PDA¹,

¹Il Personal Digital Assistant è un computer di dimensioni contenute, tali da essere portato sul palmo di una mano, dotato di uno schermo tattile.

quali calendario, rubrica, orologio, calcolatrice, blocco note, e-mail e lettore multimediale. L'esordio degli smartphone avvenne nel 1993 con il modello chiamato Simon progettato dalla IBM il quale, offrendo caratteristiche ibride derivate dalle funzionalità di cellulari e PDA, propose al mercato un prodotto che si è evoluto ed espanso sempre di più fino ai giorni nostri. Attualmente gli smartphone hanno infatti conquistato sempre di più il mercato della telefonia mobile, arrivando addirittura a superarlo in termini di vendite. Per quanto riguarda il mercato europeo, nel secondo trimestre del 2011 gli smartphone hanno totalizzato 21,8 milioni di unità vendute contro i 20,4 milioni di telefoni cellulari tradizionali[1]. Attualmente il mercato degli smartphone rappresenta il 30% di quello mondiale. Tale sorpasso è dovuto principalmente alla più ampia versatilità di questi dispositivi resa possibile grazie all'integrazione di capacità hardware e software sempre maggiori e sempre più indispensabili. Le ultime generazioni di smartphone includono accesso a internet tramite browser, pianificazione delle attività, fotocamere, ricevitore GPS e la compatibilità con i più comuni formati di file e documenti. Per quanto riguarda invece le connessioni wireless, gli smartphone utilizzano le tecnologie standard di telefonia mobile, quali GSM, GPRS, EDGE, UMTS, HSDPA, HSUPA ed anche antenne Wi-Fi integrate e moduli Bluetooth. La figura 3.1 mostra alcuni degli smartphone più diffusi attualmente in commercio.

3.2 I Tablet PC

Il tablet PC è un computer portatile che permette agli utenti di interfacciarsi con il sistema mediante una penna o l'utilizzo delle dita. Di fatto il tablet PC è un computer portatile con maggiore capacità di input. Il termine "tablet PC" divenne famoso nel 2000 quando Microsoft presentò una serie di dispositivi che rispondevano a determinate specifiche Microsoft², ma

²Particolari specifiche che permettono all'utente, grazie alle funzionalità integrate in diversi sistemi operativi Windows, l'utilizzo di inchiostro digitale e del riconoscimento della scrittura.



Figura 3.1: Alcuni smartphone disponibili in commercio

attualmente ci si riferisce con questo termine a qualsiasi tablet PC indifferentemente dal sistema operativo utilizzato. Nell'ultimo decennio sono stati prodotti diversi modelli di tablet PC basati principalmente su queste specifiche e sui sistemi operativi Windows, dotati principalmente di un display di grandezza compresa tra i dieci e i dodici pollici. Tra i vari modelli di tablet PC che sono stati introdotti nel mercato, nel 2010 si sono diffusi nuovi dispositivi ridefiniti come tablet computer che hanno abbandonato il precedente paradigma di tablet in favore di un'interfaccia differente, sviluppata esplicitamente per la gestione tramite funzionalità tattile. Il loro nome deriva dalla forma di tali dispositivi che assomiglia a quella di una tavoletta utilizzata per la scrittura. Le nuove interfacce differiscono dal classico sistema adottato dai personal computer e sono molto più simili a quelle utilizzate dagli smartphone, prevedendo l'interazione unicamente tramite il tocco delle dita e quindi implementando un'interfaccia grafica pensata appositamente

per questa nuova tipologia di interazione, avvantaggiandosi in genere della funzionalità multi-touch³ che gli schermi gestiscono. Il primo e più famoso esponente di questa recente categoria è stato l'iPad di Apple presentato da Steve Jobs il 27 gennaio 2010 ad una apposita conferenza dell'azienda tenuta allo Yerba Buena Center for the Arts Theater di San Francisco. iPad ha rinvigorito il mercato dei tablet riuscendo a vendere ben 18 milioni di dispositivi nell'arco del 2010[2]. In seguito fecero la loro comparsa anche molti altri sistemi, i quali hanno abbandonato le architetture x86⁴ e i sistemi operativi Windows in favore di architetture ARM⁵ (architettura con consumi energetici inferiori) e sistemi operativi ottimizzati per questi dispositivi, come iOS, Android e BlackBerry OS. La figura 3.2 mostra alcuni dei tablet PC che sono attualmente più usati dagli utenti.

3.3 I principali sistemi operativi per dispositivi mobili

Sia gli smartphone che i tablet di ultima generazione sono supportati da diversi sistemi operativi pensati appositamente per i dispositivi mobili. Un sistema operativo per dispositivi mobili è un sistema operativo che controlla un dispositivo mobile con lo stesso principio con cui sono controllati i computer desktop o portatili dai sistemi operativi Mac OSX, Linux e Windows. A differenza dei sistemi operativi per personal computer, quelli per dispositivi

³Multi-touch è una tecnologia che rappresenta un'evoluzione dello schermo tattile. Uno schermo tattile multitocco si differenzia dai precedenti per il fatto che è sensibile al tocco in più punti diversi della superficie contemporaneamente.

⁴Architettura di una famiglia di microprocessori, inizialmente sviluppata e prodotta da Intel. Attualmente è l'architettura più diffusa nel mercato dei PC desktop, portatili, e server economici.

⁵L'architettura ARM indica una famiglia di microprocessori RISC a 32-bit sviluppata da ARM Holdings e utilizzata in una moltitudine di sistemi embedded. Grazie alle sue caratteristiche di basso consumo (rapportato alle prestazioni) l'architettura ARM domina il settore dei dispositivi mobili dove il risparmio energetico delle batterie è fondamentale.



Figura 3.2: Alcuni tablet disponibili in commercio

mobili affrontano problematiche legate alla natura della piattaforma portatile, più critiche rispetto ad un comune computer. Per esempio la limitatezza delle risorse hardware come memoria e processore, l'assenza di alimentazione esterna, differenti protocolli di trasferimento dati per l'accesso a internet, nuovi metodi d'immissione dei dati e ridotte dimensioni dello schermo. Di seguito una descrizione dei principali sistemi operativi attualmente presenti sui dispositivi mobili di ultima generazione.

3.3.1 iOS

Denominato inizialmente iPhone OS, iOS è il sistema operativo sviluppato da Apple per i dispositivi iPod Touch, iPhone, iPad e Apple TV. L'interfaccia di iOS è basata sul concetto di manipolazione diretta tramite l'uso di funzionalità di *gesture* multi-touch⁶. Un esempio di alcune gesture multi-touch utilizzate su sistema operativo di ultima generazione è presente in figura 3.3. La risposta ai comandi dell'utente è immediata e fornisce un'interfaccia flui-

⁶È una tecnologia che rappresenta una evoluzione dello schermo tattile. Uno schermo tattile multitocco si differenzia dai precedenti per il fatto che è sensibile al tocco in più punti diversi della superficie contemporaneamente.

da. L'interazione con il sistema operativo include gesti specifici (denominati gestures) come sfiorare, toccare e pizzicare, tutti dotati di definizioni specifiche nel contesto del sistema operativo iOS e la sua interfaccia multi-touch. Degli accelerometri interni ai dispositivi che montano il sistema operativo iOS sono utilizzati da alcune applicazioni per rispondere allo scuotere del dispositivo o alla rotazione in tre dimensioni. Così come Mac OSX, iOS deriva da Unix e usa un microkernel XNU Mach basato sul sistema operativo Darwin. L'architettura di iOS ha quattro livelli di astrazione: il Core OS layer, il Core Services layer, il Media layer e il Cocoa Touch layer. Oltre alle applicazioni native comprese con il sistema, dal rilascio nel Febbraio del 2008 del kit di sviluppo iOS SDK, anche gli sviluppatori di terze parti hanno avuto la possibilità di sviluppare applicazioni native per i dispositivi iOS da rendere disponibili attraverso la piattaforma App Store. iOS è anche provvisto del browser integrato Safari Mobile che supporta il framework WebKit per la creazione di applicazioni web compatibili.

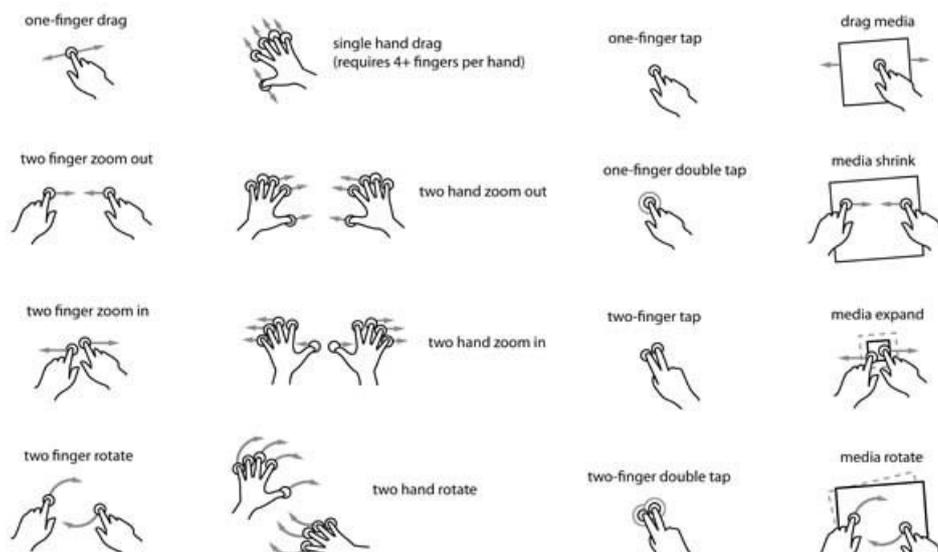


Figura 3.3: Elenco di gesture multi-touch

3.3.2 Android

Android è il sistema operativo per smartphone e tablet sviluppato da Open Handset Alliance guidata da Google. Nel 2005 Google ha acquistato l'azienda sviluppatrice iniziale del software, Android Inc., i quali cofondatori iniziarono a lavorare per Google e svilupparono una piattaforma basata sulla versione 2.6 del kernel Linux. Il 5 novembre 2007 il consorzio di produttori Open Handset Alliance presentò pubblicamente Android. Il sistema operativo Android è attualmente, in termine di vendite, il più diffuso per dispositivi mobili e, a differenza di iOS di Apple, è compatibile con dispositivi di diversi costruttori, come Samsung, LG, HTC e supporta le interfacce multi-touch gestures. Basata su kernel Linux, la piattaforma usa il database SQLite, la libreria dedicata SGL per la grafica bidimensionale e supporta lo standard OpenGL ES 2.0⁷ per la grafica tridimensionale. Le applicazioni vengono eseguite tramite la Dalvik virtual machine, una macchina virtuale adattata per l'uso su dispositivi mobili. Anche Android è fornito di applicazioni proprie con la possibilità di installare software sviluppato da terze parti attraverso i negozi online come Android Market, l'App Store gestito da Google. Tra le applicazioni preinstallate anche Android è munito di un browser basato su framework WebKit per il supporto di applicazioni web.

3.3.3 BlackBerry OS

BlackBerry OS è un sistema operativo mobile proprietario, sviluppato da RIM (Research In Motion) per la sua linea di smartphone BlackBerry. Il sistema supporta dispositivi di input specializzati che sono stati adottati da RIM, in particolare la rotellina per lo scrolling (trackball) e più recentemente il trackpad e lo schermo touchscreen. A partire dal 2011 molti sviluppatori utilizzano la piattaforma BlackBerry per sviluppare le loro applicazioni. Dalla versione 6, BlackBerry OS integra il motore render WebKit per mi-

⁷OpenGL ES è un sottoinsieme delle librerie grafiche OpenGL pensato per dispositivi integrati (telefoni cellulari, PDA, tablet, ma anche strumentazione scientifica e industriale).

gliorare la navigazione internet, supportando HTML5 e le applicazioni web sviluppate con tale linguaggio. Nel settembre 2010, RIM annuncia un nuovo sistema operativo, la piattaforma QNX⁸ per tablet proprietari che andrà a sostituire completamente BlackBerry OS a partire dalla versione 8. Il nuovo sistema, denominato anche BlackBerry Tablet OS, supporta la funzionalità multi-touch gestures. Le applicazioni per BlackBerry sono sviluppabili in Java tramite il BlackBerry Jaca SDK, mentre quelle per piattaforma QNX del tablet mediante BlackBerry Native for Tablet OS.

3.3.4 Windows Phone 7

Windows Phone è il recente sistema operativo per smartphone di Microsoft, presentato al mobile World Congress il 15 febbraio 2010. Windows Phone differisce radicalmente dalle precedenti versioni di Windows Mobile⁹ con le quali è incompatibile e supporta le funzionalità multi-touch gestures, gli schermi capacitivi, oltre ad avere una nuova interfaccia grafica, denominata Metro, e riunisce in una sola piattaforma i contenuti di Xbox Live¹⁰ e Zune¹¹. A luglio 2010 è stata rilasciata la versione beta del kit di sviluppo Windows Phone Developer Tools, un pacchetto software gratuito che permette di sviluppare applicazioni per Windows Phone 7 da parte di terzi e contiene Visual Studio 2010 Express per Windows Phone, Windows Phone Emulator, Silverlight per Windows Phone, Microsoft Expression Blend per Windows Phone e XNA Game Studio 4.0. Come browser, l'ultima versione di Windows Phone 7, denominata Mango, offre una versione di Internet Ex-

⁸QNX è un sistema operativo real-time Unix-like POSIX-compliant commerciale, basato su microkernel e mirato principalmente al mercato dei sistemi embedded.

⁹È un sistema operativo compatto basato sulle API Win32 di Microsoft. Fra i dispositivi che ne possono essere equipaggiati ci sono i Pocket PC, gli Smartphone ed i Portable Media Center.

¹⁰È un servizio online per il gioco multiplayer ideato da Microsoft per le console da gioco Xbox e Xbox 360. Ora è integrato attualmente anche nei dispositivi Windows Phone 7.

¹¹Zune è il nome della linea di prodotti multimediali di Microsoft, che include un lettore multimediale, un client software e il negozio online Zune Marketplace.

plorer Mobile basata su Internet Explorer 9 ed include un maggiore supporto agli standard web quali HTML5, CCS3 e Javascript.

3.3.5 Symbian

Symbian è un sistema operativo per dispositivi mobili prodotto da Symbian Foundation, un'organizzazione non-profit che gestisce Symbian OS. Erede del sistema operativo EPOC¹², Symbian OS gira esclusivamente su processori ARM ed è rilasciato nel 2010 sotto licenza EPL(Eclipse Public License), una licenza libera, dando a chiunque la possibilità di sviluppare applicazioni per tale piattaforma. Essendo la maggiore azionista del Symbian Foundation insieme ad altre società, Nokia ha montato sulla maggior parte dei suoi ultimi dispositivi versioni aggiornate del Symbian OS, anche se attualmente la piattaforma principale è Windows Phone 7. Le ultime versioni per smartphone, Symbian Anna e l'aggiornamento Symbian Belle (rinominato Nokia Belle), supportano le interfacce multi-touch e una migliore integrazione del framework Webkit per la applicazioni web. Per quanto riguarda invece le applicazioni native, il passaggio a Windows Phone 7 da parte di Nokia ha decretato un forte calo di interesse da parte degli sviluppatori di terze parti verso la creazione di software per Symbian OS. Il sistema operativo rimane comunque tra i più diffusi in mercato, con una forte presenza tra i sistemi operativi mobile più venduti e più utilizzati per la navigazione internet.

3.4 Le caratteristiche dei principali sistemi operativi per dispositivi mobili

La tabella 3.1 disponibile di seguito mette in evidenza le caratteristiche tecniche dei principali sistemi operativi per dispositivi mobili attualmente

¹²EPOC è una gamma di sistemi operativi per dispositivi portatili, principalmente palmari sviluppata da Psion. EPOC è l'acronimo di Electronic Piece Of Cheese (Pezzo Di Formaggio Elettronico).

presenti in mercato.

	iOS	Android	BlakBerry OS	Windows Phone 7	Symbian
Compagnia	Apple	Google	RIM	Microsoft	Symbian Foundation
Versione corrente	5.0.1	4.0	6.0.0	7.5	9.5
Famiglia di sistemi operativi	Mac OSX Unix-like	Linux	Mobile OS	Windows CE 7	Mobile OS
Architettura CPU	ARM	ARM, MIPS, Power Architecture, x86	ARM	ARM	ARM, x86
Linguaggio di programmazione	C, C++, Objective C	C, C++, Java	Java	C++	C++
Motore browser di default	WebKit	WebKit	WebKit	Internet Explorer Mobile	WebKit

Tabella 3.1: Principali sistemi operativi per dispositivi mobili

Capitolo 4

Guida Virtuale al Museo della Musica per sistemi iOS

Le ultime ricerche di mercato indicano che i dispositivi mobile sono in forte aumento ogni anno, con una propensione degli utenti verso l'acquisto di dispositivi mobili con sistema operativo Android di Google[3]. Per quanto riguarda invece l'utilizzo di internet su dispositivi mobili, una recente analisi ha evidenziato che il sistema operativo iOS di Apple è quello più utilizzato, con un 55,14% di share rispetto agli altri sistemi operativi per piattaforme portatili[4]. La figura 4.1 mostra un'analisi dei dispositivi più utilizzati per la navigazione in internet effettuata da NetMarketShare.com per il periodo di gennaio-febbraio 2012. Queste analisi di mercato, visto l'alto tasso di diffusione e quindi di accessibilità, ci hanno portato a prendere in considerazione principalmente i dispositivi iOS di Apple, in particolare iPhone, per approfondire lo studio di fattibilità di una Guida Virtuale al Museo Della Musica.

4.1 Possibili scelte progettuali

Per piattaforme iOS, così come per le altre, sono due i possibili approcci per sviluppare un'applicazione per dispositivi mobili. Lo sviluppo può essere

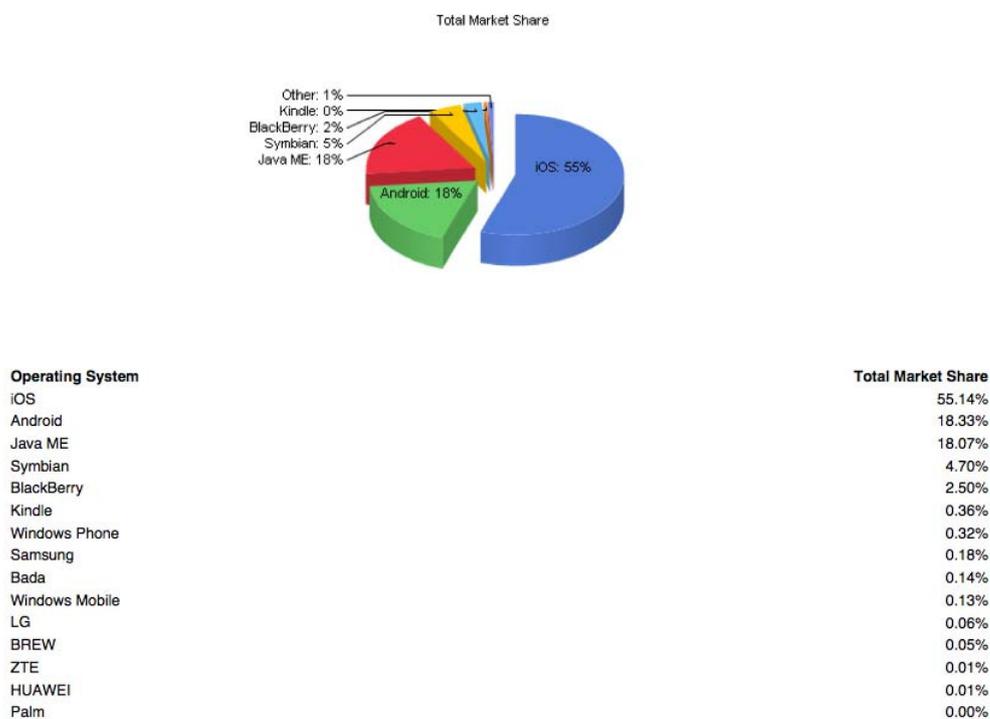


Figura 4.1: I Sistemi Operativi mobile più usati per internet

diretto verso applicazioni di tipo nativo oppure applicazioni di tipo web. La scelta di una di queste due tipologie di sviluppo deve però tenere in considerazione diversi aspetti di progettazione in base all'esigenza dell'applicazione. Di seguito viene fornita un'attenta analisi dei due possibili approcci per lo sviluppo di una Guida Virtuale al Museo della Musica.

4.2 Applicazione Nativa per iOS

Le applicazioni native sono caratterizzate dalla possibilità di interagire direttamente con le API¹ messe a disposizione dal costruttore del sistema

¹Con il termine API (Interfaccia di Programmazione di un'Applicazione) si indica ogni insieme di procedure disponibili al programmatore, di solito raggruppate a formare un set di strumenti specifici per l'espletamento di un determinato compito all'interno di un certo programma. La finalità è ottenere un'astrazione, di solito tra l'hardware e il pro-

operativo del dispositivo. Un'applicazione nativa per iOS è progettata appositamente per il sistema mobile Apple, può essere installata su iPhone e iPad e può avere accesso diretto all'hardware dei dispositivi, come accelerometro, fotocamera, altoparlanti e ricevitore GPS. Lo sviluppo per piattaforma iOS viene fatto con linguaggio Objective-C mediante l'uso di strumenti di sviluppo compresi all'interno dell'iOS SDK, come l'ambiente di sviluppo Xcode e i framework Cocoa Touch. Per lo sviluppo e la pubblicazione di applicazioni native per iOS su App Store², è necessario un computer Mac con processore Intel e l'iscrizione a pagamento all'iOS Developer Program. Gli sviluppatori possono decidere di scegliere qualsiasi prezzo per le loro applicazioni che sono distribuite tramite App Store, per le quali ricevono il 70% del ricavo. Essi possono anche optare per rilasciare l'applicazione in modo gratuito e senza pagare nessun costo aggiuntivo, eccetto la tassa di sottoscrizione al programma per sviluppatori iOS. In questo modo l'applicazione e i seguenti aggiornamenti di essa devono sottostare al processo di approvazione di Apple che, purtroppo, può allungare il ciclo di vita nello sviluppo dell'applicazione.

4.2.1 Linguaggio di sviluppo: Objective-C

Objective-C è un linguaggio di programmazione riflessivo orientato agli oggetti che si basa sul linguaggio C³. Utilizzato per lo sviluppo di applicazioni che girano sui sistemi operativi Mac OSX e iOS, Objective-C è diventato ormai il linguaggio principale che Apple ha adottato per la programmazione per computer Mac e dispositivi iPod Touch, iPhone e iPad. In modo simile a C++⁴, Objective-C identifica un insieme di estensioni al linguaggio grammatore o tra software a basso e quello ad alto livello semplificando così il lavoro di programmazione.

²App Store è un servizio realizzato da Apple disponibile per iPhone, iPod touch e iPad che permette agli utenti di scaricare e acquistare applicazioni disponibili in iTunes Store. Le applicazioni possono essere scaricate direttamente dal dispositivo o su un computer.

³Il linguaggio C è un linguaggio di programmazione procedurale e imperativo, nato per essere utilizzato con il sistema operativo Unix.

⁴Il C++ è un linguaggio di programmazione orientato agli oggetti, con tipizzazione statica. È stato sviluppato da Bjarne Stroustrup ai Bell Labs nel 1983 come un miglioramento

gio C che consentono di implementare interfacce, oggetti, protocolli e altri elementi legati alla programmazione orientata agli oggetti. Tra le caratteristiche dell'Objective-C vi sono anche l'ereditarietà, la gestione automatica e semiautomatica della memoria, le eccezioni, i blocchi e il multithreading. Supporta inoltre il polimorfismo e la ridefinizione dei metodi. Objective-C è quindi un linguaggio potente che però richiede una curva di apprendimento abbastanza ripida per sviluppare applicazioni native per dispositivi iOS.

4.2.2 Ambiente di sviluppo: Xcode



Figura 4.2: Piattaforma di sviluppo Xcode 4

Xcode è l'ambiente di sviluppo integrato (IDE, Integrated Development Environment) sviluppato da Apple per agevolare agli sviluppatori la progettazione di applicazioni dei sistemi Mac OSX e iOS. Xcode fa parte dell'iOS SDK ed include GCC, che è in grado di compilare codice C, C++, Objective C/C++ e Java. Inoltre, supporta, tra i tanti, anche i framework Cocoa e del linguaggio C. Tra i miglioramenti principali troviamo: l'introduzione del paradigma di programmazione a oggetti, funzioni virtuali, overloading degli operatori, ereditarietà multipla, template e gestione delle eccezioni.

Carbon. Grazie a Xcode gli sviluppatori intenzionati a progettare applicazioni per sistemi iOS possono usufruire di una serie di strumenti aggiuntivi come per esempio Interface Builder, un'applicazione integrata in Xcode utile per semplificare lo sviluppo dell'interfaccia grafica. Inoltre, insieme a Xcode è possibile testare direttamente le parti di codice generate utilizzando iPhone Simulator. Il tutto è fornito insieme a una ricca documentazione relativa allo sviluppo per applicazioni Apple. Xcode è attualmente disponibile alla versione 4 ed è compatibile solo con i computer Mac che montano processori Intel. La figura 4.2 mostra l'applicazione Xcode 4 per Mac OSX.

4.2.3 Pro e Contro di una applicazione nativa per iOS

Di seguito vengono riportati i vantaggi e gli svantaggi dello sviluppo di una applicazione nativa per piattaforme iOS.

Pro:

- La possibilità di accedere a tutte le caratteristiche hardware del dispositivo, come accelerometro, GPS, fotocamera, suono e vibrazione.
- Interfaccia utente molto reattiva, dal momento che vengono effettuate chiamate dirette al sistema.
- L'applicazione può funzionare anche in modalità offline, ossia in assenza di accesso a internet.
- Xcode, Interface Builder e il framework Cocoa Touch offrono un comodo e piacevole ambiente di sviluppo.
- Ampia possibilità di visibilità, diffusione e guadagno grazie alla pubblicazione dell'applicazione attraverso App Store.

Contro:

- L'applicazione nativa non può essere compatibile con altri dispositivi mobili. In quel caso si devono sviluppare altrettante copie della stessa applicazione scritte nei diversi linguaggi dei differenti sistemi operativi.
- Per poter rendere pubblica la propria applicazione nativa è necessario pagare l'iscrizione al Developer Program di Apple.
- È necessario sviluppare in ambiente Xcode su macchine con sistema operativo Mac OSX e processore Intel.
- Il processo di approvazione da parte di Apple allunga i tempi di sviluppo e testing, anche per quanto riguarda semplici aggiornamenti e correzioni a bug dell'applicazione, rendendo il tutto poco flessibile.

4.3 WebApp per iOS

Un'applicazione web, chiamata anche webapp o web application, è in pratica un sito web ottimizzato per essere utilizzato da dispositivi mobili. Si tratta di pagine web evolute in grado di offrire funzionalità complesse simili a quelle fornite dalle comuni applicazioni native dei dispositivi mobili. Per la creazione di webapp si possono utilizzare le tecnologie dello standard web, quali HTML5, CSS e JavaScript, compatibili con il motore di rendering WebKit presente in Safari Mobile e nella maggior parte dei browser di dispositivi mobili. I linguaggi di scripting, come per esempio AJAX⁵, consentono di creare degli applicativi veri e propri simili alle comuni applicazioni native, anche dal punto di vista dell'interfaccia grafica. È infatti possibile riprodurre comportamenti tipici delle interfacce software, ad esempio gestendo aggiornamento dei contenuti senza effettuare refresh delle pagine. Mentre

⁵AJAX (Asynchronous JavaScript And XML) è una tecnica per la creazione di applicazioni web interattive. Permette l'aggiornamento asincrono di porzioni di pagine HTML. Non è un linguaggio o una tecnologia specifica, è più che altro un termine che indica l'utilizzo di una combinazione di tecnologie comunemente usate sul web.

una comune applicazione nativa è installata sul dispositivo, le webapp sono invece dei collegamenti verso un applicativo remoto. Se da un lato c'è lo svantaggio che il dispositivo sia connesso alla rete, dall'altro la webapp non richiede alcuno spazio di memoria per funzionare ed è sostanzialmente indipendente dalla capacità di calcolo dell'hardware. La velocità di una webapp dipende infatti principalmente dalla bontà della connessione di rete e dalle prestazioni del server remoto nel ritornare la richiesta effettuata dall'utente. Le webapp inoltre sono compatibili con più dispositivi dal momento che utilizzano linguaggi dello standard web (con qualche accorgimento per l'ottimizzazione) e richiedono un ciclo di vita dello sviluppo più veloce poichè sono indipendenti da processi di approvazione di Apple. La figura 4.3 mostra alcune delle applicazioni web attualmente disponibili per iPhone. Tra queste vi sono applicazioni per organizzare dati di agenda, giochi interattivi e l'app in versione mobile di Facebook.



Figura 4.3: Alcune applicazioni web su Safari Mobile

4.3.1 Il motore WebKit

WebKit è un motore di layout per browser progettato per consentire il rendering delle pagine web. Nato dalla combinazione di componenti del siste-

ma grafico KDE⁶ e di tecnologie Apple, WebKit è il motore di Safari e Google Chrome, oltre ad essere usato anche come base per il browser sperimentale incluso nell'ebook reader Amazon Kindle e per i browser dei sistemi operativi per dispositivi mobili come iOS, Android, BlackBerry OS, BlackBerry Tablet OS e webOS⁷. Il motore WebKit fornisce un insieme di classi per visualizzare il contenuto web in finestre, e implementa le caratteristiche principali del browser, come ad esempio seguire i link attivati dall'utente, gestire la lista di collegamenti back-forward e la cronologia delle pagine visitate. WebKit è un progetto open source sviluppato da KDE, Apple, Nokia, Google, Samsung e altri. Mac OSX, Windows, GNU/Linux e altri sistemi operativi a base Unix sono attualmente supportati dal progetto. WebKit include due framework di basso livello: WebCore (un parser HTML e un renderer basato su KHTML) e JavaScriptCore (un motore JavaScript basato su KJS).

4.3.2 Tecnologie di sviluppo: HTML5, CSS3, JavaScript

Le principali tecnologie dello standard web necessarie per creare applicazioni web per dispositivi mobili sono HTML5, CSS e Javascript, con l'aggiunta di alcune librerie e plug-in per Javascript che rendono l'esperienza d'uso della webapp molto simile a quella di una applicazione nativa. Di seguito una descrizione dei linguaggi sopra elencati.

HTML5: è un linguaggio di markup⁸ (o meglio un insieme di tecnologie) utilizzato per la progettazione di pagine web. Si propone come evoluzione dell'attuale HTML 4.01 ed è stato concepito in alternativa a

⁶KDE (K Desktop Environment) è, come recita la pagina di benvenuto della sua documentazione, un ambiente desktop grafico per postazioni di lavoro Unix.

⁷WebOS è un sistema operativo per dispositivi portatili basato sul kernel Linux con elementi proprietari sviluppato da Palm e successivamente acquistato da Hewlett-Packard

⁸Il linguaggio di markup è un insieme di regole che descrivono i meccanismi di rappresentazione (strutturali, semantici o presentazionali) di un testo che sono utilizzabili su più supporti.

XHTML 2. Le novità introdotte rispetto alla versione precedente sono orientate a migliorare il disaccoppiamento di struttura, definita dal markup, le caratteristiche di resa (come il tipo di carattere e i colori), definite dalle direttive di stile, e i contenuti di una pagina web, definiti dal testo vero e proprio. L'HTML5 prevede anche il supporto per la memorizzazione locale di grosse quantità di dati scaricati dal browser, per consentire l'utilizzo delle applicazioni web anche in assenza di connessione a internet. HTML5 permette anche di sfruttare al massimo alcune delle funzionalità dei dispositivi mobili con un serie di API da utilizzare per accedere a caratteristiche e dati specifici dei vari terminali. Attualmente è già implementata nella maggior parte dei browser, smartphone e tablet la geolocalizzazione⁹, ma sono in fase di progettazione anche altre API per utilizzare la fotocamera, la lista dei contatti o i dispositivi audio. Inoltre sono stati introdotti elementi specifici per il controllo di contenuti multimediali senza più il bisogno di installare plug-in nel browser di terze parti (tipo Adobe Flash o Microsoft Silverlight) ed è stato supportato l'elemento *canvas* che permette di utilizzare JavaScript per creare animazioni e grafica bitmap. La figura 4.4 mostra il logo del linguaggio HTML5.

CSS3: conosciuto anche come Cascading Style Sheet, il CSS è un linguaggio usato per definire la formattazione delle pagine HTML, XHTML e XML. Tale linguaggio è incluso in file detti fogli di stile, o meglio style sheets. Con l'introduzione dei fogli di stile si è voluto separare definitivamente il contenuto delle pagine web dalla formattazione, ottenendo così un codice più chiaro e di più semplice manutenzione, oltre che maggiormente portabile. Il CSS3 è l'ultima versione del linguaggio in attesa di una standardizzazione vera e propria ma i moderni browser (tra questi tutti quelli con motore WebKit) implementano già molte delle sue funzioni. Tra queste vi sono nuovi effetti grafici, come

⁹È l'identificazione della posizione geografica nel mondo reale di un dato oggetto, come ad esempio un telefono cellulare o un computer connesso a Internet.



Figura 4.4: Il logo di HTML5

ombreggiatura, effetto ombra, trasformazioni e animazioni. Inoltre si possono applicare sfondi multipli ad uno stesso elemento, organizzare contenuti su più colonne e creare box con angoli arrotondati senza l'uso di immagini aggiuntive.

JavaScript: è un linguaggio orientato agli oggetti comunemente usato per la progettazione di siti web. Sviluppato originariamente da Brendan Eich della Netscape Communications con il nome di Mocha e successivamente di LiveScript, fu rinominato in seguito JavaScript e venne formalizzato con una sintassi più vicina a quella del linguaggio Java di Oracle, anche se ci sono notevoli differenze. È un linguaggio interpretato e viene eseguito direttamente sul client e non sul server. Il vantaggio di questo approccio è che, anche con la presenza di script particolarmente complessi, il server non viene sovraccaricato a causa delle richieste dei client. Un browser tipicamente incorpora un interprete JavaScript e quando viene visitata una pagina web che contiene codice JavaScript, quest'ultimo viene portato in memoria ed esegui-

to dall'interprete contenuto nel browser. Le interfacce che consentono a JavaScript di rapportarsi con un browser sono chiamate DOM¹⁰. JavaScript in ambito web è usato principalmente per la scrittura di piccole funzioni integrate nelle pagine HTML che interagiscono con il DOM del browser per compiere alcune azioni non possibili normalmente con l'HTML statico, come per esempio controllare i valori presenti nei campi di input, nascondere o visualizzare determinati elementi.

4.3.3 Pro e Contro di una webapp per iOS

Di seguito sono elencati i vantaggi e gli svantaggi per quanto riguarda lo sviluppo di applicazioni web per sistemi operativi iOS.

Pro:

- L'applicazione è multiplatforma, ossia è compatibile con qualsiasi dispositivo che integri un browser web e non c'è bisogno di creare più copie di applicazione. È solo necessario qualche ottimizzazione del codice per adattarla al meglio su ogni piattaforma.
- I costi di sviluppo sono notevolmente più bassi rispetto a quelli per la creazione di un'applicazione nativa.
- È possibile sviluppare su qualsiasi computer senza alcuna restrizione per quanto riguarda l'ambiente di sviluppo e il sistema operativo.
- Non è sottoposta a nessun tipo di approvazione da parte di Apple.
- Ciclo di vita di sviluppo più veloce, con possibilità di rilasciare l'applicazione o aggiornarla in tempo reale.

¹⁰Document Object Model (spesso abbreviato come DOM) è lo standard ufficiale del W3C per la rappresentazione dei documenti strutturati in maniera da essere neutrali sia per la lingua che per la piattaforma.

Contro:

- Non si può accedere a tutte le caratteristiche hardware del dispositivo.
- Si deve sviluppare un proprio metodo di pagamento se si vuole mettere in vendita l'applicazione.
- Prestazioni meno reattive di un'applicazione nativa.
- Può essere difficile ottenere sofisticati effetti nell'interfaccia utente.

Capitolo 5

La WebApp

L'analisi dei due tipi di approccio da poter intraprendere per lo sviluppo di un'applicazione relativa a una Guida Virtuale al Museo della Musica per iOS ci ha permesso di valutare come più vantaggioso il processo di progettazione mediante standard web. Tralasciando il lato economico, il fattore che più ha inciso sulla nostra decisione è stato principalmente l'alto tasso di compatibilità che una webapp può avere con diversi dispositivi, che siano smartphone, tablet o computer laptop e desktop. Questo permette infatti di poter progettare da principio un unico portale web accessibile da dispositivi con differente sistema operativo e hardware. Ora però andiamo a vedere quali siano gli strumenti per poter realizzare le classiche funzionalità di un'applicazione nativa con una webapp, tenendo in considerazione i requisiti analizzati in precedenza per una Guida Virtuale al Museo della Musica.

5.1 L'architettura di una applicazione web

Generalmente le applicazioni web sono strutturate su un'architettura client-server¹ a tre livelli in cui la presentazione, l'elaborazione dei processi e la

¹Un'applicazione client-server (letteralmente cliente-servente) è un tipo di applicazione di rete nel quale un computer client istanzia l'interfaccia utente di un'applicazione connettendosi ad una server application o ad un sistema di database.

gestione dei dati sono processi logicamente distinti. Questo modello è chiamato architettura Three-tier ed è composto, appunto da tre moduli o strati indipendenti che sono il Livello di Presentazione, il Livello Applicazione e il Livello Dati. Il Livello Presentazione è quello più alto ed è indentificato dal browser web del dispositivo client da cui l'utente fa richieste mediante l'interfaccia dell'applicazione. Le richieste sono inviate al livello intermedio, denominato Livello Applicazione, ovvero il motore dinamico del web server il quale grazie a un linguaggio di scripting per creazione di pagine dinamiche si occupa da una parte di interpretare e gestire le interrogazioni al motore database del Livello Dati e dall'altra di generare il risultato delle richieste in output verso il browser web che lo interpreta e lo restituisce all'utente sotto forma di pagine web. La figura 5.1 mostra un esempio grafico della struttura Three Tier per applicazioni web.

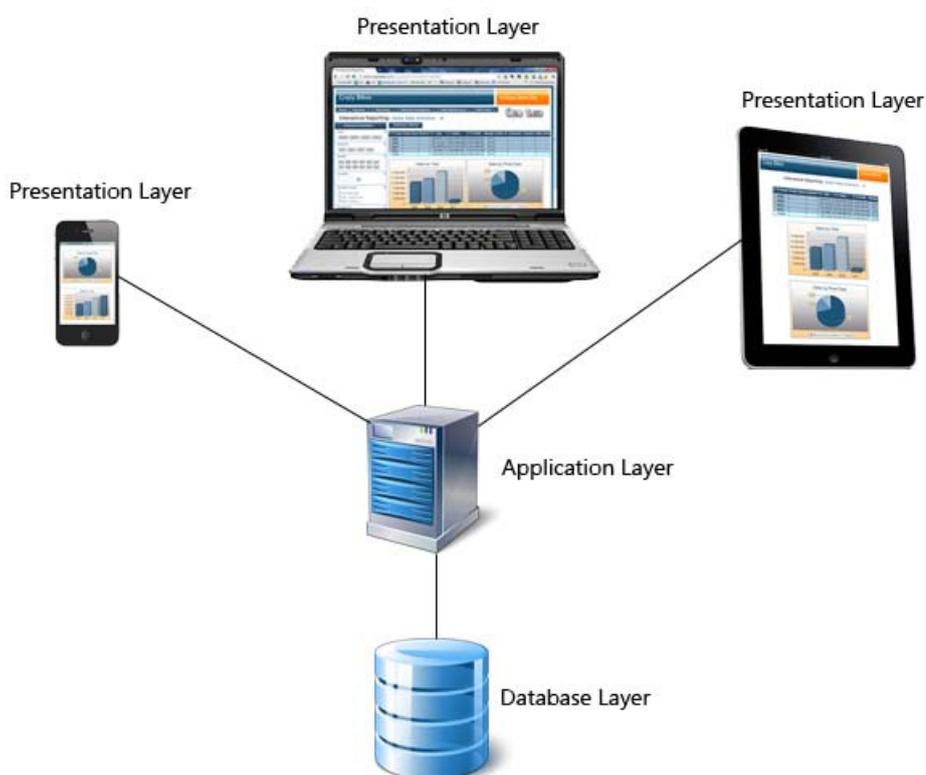


Figura 5.1: Il modello Three Tier per applicazioni web

Da notare la compatibilità multiplatforma del Presentation Layer, ossia il Livello Presentazione. L'architettura Three Tier può essere composta da svariati browser client, un web server Apache HTTP Server con linguaggio PHP e un motore MySQL per il database.

Apache HTTP Server: è il web server modulare più diffuso al mondo in grado di operare da sistemi operativi UNIX/Linux e Microsoft. Operativamente, è composto da un demone², in ambiente UNIX, o da un servizio, in ambiente Microsoft, che sulla base delle impostazioni contenute nel file di configurazione *httpd.conf* permette l'accesso a uno o più siti, gestendo varie caratteristiche di sicurezza e potendo ospitare diverse estensioni per pagine attive (o dinamiche), come PHP o Jakarta/Tomcat.

PHP: 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. Attualmente è utilizzato principalmente per sviluppare applicazioni web lato server ma può essere usato anche per scrivere script a riga di comando o applicazioni standalone con interfaccia grafica. Fornisce un'API specifica per interagire con Apache, nonostante funzioni naturalmente con numerosi altri server web. È anche ottimamente integrato con il database MySQL, per il quale possiede più di una API.

MySQL: è un Relational Database Management System³, composto da un client con interfaccia a caratteri e un server, entrambi disponibili sia per sistemi Unix come GNU/Linux che per Windows. Uno dei programmi più utilizzati per amministrare i database MySQL è phpMyAd-

²In informatica un demone (daemon in inglese) è un programma eseguito in background dal sistema operativo, senza che sia sotto il controllo diretto dell'utente.

³Anche conosciuto come RDBMS, indica un database management system basato sul modello relazionale.

min⁴ (richiede un server web come Apache HTTP Server ed il supporto del linguaggio PHP). Si può utilizzare facilmente tramite un qualsiasi browser.

5.2 L'interfaccia grafica

Il Livello Presentazionale del modello Three Tier è composto dal browser client e nel nostro caso principalmente da Safari Mobile per iOS. Come abbiamo già anticipato, l'interfaccia grafica dell'applicazione web può essere implementata utilizzando gli standard web HTML5, CSS e JavaScript in modo da essere il più simile possibile, sia nell'aspetto che nei comportamenti, ad una applicazione nativa.

5.2.1 Look and Feel di una applicazione nativa

Esistono alcuni semplici strumenti per rendere da subito una pagina web molto simile ad un'applicazione nativa. Per esempio, attraverso Safari Mobile è possibile aggiungere un'icona Web Clip alla propria schermata Home come fosse un'applicazione. In questo caso è possibile allegare alla pagina web un'icona personalizzata aggiungendo una semplice linea di codice html nella sezione *head* del file. Ecco un esempio:

```
<head>
<link rel="apple-touch-icon" href="IconaWebApp.png" />
</head>
```

In questo modo è possibile accedere alla pagina web direttamente dalla schermata home tramite l'apposita icona personalizzata. Per donare ulteriormente un aspetto nativo alla applicazione web è possibile eliminare la barra

⁴PhpMyAdmin è un'applicazione PHP libera che consente di amministrare in modo semplificato database di MySQL tramite un qualsiasi browser. L'applicazione è indirizzata sia agli amministratori del database, sia agli utenti. Gestisce i permessi prelevandoli dal database MySQL.

di Safari Mobile consentendo di avere una modalità a schermo intero con il seguente codice:

```
<head>
<meta name="apple-mobile-web-app-capable" content="yes" />
</head>
```

Una volta inserito il meta tag *apple-mobile-web-app-capable* si ha la possibilità di controllare anche il colore di sfondo della barra di stato posta in alto nella schermata, tramite il tag:

```
<head>
<meta name="apple-mobile-web-app-status-bar-style" content="black" />
</head>
```

Inoltre, all'avvio della webapp è possibile aggiungere anche una schermata iniziale, proprio come una applicazione nativa. Sempre all'interno del tag head bisogna inserire il seguente codice:

```
<head>
<link rel="apple-touch-startup-image" href="ImmagineStartup.png" />
</head>
```

Queste sono semplici modifiche che, se unite insieme a un buon utilizzo di HTML, CSS, JavaScript e Ajax, possono rendere una normale pagina ottimizzata per iPhone come una vera e propria applicazione[5]. Per assottigliare sempre di più il divario tra pagina web e applicazione nativa vengono in aiuto anche dei framework e plug-in open source che consentono di ampliare le prestazioni dell'interfaccia delle applicazioni preservando la compatibilità per diversi dispositivi mobili. La figura 5.2 mostra l'evoluzione di un'applicazione web mediante gli strumenti appena descritti che abbiamo implementato nel prototipo di una Guida Virtuale al museo della musica. Per questo esempio abbiamo trasformato una normale pagina web in un software con interfaccia simile alle applicazioni native per iOS, emulando anche l'installazione sul dispositivo.

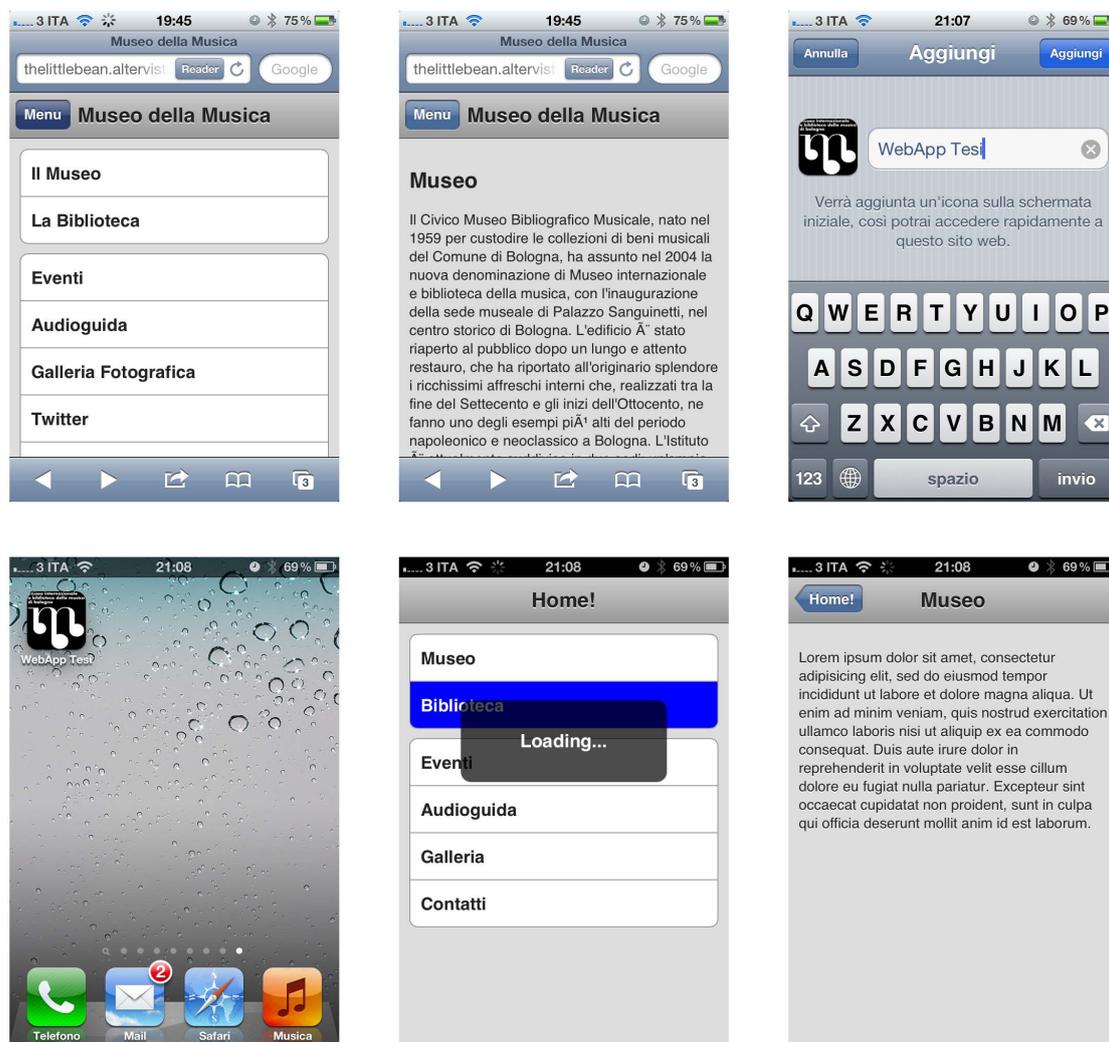


Figura 5.2: Alcune modifiche dell'applicazione web per iOS

Di seguito un elenco di framework e plug-in più utilizzati per lo sviluppo di applicazioni web.

JQuery: è una libreria di funzioni javascript, multipiattaforma per le applicazioni web, che ha l'obiettivo di semplificare la programmazione lato client delle pagine HTML. Tramite l'uso della libreria JQuery è possibile, con poche righe di codice, effettuare svariate operazioni, come ad esempio ottenere l'altezza di un elemento, o farlo scomparire con effetto

dissolvenza. Anche la gestione degli eventi è completamente standardizzata e gestita automaticamente, assieme alla loro propagazione. Stessa cosa per quanto riguarda l'utilizzo di AJAX, in quanto sono presenti alcune funzioni molto utili e veloci che si occupano di istanziare i giusti oggetti ed effettuare la connessione e l'invio dei dati.[6]

JQuery Mobile: è un framework per applicazioni web ottimizzata per dispositivi mobili con interfaccia tattile sviluppata dal team di JQuery. Lo sviluppo è principalmente incentrato sulla compatibilità con una vasta gamma di smartphone e tablet pc.[7]

JQTouch: è un plug-in di JQuery che consente l'implementazioni di animazioni native, navigazioni automatiche e diversi temi per i dispositivi mobili che supportano WebKit. Esso permette di sviluppare applicazioni web con un look and feel nativo per il dispositivo di destinazione utilizzando HTML, CSS e JavaScript.[8] La figura 5.3 mostra l'interfaccia del nostro prototipo implementata mediante JQTouch.

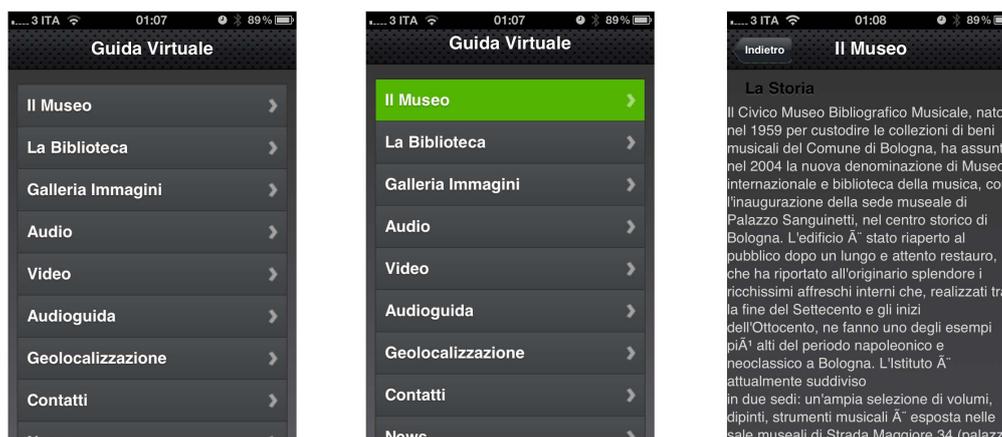


Figura 5.3: Esempio di interfaccia con JQTouch

Sencha Touch: potente piattaforma sviluppata espressamente per creare webapp per dispositivi mobili con HTML5, sfruttando le capacità di

CSS e JavaScript. Compatibile con i browser WebKit, utilizza un modello MVC⁵ per lo sviluppo.[9]

The-M-Project: framework JavaScript che adotta il paradigma MVC. Compatibile sia con dispositivi che supportano WebKit sia con Windows Phone 7.[10]

Entrambi i framework possono essere scaricati gratuitamente dalle pagine ufficiali dei progetti e implementati nella propria applicazione semplicemente includendo i file JavaScript all'interno del codice html.

5.3 I contenuti multimediali

La consultazione di materiale multimediale mediante l'applicazione web della Guida Virtuale è possibile utilizzando le nuove API comprese in HTML5. Gli utenti possono arricchire in questo modo la propria ricerca di informazioni relative a un'opera o un artista con immagini, audio e video.

5.3.1 La galleria immagini

La normale visualizzazione di immagini mediante dispositivi mobili può essere arricchita con l'uso di diversi framework per rendere compatibile la consultazione del materiale mediante le gesture multi-touch. Un esempio di implementazione è possibile utilizzando la piattaforma PhotoSwipe compatibile con JQuery Mobile. L'integrazione dei file JavaScript all'interno del codice HTML permettono una visualizzazione a ghiera di miniature, le quali, una volta selezionate, aprono una galleria immagini a tutto schermo con la quale è possibile interagire mediante gesture, ossia Slide e Zoom.[11] La figura 5.4 mostra un esempio di galleria immagini utilizzando la piattaforma PhotoSwipe.

⁵Model-View-Controller (MVC, talvolta tradotto in italiano Modello-Vista-Controllo) è un pattern architetturale molto diffuso nello sviluppo di interfacce grafiche di sistemi software object-oriented.



Figura 5.4: Esempio galleria immagini con PhotoSwipe

5.3.2 Video e audio

GLi standard web HTML, CSS e JavaScript forniscono tutto il necessario per pubblicare testo, immagini e aggiungere interazione alle applicazioni web. Più complicato è stato invece portare su dispositivi mobili la compatibilità con contenuti audio e video, in particolare su iOS a causa della nota incompatibilità con plug-in come Adobe Flash. Fortunatamente le nuove API di HTML5 riescono a colmare questa lacuna introducendo i nuovi tag per inserire contenuti audio e video senza il bisogno di plug-in esterni installati sul dispositivo.

Il tag video: una delle incompatibilità più note con i dispositivi iOS è appunto quella con il plug-in Flash che permette di visualizzare contenuti video su siti web per computer ma non su iPhone e iPad. Il nuovo tag introdotto con HTML5 ci permette ora di visualizzare video in modo nativo, senza l'ausilio di nessun supporto aggiuntivo. Normalmente iPhone e iPad per la visualizzazione di video lanciano un media player integrato in iOS che occupa l'intero schermo, dove l'utente per ritornare alla pagina deve utilizzare il pulsante *Done*. Per inserire un filmato presente sul proprio server web all'interno di una webapp è necessario il seguente codice di esempio:

```
<video width="100%" src="video.mp4" controls></video>
```

Il tag video può essere implementato con diversi attributi. *Controls* per esempio permette al browser di mostrare i controlli del lettore video, mentre *Poster* aggiunge un'immagine di anteprima. iOS supporta i formati video H.264, MPEG-4 e M-JPG.

Il tag audio: così come per il tag video, anche l'audio può essere inserito nell'applicazione web permettendo di ascoltare brani con la riproduzione audio nativa del browser Safari Mobile. Funziona come il tag video ma è compatibile con un sottoinsieme dei suoi attributi. Ecco il codice di esempio:

```
<audio src="audio.mp3" controls autobuffer></audio>
```

Il codice di esempio mostra oltre all'attributo per i controlli del lettore audio anche un attributo per bufferizzare il file audio durante l'ascolto. iOS è compatibile con file audio di tipo AAC, HE-ACC e MP3.[12]

5.4 L'audioguida

Come abbiamo anticipato in sede di analisi dei requisiti, la Guida Virtuale al Museo della Musica deve poter essere utilizzata anche come strumento di supporto durante una visita al museo. Sfruttando le possibilità offerte da HTML5, CSS e JavaScript è possibile creare una audioguida per offrire agli utenti la consultazione di commenti o brani audio relativi al materiale esposto nel percorso museale. L'idea principale è quella di emulare le attuali audioguide noleggiabili all'interno dei musei con l'implementazione di un lettore multimediale integrato nella webapp. Un plug-in di JQuery, chiamato JPlayer, permette di creare playlist di tracce audio in streaming utilizzando HTML5, CSS e JavaScript per la visualizzazione di un'interfaccia in stile media player con controlli a schermo e possibilità di ascoltare diversi formati

audio. Utilizzando JPlayer sarebbe possibile offrire agli utenti un'audioguida con relative tracce audio correlate ai reperti museali mediante un codice identificativo.[13] La figura 5.5 mostra un esempio di lettore audio sviluppato mediante JPlayer da implementare come strumento per l'audioguida.

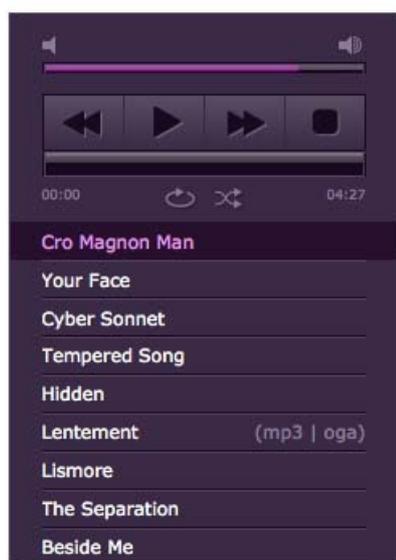


Figura 5.5: Esempio del lettore JPlayer

5.5 La geolocalizzazione

Un altro servizio offerto dalle nuove API di HTML5 è la geolocalizzazione mediante il GPS integrato all'interno dei sistemi iOS. In passato per localizzare un dispositivo si faceva uso dell'IP per fare una stima della posizione geografica in cui si trovava l'utente. Ora i browser più moderni, come Safari Mobile e tutti quelli con motore WebKit, supportano le API di HTML5 permettendo il recupero delle informazioni geografiche di posizionamento via JavaScript. Pensando in particolare a una Guida Virtuale al Museo della Musica, tale servizio potrebbe essere implementato nell'applicazione web per fornire agli utenti le posizioni geografiche di strutture correlate al museo. Il

Museo Internazionale e Biblioteca della Musica di Bologna offre ai propri visitatori una guida sotto forma di libro dal titolo *... e tutta la città era in suoni. I luoghi della storia della musica di Bologna* che fornisce informazioni su tutti i luoghi della musica di Bologna mostrando le posizioni su di una mappa. Tramite le API di geolocalizzazione è ora possibile implementare in una webapp le informazioni geografiche di questi luoghi con visualizzazione su mappa e fornire indicazioni per il raggiungimento del luogo di destinazione. Inoltre, in questo modo sarebbe possibile anche indicare ai turisti i servizi adiacenti al museo, come alberghi, locali e ristoranti. La figura 5.6 mostra l'implementazione della funzionalità di geolocalizzazione nel nostro prototipo di Guida Virtuale al museo della musica. L'applicazione all'avvio chiede all'utente se accettare di condividere la propria posizione. Il nostro prototipo ritorna semplicemente la ricezione dei dati di geolocalizzazione del dispositivo stampando a schermo Latitudine e Longitudine.



Figura 5.6: Esempio servizio di geolocalizzazione

5.6 L'interazione con i servizi nativi di iOS

Nonostante una webapp non riesca ad utilizzare tutte le caratteristiche hardware dei dispositivi iOS, sono disponibili alcuni strumenti per poter far interagire un'applicazione web con i servizi nativi del sistema operativo Apple. In particolare con le applicazioni Telefono, Mail, Messaggi e Mappe, che possono essere utili per implementare un servizio di tipo *Contatti* all'interno della webapp. Per fare questo si utilizza il tag *a* di HTML in questo modo:

```
<a href="protocolServiceScheme:protocolParameter">NomeLink</a>
```

Al posto di *protocolServiceScheme* si deve indicare l'applicazione con cui interagire, mentre nel campo *protocolParameter* si indica il parametro da passare.

Telefono e Messaggi

Per avviare una chiamata:

```
<a href="tel:1-000-555-5000">Invia messaggio 1-000-555-5000</a>
```

Per inciare un messaggio SMS o iMessage:

```
<a href="sms:1-000-555-5000">Chiama 1-000-555-5000</a>
```

Indicando come protocolli di servizio *tel* o *sms* e come parametro il numero di telefono. La figura 5.7 mostra un esempio dell'interazione tra il nostro prototipo di applicazione web e i servizi nativi di iOS per il menù Contatti.

Mail

Per inviare un'e-mail è possibile utilizzare il seguente protocollo:

```
<a href="mailto:info@museo.it">Contatta il Museo via mail</a>
```

Indicando come protocollo di servizio *mailto* e come parametro l'indirizzo e-mail di destinazione. È inoltre possibile specificare attributi all'interno del tag relativi al corpo del messaggio, l'oggetto dell'e-mail, multipli indirizzi di destinazione, il CC Recipients e il BBC recipients.

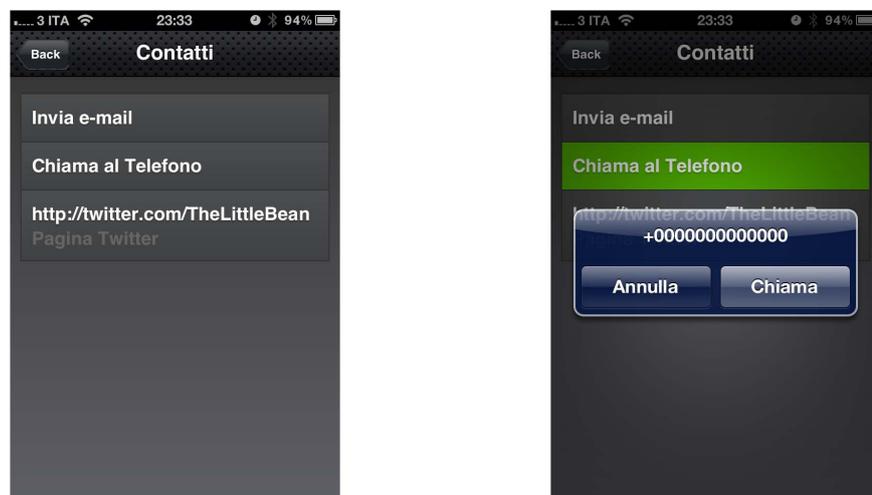


Figura 5.7: Esempio servizio Contatti

Mappe

Un uso alternativo alle API di HTML5 per la geolocalizzazione all'interno della webapp è possibile richiamando il servizio Mappe di iOS. In questo caso però si è obbligati ad uscire dall'applicazione. Per utilizzare Mappe si deve inserire il seguente codice:

```
<a href="http://maps.google.com/maps?q=bologna">Mappa</a>
```

L'applicazione Mappe non dispone di un proprio schema di mappe e il link alla mappa viene specificato utilizzando una normale sintassi del protocollo HTTP. Safari reindirizza la richiesta HTTP al server delle mappe di Google, `maps.google.com`, e quindi apre la sua risposta HTTP utilizzando l'applicazione nativa Mappe.

5.7 Memorizzazione dei dati in una webapp

Uno dei limiti delle pagine web in passato era l'impossibilità di memorizzare in modo persistente i dati lato client. Con l'avvento di HTML5 sono

invece state fornite della API che permettono ad applicazioni web di utilizzare uno strumento per il salvataggio dei dati sul dispositivo. Sono tre i sistemi di salvataggio persistente possibili per un'applicazione web: `localStorage`, `sessionStorage` e il database lato client. `localStorage` e `sessionStorage` sono molto simili ai cookie, in quanto permettono di utilizzare JavaScript per impostare coppie nome/valore che si possono recuperare in caricamenti multipli di una pagina. Tuttavia, a differenza dei cookie i dati trattati con `localStorage` e `sessionStorage` non sono inviati con le richieste del browser attraverso la rete ma risiedono nel client. In questo modo è possibile memorizzare molti più dati rispetto a quanto si può con i cookie.

localStorage: i dati rimangono salvati anche dopo che la finestra dell'applicazione web viene chiusa e sono disponibili per tutte le schede o finestre caricate dallo stesso URL. Questo tipo di memorizzazione è più utile per salvare le preferenze utente.

sessionStorage: i dati sono memorizzati con la singola finestra. Le restanti finestre o schede non possono accedere ai dati memorizzati, che vengono cancellati una volta chiusa la finestra. Questo tipo di memorizzazione è utile per dati specifici della finestra, come l'ordinamento di una tabella.

Database lato client: le nuove caratteristiche di HTML5 permettono di utilizzare un semplice database in JavaScript per memorizzare in modo persistente dati in modo relazionale. È possibile utilizzare il linguaggio SQL standard per creare tabelle, inserire, aggiornare o eliminare elementi del database. Le API dei database JavaScript supportano inoltre anche le transazioni.

Queste nuove possibilità di memorizzare dati all'interno dell'applicazione introducono una serie di servizi che possono essere offerti agli utenti finali. Rimanendo in tema di Guida Virtuale, l'utente utilizzando l'applicazione potrebbe salvare gli elementi di interesse in un personale database relazionale in modo da poter tenere in memoria le informazioni relative alle opere o agli

artisti preferiti da poter consultare in un secondo momento senza bisogno di effettuare nuovamente ricerche nel database del server web. Il tutto richiede però il collegamento a internet, poichè l'applicazione per fornire i dati salvati deve comunque collegarsi al codice presente nel server remoto per scaricare le risorse correlate.

5.8 La modalità offline

Per ovviare all'assenza di connessione, fortunatamente HTML5 permette di eseguire applicazioni web anche quando non c'è collegamento a internet. Utilizzando la modalità offline di HTML5 possiamo risolvere questo problema memorizzando qualsiasi tipo di risorsa all'interno della cache della webapp. I file che devono essere memorizzati nella cache vanno dichiarati all'interno di una file *.manifest*. Safari cercherà tale file prima di ogni sua elaborazione a lato server, evitando il download di file precedentemente scaricati e memorizzati. In questo modo i dati nella cache rimangono persistenti tra le varie sessioni del browser, quindi una risorsa presente in precedenza nella cache può continuare a essere accessibile anche in assenza di rete o, nel caso di iOS, in Modalità Aereo. Grazie a questa nuova funzione di HTML5 è quindi possibile garantire agli utenti un uso dell'applicazione web il più possibile simile a quella di una applicazione nativa. Unendo tale funzionalità alla possibilità di memorizzare dati specifici in modo persistente, gli utenti potrebbero selezionare le informazioni relative a un percorso museale prima di andare a visitare il museo, prevenendo l'assenza di connessione all'interno dell'istituto.

Capitolo 6

La Realtà Aumentata

Una delle principali sperimentazioni in ambito museale degli ultimi tempi è l'uso della Realtà Aumentata come strumento di interazione tra i visitatori di un museo e le esposizioni. La Realtà Aumentata consiste nella sovrapposizione di elementi digitali su una visione dal vivo di uno spazio reale, attraverso uno smartphone o dispositivi mobili con fotocamera integrata. Purtroppo però, attualmente le tecnologie degli standard web di HTML5, CSS e JavaScript non permettono l'accesso diretto alla fotocamera dei sistemi portatili, anche se ci sono sperimentazioni che porteranno a futuri sviluppi di API apposite. Per sorpassare questo ostacolo al momento è possibile solamente trasformare un'applicazione web in una applicazione nativa, ma senza sviluppare una copia in linguaggio nativo per dispositivi, bensì creando un'applicazione ibrida aggiungendo le funzionalità di applicazioni native con l'ausilio di piattaforme dedicate.

6.1 L'applicazione ibrida

L'applicazione ibrida è un webapp sotto forma di applicazione nativa, con i relativi vantaggi che derivano dalle due tipologie. Grazie a strumenti open source come PhoneGap, l'applicazione web può essere confezionata come una vera applicazione nativa e in questo modo gli sviluppatori possono accede-

re a funzionalità avanzate dei dispositivi e pubblicare l'applicazione per la distribuzione su App Store. Il tutto, come dimostrano svariati documenti, manuali e schede tecniche di applicazioni reperibili via internet, garantendo un ciclo di vita di sviluppo molto veloce.

6.1.1 Il Framework PhoneGap

PhoneGap è una framework di sviluppo open source realizzato da Nitobi Software che consente la programmazione di applicazioni native per dispositivi mobili utilizzando gli standard web HTML5, CSS e JavaScript invece dei linguaggi nativi dei sistemi operativi mobili. Attualmente permette una piena compatibilità con i maggiori sistemi operativi per piattaforme mobili e con le relative caratteristiche hardware. In questo modo, tramite JavaScript è possibile accedere alle funzionalità hardware del dispositivo che normalmente erano inaccessibili, mantenendo compatibilità e garantendo la possibilità di pubblicare l'applicazione sugli store digitali per una maggiore visibilità.[14]

6.2 La piattaforma Layar

Layar è una piattaforma per la realtà aumentata progettata per dispositivi mobili, come iPhone e smartphone Android. Disponibile come applicazione gratuita, gli sviluppatori possono anche integrare un lettore di realtà aumentata, chiamato Layar Player, all'interno della propria applicazione nativa. La piattaforma Layar permette di sovrapporre alle immagini riprese con la fotocamera dei dispositivi dei layer informativi altamente interattivi. In particolare in base alla geolocalizzazione dei dispositivi è possibile visualizzare attraverso lo schermo la distanza da un punto di interesse, impostare la navigazione al punto di interesse via Google Maps o consultare delle informazioni relative al punto di interesse selezionato, che possono essere collegamenti a pagine web, immagini o contenuti video e audio. I punti di interesse vengono georeferenziati indicando le coordinate di latitudine e longitudine e per ogni punto è possibile mostrare una serie di informazioni. Tutti i dati risiedono

all'interno di un database SQL nel quale si memorizzano i punti di interesse che devono essere gestiti dal sistema, oltre i file relativi all'aspetto grafico delle informazioni visualizzate a schermo e le azioni impostate per ogni punto. Una volta lanciato il lettore di layer della propria applicazione viene scaricato il layer informativo dei punti di interesse dal sito ufficiale Layar per poter visualizzare su schermo le informazioni relative al punto di interesse. In questo modo il dispositivo utilizzando il GPS interno e la Bussola localizzerà la posizione in relazione all'orientamento, interrogando in seguito il database per visualizzare i punti di interesse su schermo. La figura 6.1 mostra l'architettura della piattaforma Layar per la Realtà Aumentata su dispositivi iOS e Android.

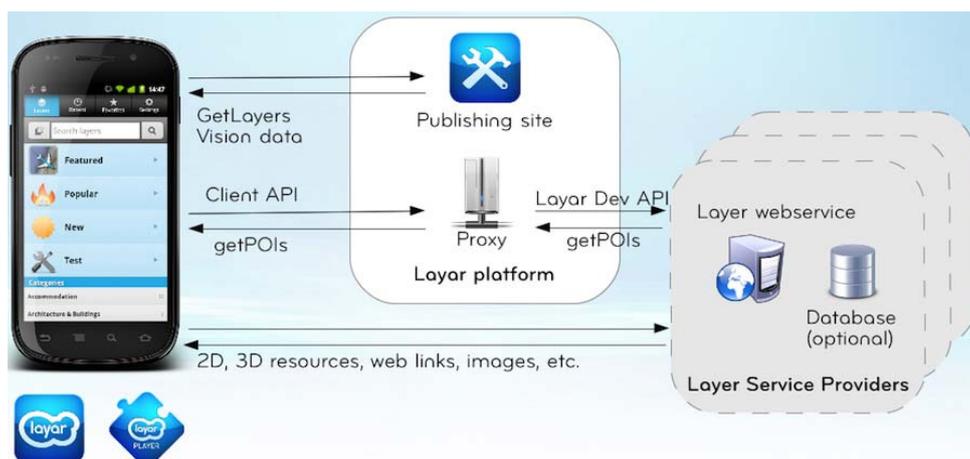


Figura 6.1: Architettura della piattaforma Layar

6.2.1 Layar Vision

Layar Vision è un'estensione della piattaforma Layar per sistemi iOS e Android che permette la creazione di strati informativi legati a oggetti o immagini del mondo reale. Oltre a sfruttare le capacità del GPS e della bussola, Layar Vision utilizza anche tecniche di riconoscimento visivo, in modo simile ai QR Code, per associare, in tempo reale, informazioni di vario genere oppure mostrare animazioni dell'oggetto ricostruito in 3D. Per creare un layer di

tipo Vision è necessario caricare sui server di Layar l'immagine di riferimento dell'oggetto e associare le relative azioni. Utilizzando questa tecnologia, una Guida Virtuale al Museo della Musica potrebbe ampliare l'esperienza esplorativa del percorso museale fornendo azioni precise legate ad ogni oggetto esposto. In particolare, gli spartiti delle opere presenti nelle teche potrebbero essere associate alla traccia audio del brano stesso, oppure potrebbero essere allegate ad essi le rappresentazioni in tre dimensioni degli strumenti utilizzati. Bisogna però considerare che attualmente la conversione della piattaforma Layar attraverso PhoneGap è sprovvista di qualsiasi documento o manuale, quindi sarebbe necessario dedicare uno studio a parte per approfondire ulteriormente lo sviluppo di una tecnologia tipicamente nativa per la compatibilità con applicazioni web.

Conclusioni e sviluppi futuri

Obiettivo principale di questo documento era la valutazione delle tecnologie che possono permettere la realizzazione di un software per Musei di Musica. Dopo una definizione introduttiva degli istituti museali, tenendo in considerazione il Museo Internazionale e Biblioteca della musica di Bologna, si è proseguiti nel descrivere i requisiti di una Guida Virtuale per Musei della Musica durante la quale sono state evidenziate le due modalità d'uso fondamentali: la modalità d'uso in remoto, necessaria a fornire agli utenti un'esplorazione virtuale dei musei, e la modalità in loco che deve permettere l'utilizzo del software anche durante una reale visita agli istituti museali. Questi requisiti hanno poi orientato la ricerca delle piattaforme hardware verso dispositivi mobili come smartphone e tablet, i quali permettono in primis l'utilizzo in mobilità e in secondo luogo garantiscono l'accessibilità a un'ampio numero di possibili utenti poichè risultano tra i più diffusi in mercato. Dopo aver fornito un panorama delle diverse piattaforme mobili più utilizzate in termini di hardware e software, si sono analizzate le diverse scelte progettuali di una piattaforma in particolare, il sistema operativo iOS di Apple, attualmente il dispositivo mobile più utilizzato per la navigazione in internet. Lo studio di fattibilità è proseguito con una valutazione delle scelte progettuali per lo sviluppo di una applicazione per iPhone, tenendo sempre in considerazione la possibilità di offrire un unico software multipiattaforma, ossia compatibile con il maggior numero di sistemi operativi. Sono stati presi in esame lo sviluppo di una applicazione nativa e quello di una applicazione web, elencando vantaggi e svantaggi di entrambe. In base ai requisiti posti

in fase di analisi si è deciso di approfondire lo studio di fattibilità di una webapp con l'ausilio degli standard web HTML5, CSS e Javascript, cercando di andare a colmare le lacune che comporta questo genere di approccio progettuale. L'analisi della progettazione per sistema operativo iOS ha quindi focalizzato lo studio verso i punti deboli che contraddistinguono le applicazioni web su dispositivi mobili, trovando gli strumenti adatti per risolvere il divario tra i due approcci. Per dimostrare la fattibilità delle funzionalità richieste in sede di analisi dei requisiti è stata fornita una panoramica delle API di HTML5, che unite all'utilizzo di JavaScript e diversi framework open source ha portato alla realizzazione teorica dei servizi di cui la Guida Virtuale al museo della musica deve godere. Un'ultima analisi è stata invece dedicata alla Realtà Aumentata, tecnologia che attualmente non è ancora possibile implementare in una applicazione web per via dell'impossibilità di accedere alla fotocamera. Sono stati forniti dettagli relativi alla possibilità di rendere ibrida l'applicazione mediante il framework PhoneGap per poter utilizzare la piattaforma Layar in modo nativo, facendo emergere alcune complessità e rimandando la risoluzione di tale problema a uno studio più approfondito. Ulteriori studi potrebbero essere approfonditi anche per quanto riguarda la funzionalità di salvataggio dei dati dell'applicazione web, valutando i possibili servizi da offrire agli utenti tramite l'uso dei sistemi *localStorage*, *sessionStorage* e del database lato client. Per esempio inserendo la possibilità di salvare mediante un calendario gli eventi del museo di maggiore interesse. Un'altra funzionalità che merita senz'altro uno sviluppo futuro è la modalità offline, per capire di preciso le dimensioni massime di capienza di dati offline che un'applicazione web può tenere in memoria. In questo modo sarebbe interessante determinare se, nel nostro caso, un utente in assenza di connessione ad internet possa mantenere memorizzati all'interno della Guida Virtuale anche le tracce audio e i brani presenti nell'audioguida.

Bibliografia

- [1] Wikipedia *Smartphone* <http://it.wikipedia.org/wiki/Smartphone>
- [2] Wikipedia *Tablet Computer* http://it.wikipedia.org/wiki/Tablet_computer
- [3] Vabsite *Global Mobile Phone Smartphone Sales Statistic & Trends*
<http://www.vabsite.com/2011/09/mobile-users-statistics-usage-trends.html> (02/2012)
- [4] NetMarketShare.com *Mobile/Tablet Operating System Market Share Year 2012* <http://www.netmarketshare.com/operating-system-market-share.aspx?qprid=8&qpcustomd=1> (02/2012)
- [5] *Sviluppare applicazioni per iPhone* di Jonathan Stark. Copyright 2010 Tecniche Nuove, 978-88-481-2506-2
- [6] JQuery <http://jquery.com/>
- [7] JQuery Mobile <http://jquerymobile.com/>
- [8] JQTouch <http://www.jqtouch.com/>
- [9] Sencha Touch <http://www.sencha.com/products/touch>
- [10] The M Project <http://the-m-project.net/>
- [11] PhotoSwipe <http://www.photoswipe.com/>
- [12] *Pro iOS Web Design and Development: HTML5, CSS3, and JavaScript with Safari* di Andrea Picchi. Copyright 2011 Apress, 978-1-4302-3247-6

- [13] JPlayer <http://jplayer.org/>
- [14] PhoneGap <http://phonegap.com/>
- [15] Layar <http://www.layar.com/>

Ringraziamenti

Giunto al termine di questo lungo percorso per il conseguimento della Laurea Triennale in Informatica, volevo ringraziare tutte le persone che hanno reso possibile questo mio traguardo. In particolare volevo ringraziare il professor Fabio Panzieri per avermi dato la possibilità di approfondire con questa tesi il mio interesse verso lo sviluppo di applicazioni per piattaforme mobili iniziata con il tirocinio grazie al professor Danilo Montesi. Ringrazio i suoi consigli e soprattutto la sua disponibilità nell'avermi permesso di conseguire la laurea nei tempi che mi ero prefissato.

Il ringraziamento più grande va ai miei genitori, Paolo e Maddalena, per il sostegno che mi hanno sempre dato nel finire questo percorso e per avermi sopportato nei vari momenti di sconforto, così come mio fratello Luca, sua moglie Roberta e soprattutto il mio nipotino Andrea. I suoi “in bocca al lupo” sono stati fondamentali per ogni mio ultimo esame.

Un grande ringraziamento va verso i miei colleghi universitari che ho conosciuto nei primi anni di università con i quali ho affrontato i primi esami, anche se ormai ci siamo persi un po' di vista, grazie Davide, Gioele, Domenico, Ugo, Mirco, Francesca, Gianluca e Chiara.

Un altro grande ringraziamento lo dedico invece ai miei colleghi di tirocinio, Paolo, Marco e Giacomo, che hanno sopportato le mie lacune e mi hanno supportato nell'approfondire il mio interesse verso lo sviluppo di applicazioni per iPhone e iPad. Grazie, siete stati mitici. Con loro volevo ringraziare anche tutti gli altri colleghi universitari con cui ho studiato negli ultimi due anni e mi hanno aiutato ad affrontare gli esami finali. Grazie ad Alkida, Mat-

teo, Fayad e Ivan per il sostegno che mi avete dato per Sistemi Operativi. È stata dura ma ce l'abbiamo fatta. Un grazie enorme anche a Ruggero per i suoi appunti a me ormai indispensabili. Senza di quelli probabilmente sarei ancora indietro di alcuni esami. Grazie infinite anche a Francesca, Michele, Martino e Davide per avermi fatto sentire parte di un gruppo di persone eccellenti nonostante il divario di età.

Un ringraziamento di cuore è dedicato in particolare ai miei amici di sempre, Marco, Silvia, Jimmy, Pigi, Freddy, Moreno, Baggio, Valentina, Chiara, Bale, Veronica, Matteo, Alessandro, Andrea, PierLuigi, Vittorio, Antonello, Claudio, Marchetto, Federica, Daniele, Alice, Elena e Miriam.

Grazie per aver creduto in me.