

UNIVERSITÀ DEGLI STUDI DI PADOVA
FACOLTÀ DI INGEGNERIA
CORSO DI LAUREA IN INGEGNERIA INFORMATICA

Tesi di laurea

Apple iPad: analisi della piattaforma e sviluppo di un'applicazione

Relatore:
Massimo Rumor

Laureando:
Andrea Mattiuz

Padova, 25 novembre 2010

Indice

Introduzione	ix
1 L'azienda: ADL Ingegneria Informatica	1
I Apple iPad: studio della periferica	3
1.1 Descrizione e funzionamento	7
2 I competitors	15
2.1 HP slate	16
2.2 Microsoft Courier	17
2.3 Archos 9 Tablet PC	18
2.4 Notion Ink Adam	19
2.5 Fusion Garage Joojoo	20
2.6 ExoPC Slate	21
2.7 ICD Gemini	21
2.8 Samsung Galaxy Tab	22
2.9 Toshiba Folio 100	23
2.10 Blackberry Playbook	24
3 Il sistema operativo	27
3.1 La critica più grande: il lancio senza multitasking	29
3.2 Le novità del sistema operativo 4.2	31
3.3 Applicazioni preinstallate	31
4 Lo sviluppo di applicazioni	35
5 L'ambiente di sviluppo Apple	39
5.1 Xcode	39
5.1.1 Xcode: il debugger	41
5.2 Interface Builder	42

5.3	Strumenti	43
5.4	Clausole per lo sviluppo software	44
6	Il linguaggio Objective C	47
6.1	Le classi	47
6.2	I metodi	49
6.3	Le variabili di istanza	51
6.4	Protocolli e delegati	51
6.5	Le proprietà	52
6.6	Il costruttore	52
6.7	Super e self	53
6.8	Override ed overload	53
6.9	L'oggetto ID	54
6.10	Gestione della memoria	54
II	Sviluppo di un'applicazione gestionale: iOrder	55
7	Raccolta ed analisi dei requisiti	59
7.1	Specifiche di massima	59
7.2	Analisi in dettaglio: requisiti dell'applicazione	59
8	Progettazione della base di dati	65
8.1	Progettazione concettuale	65
8.1.1	Dizionario dei dati	67
8.1.2	Regole di vincolo	67
8.1.3	Regole di derivazione	67
8.2	Progettazione logica	69
8.2.1	Volumi dei dati	69
8.2.2	Analisi delle ridondanze	70
9	Progettazione dell'interfaccia utente	73
10	Sviluppo del progetto	77
10.1	Implementazione della base di dati	77
10.2	Implementazione del codice	79
10.3	Gestione dei trasferimenti	96
	Conclusioni e sviluppi futuri	99
	Bibliografia	101

Elenco delle figure

1	Evoluzione della disponibilità di applicazioni nell'Apple App Store nel periodo Maggio 2008 – Agosto 2010	x
1.1	Apple iPad	5
1.2	Vendite dispositivi portatili Apple al 6 settembre 2010	6
1.3	Apple iPad: panoramica	8
1.4	Apple iPad: tasti	9
1.5	iBooks	10
1.6	I programmi della suite iWork: Keynote, Numbers e Pages	11
1.7	Esempio di applicazione medica	12
2.1	Calo della crescita nel mercato dei netbook negli USA	16
2.2	HP Slate	17
2.3	Microsoft Courier	18
2.4	Archos 9 Tablet PC	18
2.5	Notion Ink Adam	19
2.6	Fusion Garage Joojoo	20
2.7	ExoPC Slate	21
2.8	ICD Gemini	22
2.9	Samsung Galaxy Tab	23
2.10	Toshiba Folio 100	24
2.11	Blackberry Playbook	24
3.1	Gli strati del sistema operativo iOS	27
3.2	Calendario	32
3.3	Immagini in modalità “luoghi”	34
4.1	SDK Apple	35
5.1	Finestra del progetto in Xcode	40
5.2	L'esecuzione di un progetto da Xcode	40
5.3	Il debugger di Xcode	42

5.4	Interface Builder	43
5.5	Strumenti	44
6.1	Esempio di interface	48
6.2	Esempio di implementation	49
6.3	Esempio di metodo	49
7.1	Diagramma di sequenza	60
7.2	Diagramma dei casi d'uso	61
7.3	Diagramma delle attività (con swimlane)	62
8.1	Schema concettuale Entità-Relazione	66
8.2	Schema logico	71
9.1	Split view	74
9.2	Alert view	74
9.3	Barra di navigazione	75
9.4	Barra di ricerca con scope bar	76
9.5	DatePicker	76
10.1	SQLite Manager	78
10.2	Architettura dell'applicazione iOrder	80
10.3	Vista clienti	82
10.4	Vista dettaglio cliente	83
10.5	Alert view nuovo ordine	84
10.6	Vista inserimento nuovo cliente	85
10.7	Vista pagamenti	86
10.8	Struttura riga nella tabella degli articoli	86
10.9	Ricerca degli articoli	87
10.10	Vista nuova riga d'ordine	88
10.11	Vista ordine in corso	89
10.12	Struttura riga d'ordine	89
10.13	Cancellazione righe d'ordine	90
10.14	Invio e-mail di conferma	91
10.15	Vista storico ordini	93
10.16	Vista ordine dello storico	94
10.17	Vista impostazioni	95
10.18	Vista sincronizza	96

Elenco delle tabelle

7.1	Lista delle operazioni con frequenza	64
8.1	Lista delle entità	68
8.2	Lista delle associazioni	69
8.3	Volumi dei dati	70

Introduzione

Premessa

L'INFORMATICA È SOGGETTA ad una continua evoluzione, ed una tecnologia che sembra innovativa può rivelarsi ben presto obsoleta o non prendere piede nel mondo degli addetti ai lavori. Osservando l'andamento del mercato tecnologico degli ultimi tempi è emerso come le persone stiano orientando sempre di più il loro interesse verso il mondo dei dispositivi mobili: telefoni cellulari, lettori mp3 e computer portatili permettono agli utilizzatori (più o meno esperti) di comunicare, ascoltare musica e navigare in internet in ogni parte del mondo.

L'evoluzione del marketing propende in modo sempre più marcato verso tale direzione; inizialmente il mondo mobile era abbastanza trascurato e di secondaria importanza, mentre oggi è esploso come una delle fonti principali di successo (e di ricavi) per le aziende che operano in questo settore. Di questi tempi, nei giorni in cui i social network spopolano, la tecnologia “on the move” è essenziale per l'utente che desidera interagire in qualunque momento.

Quando si parla di tecnologia e dispositivi portatili di ultima generazione, una delle prime aziende che vengono in mente è senza dubbio Apple e infatti, come indicato anche da un recente studio dell'autorevole rivista di economia e finanza Forbes, la casa della mela morsicata, grazie a prodotti come iPhone ed iPod, è il marchio più famoso al mondo. Questo anche se (o forse proprio perchè) l'ambiente Apple è sempre stato circondato da un alone di mistero per le tecnologie utilizzate e per le scelte progettuali rivoluzionarie molte volte criticate per essere chiuse, controcorrente e contrarie alle logiche di mercato, ma che poi si rivelano fonte di grandi successi.

A dimostrazione di questo il fatto che è sempre più frequente, per la gente comune, confondere la categoria di un dispositivo portatile con il nome che questa azienda ha dato alla sua periferica: ad esempio quante volte si sente parlare di iPod anche se il lettore mp3 in questione è di tutt'altra marca?

Altro evento che sta rivoluzionando il pianeta informatico è il progressivo abbandono dell'acquisizione di software e materiale multimediale per via tradizionale a favore degli app store presenti su internet.

In questo campo Apple è considerata il precursore con il lancio dell'iTunes Music Store prima e dell'App Store poi, due negozi on-line che con il passare del tempo si sono dimostrati essere delle galline dalle uova d'oro, grazie anche alla progressiva diminuzione dei prezzi ed all'aumento del materiale disponibile che, nel secondo dei due casi, permette a chiunque di pubblicare e vendere applicazioni con una vantaggiosa quota di ricavo.

In tale contesto ha fatto da poco la sua comparsa un dispositivo di nuova generazione, una via di mezzo tra un computer ed una console portatile: un concentrato di tutte quelle periferiche mobile che catturano maggiormente l'attenzione dei fruitori multimediali dei giorni nostri. Un dispositivo che tra mille polemiche e un rincorrersi di voci sta premiando l'arguzia e la propensione al rischio di Apple e del suo amministratore delegato Steve Jobs: iPad.

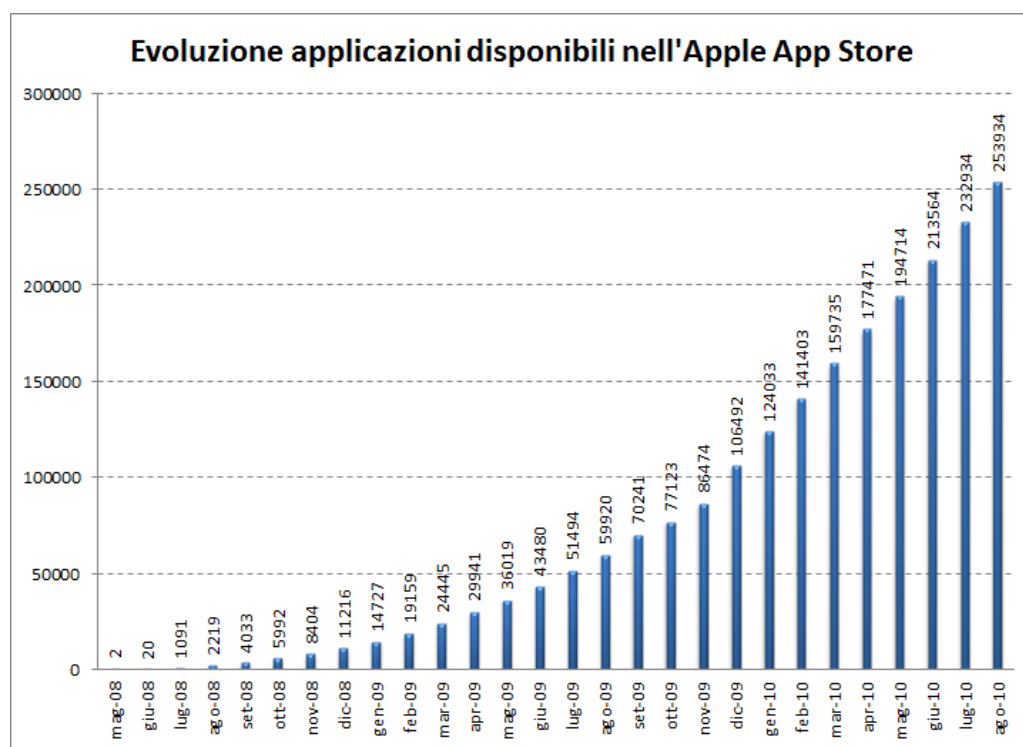


Figura 1: Evoluzione della disponibilità di applicazioni nell'Apple App Store nel periodo Maggio 2008 – Agosto 2010

Obiettivi

L'interessamento verso un mondo in così rapida e continua espansione ed al tempo stesso poco conosciuto come il pianeta Macintosh ha permesso, prendendo al volo l'occasione del lancio sul mercato dell'innovativo dispositivo Apple iPad e dall'opportunità offerta dall'azienda ADL Ingegneria Informatica di Conegliano Veneto (TV), di incentrare il lavoro da svolgersi durante il tirocinio sullo studio dell'innovativa periferica portatile. Sono stati affrontati un'analisi del suo mondo di applicazione, dell'utilità ed dei concorrenti sul mercato, insieme allo sviluppo di un applicativo gestionale mobile che permettesse di sfruttare le caratteristiche della piattaforma e di conoscere un linguaggio di programmazione poco conosciuto ma che sta salendo alla ribalta proprio perché usato per lo sviluppo di software da utilizzare su dispositivi di questa azienda.

Risorse

Durante il periodo trascorso all'interno dell'azienda sono state gentilmente rese disponibili varie risorse con le quali è stato possibile svolgere il lavoro previsto.

- Computer desktop con sistema operativo Microsoft Windows XP: utilizzato per svolgere l'attività di ricerca informazioni e studio riguardanti il mondo Apple ed in particolare il dispositivo iPad, le sue caratteristiche e funzioni peculiari ed i suoi competitors sul mercato;
- Computer desktop Apple iMac dotato di sistema operativo Snow Leopard 10.6.4: indispensabile per sviluppare l'applicativo gestionale e testarlo grazie ai programmi resi disponibili da Apple tra cui l'ambiente di sviluppo dedicato ed il simulatore per iPad e iPhone;
- Stampanti laser in bianco e nero ed a colori di ultima generazione per una stampa rapida e di qualità in caso di necessità;
- Collegamento internet all'interno di rete lan aziendale;

Capitolo 1

L'azienda: ADL Ingegneria Informatica

ADL INGEGNERIA INFORMATICA è un'azienda presente a Conegliano Veneto (TV) dal 1995, opera nel settore dell'innovazione tecnologica ed è specializzata nell'analisi, integrazione ed implementazione di sistemi informatici aziendali.

L'obiettivo dell'azienda è offrire prodotti di alta qualità e completi sotto ogni punto di vista (grafico, informatico, tecnologico, ecc. . .), per i quali la progettazione e la realizzazione richiedono competenze interdisciplinari: ADL si avvale quindi di professionisti, del settore e non, e di alcune importanti cooperazioni con società e studi esterni.

La società crede nel concetto di formazione continua del personale, tutti i suoi collaboratori infatti dispongono di titoli di studio adeguati e certificazioni europee, e si aggiornano seguendo corsi e seminari di alto livello; la ditta, inoltre, partecipa costantemente a fiere, eventi e congressi, nazionali ed internazionali, per restare sempre al passo coi tempi e con le tecnologie di ultima generazione.

Le competenze professionali e la struttura di ADL permettono di fornire al cliente, dal professionista all'impresa di medie e grandi dimensioni, un servizio di supporto e consulenza completo a 360 gradi per tutte le componenti del sistema informatico (hardware, software ed infrastrutture eventualmente necessarie come LAN, WAN, sistemi di lettura bar code, sistemi di identificazione con trasponder, sistemi di stampa industriali, sistemi di decodifica, Palm Pc, ecc. . .).

L'azienda si compone infatti di quattro divisioni principali, grazie alle quali può proporsi ai propri clienti come global partner (unico partner), e queste sono:

- divisione LAN, internetworking e security;
- divisione software engineering;
- divisione sistemi (prodotti software e hardware commerciali);

- divisione eMemo (Internet solution provider).

Uno dei punti di forza di ADL Ingegneria Informatica è costituito dallo sviluppo di CMS (Content Management System), i quali facilitano la gestione da parte dell'utente di alcune aree del proprio sito web, permettendo l'inserimento e la modifica dei contenuti con pochi e semplici click del mouse.

Tutto ciò produce una sostanziale diminuzione dei costi di mantenimento e supporto tecnico per il cliente, non necessitando di competenze specifiche e complessi programmi per effettuare delle semplici variazioni al proprio prodotto.

ADL Ingegneria Informatica è un'azienda certificata dal 1999 ISO 9001:2008, progettazione inclusa, con l'ente Accreditato di certificazione SGS: è richiesta una verifica annuale della propria organizzazione e struttura operativa per garantire una sempre maggiore qualità dei servizi proposti ed un'attenzione continua alle esigenze dei propri clienti.

L'azienda ha inoltre altre importanti certificazioni:

- Acronis Certified Engineer
- AT Allied Telesyn Certified Partner
- Cisco Reseller Partner
- HP Business Partner
- IPI Intel Product Integrator
- Microsoft Certified Professional
- Microsoft Small Business Specialist
- Network Instruments Observer Certified
- PANDUIT System Integrator
- WatchGuard Certified System Professional
- Zyxel Certified Partner

Parte I

Apple iPad: studio della periferica

Introduzione del dispositivo

LA DATA IN CUI APPLE ha presentato il suo nuovo dispositivo portatile, iPad, ha generato in tutto il mondo un rincorrersi frenetico di voci e notizie su di esso e sul nuovo segmento di mercato in cui si inserisce; questa spirale di pubblicità non ha fatto altro che alimentare il suo successo, e probabilmente continuerà a farlo.

Il 27 gennaio 2010 l'amministratore delegato dell'azienda, Steve Jobs ha introdotto la periferica spiegando che ormai tutti utilizzano sistemi portatili e smartphone, ma anche che queste due categorie di prodotti lasciano spazio per qualche cosa nel mezzo, una terza via che sia in grado di unirli, un anello mancante. Per creare la nuova categoria è necessario però che questa sia migliore di entrambe le altre: deve essere allo stesso tempo migliore di uno smartphone e di un notebook.



Figura 1.1: Apple iPad

Una delle critiche che da subito è stata mossa all'iPad è il fatto di non essere portatile come il primo e, al contempo, non avere le performance paragonabili a quelle del secondo, considerando però il rovescio della medaglia è più portatile di un notebook e più performante di uno smartphone. Come Jobs ha detto alla presentazione, è stato pensato per un uso abbastanza specifico: navigare sul web, leggere le email, guardare foto, video e film e leggere libri e riviste. Questo non esclude usi diversi, infatti questo tablet può diventare senza problemi un'ottima console portatile grazie ai giochi che verranno sviluppati, o uno strumento di produttività grazie alla suite iWork. Per non parlare delle migliaia di applicazioni che verranno sviluppate e che porteranno ad un utilizzo dell'iPad che magari adesso non è possibile immaginare (oltre alle 250.000 e più già disponibili sull'App Store).

Il giorno del suo lancio negli USA, il 4 aprile 2010, iPad ha venduto 300.000 unità ed in meno di un mese è riuscito a varcare la soglia del milione, superando perfino il successo

del lancio dell'iPhone che tre anni prima impiegò oltre 70 giorni per raggiungere lo stesso risultato. Si stima che Apple potrà superare entro fine anno i nove milioni di dispositivi venduti; e questi dati sono in continua rivisitazione al rialzo. Due società di investimenti americane (UBS Investment Research e Barclays Capital) stimano che le vendite di iPad nel prossimo anno saranno rispettivamente di 28 o 20 milioni di unità.

Al momento i dati certi divulgati da Apple indicano le vendite di iPad del terzo trimestre 2010 a quota 3,37 milioni, e, al 6 settembre di quest'anno, l'azienda ha dichiarato di aver distribuito 7,2 milioni di tali dispositivi.

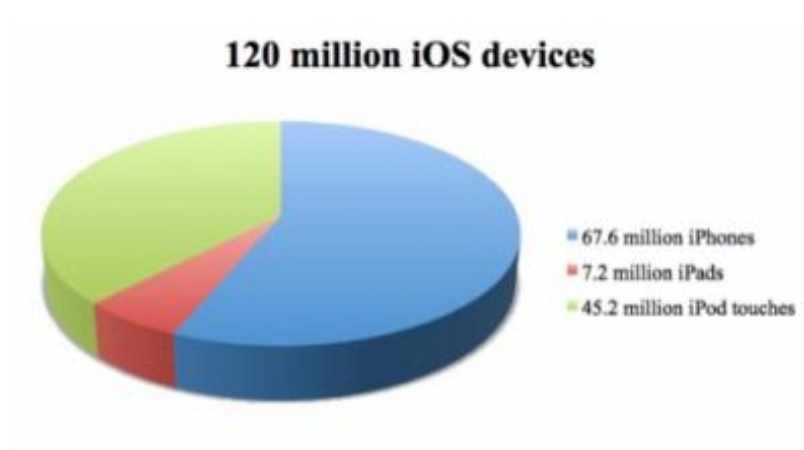


Figura 1.2: Vendite dispositivi portatili Apple al 6 settembre 2010

Per quel che riguarda il mercato italiano il tablet è stato reso disponibile in pre-ordine il 10 maggio e alla data del lancio, il 28 dello stesso mese, erano state già prenotate 125.000 unità.

Al momento attuale si trovano in commercio sei diversi modelli di iPad, tre dei quali con il solo collegamento WiFi e altri tre con la possibilità di collegarsi anche in 3G.

Citando la dichiarazione di Steve Jobs:

“A dar retta agli esperti, lo avremmo dovuto proporre a meno di 1000 dollari, che è il linguaggio in codice per dire 999 dollari. Quando abbiamo deciso di svilupparlo, abbiamo fissato ambiziosi obiettivi tecnologici, ma anche aggressivi obiettivi di prezzo. E proprio come siamo stati capaci di rispondere od eccedere gli obiettivi tecnologici, abbiamo risposto anche agli obiettivi di costi...il prezzo di iPad parte da 499 dollari”.

I prezzi restano esattamente come quelli applicati negli Stati Uniti, e non tengono conto del cambio Euro - Dollaro (spese di spedizione e IVA comprese):

- Apple iPad 16GB Wi-Fi: 499 Euro;
- Apple iPad 32GB Wi-Fi: 599 Euro;
- Apple iPad 64GB Wi-Fi: 699 Euro;
- Apple iPad 16GB Wi-Fi + 3G: 599 Euro;
- Apple iPad 32GB Wi-Fi + 3G: 699 Euro;
- Apple iPad 64GB Wi-Fi + 3G: 799 Euro.

1.1 Descrizione e funzionamento

L'azienda di Cupertino ha ripreso anche per iPad le linee minimali dei suoi prodotti precedenti, infatti, nella cornice in alluminio satinato resa più spessa solo da un bordo nero creato sul vetro di protezione frontale, è presente un unico tasto centrale, mentre il retro del dispositivo mostra quello che Apple indica come chassis unibody: un singolo pezzo di alluminio senza giunture o viti fa da case per la periferica rendendola dunque molto solida e priva di parti che potrebbero vibrare o far rumore.

Fra tutti i componenti necessari a realizzare un sistema completo come iPad, lo schermo riveste sicuramente la maggiore importanza. Esso serve non solo per visualizzare informazioni ma anche come unica periferica di input e dunque di interazione con il dispositivo stesso; il principale metodo per l'immissione del testo è una tastiera virtuale che quando attivata occupa circa metà display con tradizionale layout QWERTY.

I giudizi sull'efficacia di questa funzionalità sono contrastanti, soprattutto nel caso di lunghe digitazioni: in caso di necessità è possibile avvalersi di una tastiera reale da collegare al dispositivo e disponibile come accessorio.

Anche la scelta di utilizzare un display multitouch di tipo capacitivo e retroilluminato a LED è stata contestata da molti, dato che questi tipi di schermi possono affaticare la vista dopo poche ore di utilizzo, cosa effettivamente non positiva per il fine di questa creazione. Fortunatamente Apple ha pensato anche a questo: l'LCD montato su iPad usa la tecnologia IPS, che non solo garantisce una qualità superiore dell'immagine e consumi più bassi, ma anche una migliore risposta all'angolo di visualizzazione ed un affaticamento minore della vista anche dopo ore d'uso, come tra l'altro confermato da alcuni medici che hanno avuto modo di provare l'iPad.

La diagonale dello schermo, dalla quale dipende poi il form factor dell'intero prodotto, è di 9,7 pollici e la risoluzione nativa è di 1024*768 pixel, con una definizione di 132 pixel



Figura 1.3: Apple iPad: panoramica

per pollice; per definizione dell'immagine, intensità del nero, qualità dei colori, lo schermo è paragonabile a quello dei migliori portatili di fascia alta.

Le dimensioni complessive misurano 189,7 millimetri di larghezza, 242,8 di lunghezza e 13,4 di spessore mentre il peso si aggira sui 700 grammi, a seconda della versione con solo WiFi o con anche connettività 3G.

L'interfaccia è prevalentemente basata sulla tecnologia touch, fanno eccezione solamente il tasto home alla base del dispositivo, e lungo i bordi della cornice in alluminio quello di accensione/spegnimento, oltre ai tasti di regolazione del volume e del muto sul profilo destro; iPad è infine dotato di connettore dock per le operazioni di sincronizzazione al PC o al Mac via adattatore USB (che avvengono tramite bus USB, come per iPhone o iPod), di microfono e altoparlante integrati e, nelle versioni 3G, di un apposito cassetto per accogliere le micro SIM GSM.



Figura 1.4: Apple iPad: tasti

Rilevanti per un dispositivo di questo genere sono le mancanze all'appello di una porta USB e di uno slot per l'espansione della memoria, seppure anche in questo caso siano disponibili degli appositi accessori, oltre all'assenza di una videocamera integrata.

Sul fronte tecnologico il nuovo iPad è basato su un inedito system-on-a-chip proprietario Apple A4 che è il primo processore creato direttamente dall'azienda: si tratta di un processore ARM da 1Ghz, sviluppato in modo tale da ridurre i consumi, abbassando in alcuni casi la velocità di clock e spegnendo parti del processore stesso quando non vengono utilizzate dal sistema.

Il tablet Apple si compone inoltre di accelerometri, sensori di luminosità ambientale, bussola digitale e modulo di geolocalizzazione satellitare con funzionalità Assisted-GPS.

Come accennato in precedenza, sono presenti anche connettività WiFi e modulo 3G (l'ultimo solo in alcune versioni e abilitato unicamente per la trasmissione dati) ed infine Bluetooth 2.1 con tecnologia Enhanced Data Rate.

iPad è stato realizzato ponendo attenzione al rispetto dell'ambiente, con l'impiego di un display privo di arsenico e di mercurio per la costruzione di vetro ed LCD, mentre l'intera struttura è costituita da alluminio e vetro riciclabili.

Una particolarità del dispositivo riguarda l'utilizzo delle Micro-SIM nel modello 3G. Questo infatti consente di navigare su internet non per mezzo di una classica SIM che tutti gli operatori attualmente forniscono, ma di un modello più piccolo chiamato appunto Micro-SIM. Al momento del lancio né negli USA né in Europa nessuno adoperava tale formato: tutti gli operatori che intendevano commercializzare l'iPad hanno dovuto adattarsi e creare delle schede che potessero essere inserite nell'apposito vano. Probabilmente con questa scelta Apple ha voluto fare in modo che le Micro-SIM abbinate all'acquisto di un iPad tramite abbonamento non venissero impiegate anche su altri dispositivi e che quindi le offerte dei vari operatori vengano utilizzate solo su questa periferica.



Figura 1.5: iBooks

Per quanto riguarda gli ebook Apple ha presentato l'iBookStore, destinato ad essere quello che per le applicazioni è stato e continua ad essere l'App Store: un'enorme vetrina di libri tramite la quale l'utente può trovare praticamente di tutto, leggere alcune pagine insieme alle info su autore e trama ed acquistare con un click. Immediatamente il libro digitale viene scaricato ed è pronto per essere letto mediante l'applicazione gratuita iBooks, che simula anche lo sfogliarsi delle pagine. L'utente ha la possibilità, per ciascun libro, di visualizzare il font che preferisce e di modificare la dimensione del carattere a seconda delle

proprie esigenze. Le opere, a discrezione dell'editore, possono essere in bianco e nero, a colori e anche contenere video. Per gli ebook commercializzati su iBooks Apple usa il formato ePub: uno standard open source specifico e basato su xml dinamico, già diffuso, semplice da utilizzare e formattare. I libri digitali sono venduti a prezzi variabili in base all'importanza ed alla data di pubblicazione. Oltre ai classici libri è possibile anche acquistare singole copie di quotidiani e riviste od abbonarsi ad esse, proprio come si fa con le copie cartacee. Tra le testate che hanno già ufficializzato il loro ingresso su iPad ci sono il New York Times, il Wall Street Journal e numerosi editori di fumetti; ma anche in Italia Repubblica, Corriere della Sera, La Stampa, La Gazzetta dello Sport e molte altri titoli si possono già consultare in formato digitale.

Altre funzionalità di questo nuovo tablet sono la gestione delle e-mail, grazie ad un client specificatamente progettato per adattarsi alle caratteristiche del dispositivo, la visualizzazione di album e fotografie che, grazie ad un apposito accessorio, è possibile importare direttamente da una macchina fotografica digitale, oltre a rubrica indirizzi, calendari ed agende, ed ancora Google Maps, apposite versioni di iTunes Store e di App Store; non poteva mancare inoltre la gestione della musica effettuata attraverso le conosciute funzioni iPod.

La navigazione web è affidata al browser Safari che si occupa anche di effettuare un rendering perfetto delle pagine adattandole alla risoluzione del dispositivo. Viste le diatribe fra Apple ed Adobe, l'iPad non supporta i siti web ed i componenti (banner pubblicitari, applicazioni e giochi) realizzati in Flash, che dunque non possono essere visualizzati: questa è una delle critiche più feroci che hanno accompagnato l'uscita di questa periferica. Lo standard supportato è invece HTML5, la nuova frontiera del web che, pian piano, tutti i browser cominciano a supportare. Le novità introdotte dalla nuova versione del linguaggio di markup sono numerose, una delle più interessanti è senz'altro la possibilità di riprodurre video in maniera nativa (e quindi senza l'ausilio di plugin aggiuntivi) sui browser.



Figura 1.6: I programmi della suite iWork: Keynote, Numbers e Pages

Un'altra novità è la suite iWork, dove Keynote, Numbers e Pages (per presentazioni, documenti di testo e i fogli di calcolo) sono state opportunamente riscritte ed adattate per sfruttare le caratteristiche di questo nuovo dispositivo: non solo uno strumento di svago e per il tempo libero quindi, ma anche un ausilio per la produttività personale. Le applicazioni di iWork sono disponibili su App Store al prezzo di 9,99 dollari ciascuna.

Per quanto concerne i software di terze parti l'ultimo dispositivo della Mela è in grado di eseguire, nativamente, qualunque applicazione già creata per iPhone o iPod touch in due modalità: in risoluzione originale (ed in questo caso verrà visualizzata la piccola applicazione nel mezzo dello schermo) oppure a tutto schermo, con qualche inevitabile artefatto grafico. Con il rilascio del nuovo SDK però gli sviluppatori hanno la possibilità di modificare oppure scrivere nuove applicazioni appositamente dedicate.

Un punto focale per il giudizio di un dispositivo di questo genere è sicuramente la durata della batteria, ed Apple indica un'autonomia di dieci ore per navigare sul web, ascoltare musica o guardare video. Jobs ha sottolineato come l'obiettivo sia quello di permettere ad iPad di poter sostenere una piena operatività, anche multimediale, per un viaggio aereo intercontinentale. Il tempo di standby viene invece indicato essere di oltre un mese; un ottimo risultato, tenendo presente le dimensioni dello schermo e l'hardware interno.



Figura 1.7: Esempio di applicazione medica

In ambito lavorativo l'iPad si presta a diverse implementazioni, ad esempio alcuni piloti americani ritengono che il nuovo dispositivo Apple, grazie al grande e luminoso display, sarebbe perfetto per un suo utilizzo in cabina di pilotaggio, magari con applicazioni specifiche che aiutino i piloti durante il volo, sfruttando GPS e connessione 3G. Anche alcune

associazioni mediche negli Stati Uniti stanno guardando con molto interesse al nuovo tablet, che consentirà di avere sotto mano tutta una serie di informazioni sui pazienti e le loro cartelle cliniche (corredate di immagini, testo e video).

Chi deciderà di acquistare iPad troverà nella confezione solo alcuni accessori essenziali, come da tradizione Apple: alimentatore da rete e cavo USB, un pieghevole cartaceo con qualche spiegazione sul funzionamento ed una serie di adesivi. Come detto in precedenza sono però già previsti alcuni accessori espressamente dedicati tra cui un dock provvisto di tastiera ed una particolare custodia che, opportunamente piegata, può fungere come supporto da tavolo al dispositivo.

Capitolo 2

I competitors

DA QUANDO È STATA ANNUNCIATA l'uscita di iPad, tutte le grandi aziende di tecnologia si sono affrettate a presentare oppure ad alimentare le indiscrezioni su prodotti che puntano ad essere i diretti avversari del dispositivo di casa Apple. Al momento però, trasmentite, cancellazione di progetti e pubblicità diretta ed indiretta di tale periferica, questo risulta pressoché l'unico conosciuto dalla gente. I dati sul numero di iPad prenotati e venduti prima negli USA e poi in Europa evidenziano come la strada imboccata da questo tablet sia quella di un grande successo, a maggior ragione dopo che due produttori del calibro di HP e Microsoft hanno deciso di annullare, almeno per il momento, i rispettivi progetti.

Secondo quanto rilevato da Morgan Stanley il 44% degli acquirenti di iPad si è dotata di uno dei tablet di Apple per sostituire il proprio netbook, oppure l'ha comprato perché voleva un ultraportatile preferendolo a tutte le soluzioni dotate di tastiera. Qualcuno ha dichiarato di essersi deciso ad acquistare un iPad per sostituire il proprio notebook, sorvolando sulla mancanza di Flash, di porte Usb e sulla potenza inferiore rispetto a quella di un portatile tradizionale.

Il 28% degli intervistati ha invece dichiarato che, avendo deciso di acquistare un e-reader, ha preferito orientarsi verso l'offerta di Apple anziché verso quelli che potremmo considerare e-reader puri, proposti dai concorrenti. Il 27%, infine, si è spinto a comprare un iPad in sostituzione di un desktop.

Questa stessa ricerca rileva per il mercato dei netbook negli Usa un calo della crescita dal +641% del luglio dello scorso anno al 5% di aprile. Secondo la società di analisi è l'iPad che comincia a far sentire i suoi effetti. Il picco raggiunto lo scorso agosto con un eccezionale +641% rispetto al 2008 è solo un ricordo da tempo visto che la curva delle vendite da allora è in progressivo appiattimento, ma da gennaio, ovvero dall'annuncio della periferica Apple, la flessione è continuata riducendo sensibilmente la percentuale di crescita mese dopo mese: dicembre +179%, gennaio +68%, febbraio, +53%, marzo +25%, aprile +5%.

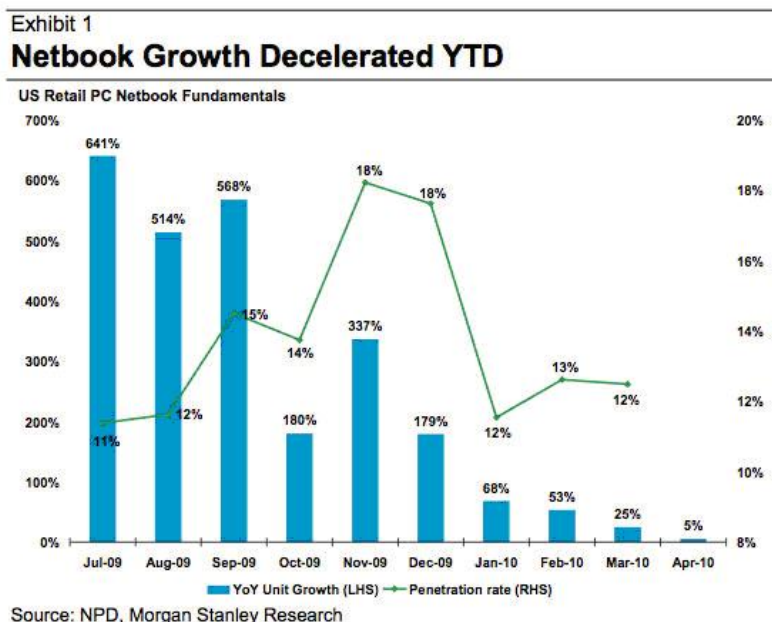


Figura 2.1: Calo della crescita nel mercato dei netbook negli USA

Certo il dispositivo Apple non sarà l'unico colpevole, ma sicuramente è uno dei maggiori indiziati.

Ecco presentata un'analisi dei principali dispositivi che dovrebbero (o avrebbero dovuto) dividersi il mercato con l'iPad. (sono esclusi quei prodotti che al momento non presentano delle caratteristiche ben definite)

2.1 HP slate

Il dispositivo HP Slate era atteso (insieme al Microsoft Courier, vedi dopo) come l'avversario per eccellenza dell'iPad Apple, per le promesse e le voci che si rincorrevano su internet.

Invece indiscrezioni sostengono che l'azienda non fosse soddisfatta di Windows 7 come sistema operativo che avrebbe dovuto equipaggiare il tablet e per questo abbia terminato il progetto, almeno con tali caratteristiche. Stando alle notizie, HP potrebbe abbandonare anche l'hardware Intel, perché non in grado di assicurare un'autonomia della batteria all'altezza delle previsioni della casa produttrice. Molto probabilmente questa decisione è frutto anche di un cambio di strategia da parte di Hp dopo la recente acquisizione di Palm e di webOS, sistema operativo decisamente più portato per il mondo mobile.

Si profila quindi un ritardo a data da destinarsi nel lancio di un prodotto HP in questo mercato.



Figura 2.2: HP Slate

Ad ogni modo le caratteristiche tecniche trapelate in precedenza parlavano di un dispositivo con display touch screen multitouch di 8,9" con risoluzione di 1024*600 pixel, fornito di accelerometro, sensore di luminosità e supporto per il pennino. Il processore designato era un Intel Atom Z530, la memoria RAM di 1 GB e per l'archiviazione dati era disponibile la scelta tra due unità SSD da 32 e 64 GB con uno slot per schede di memoria SD/SDHC. Per le connessioni erano disponibili i supporti per Wi-Fi (802.11b/g), Bluetooth, 3G (opzionale), GPS, e USB 2.0. L'HP Slate avrebbe dovuto avere anche una fotocamera da 3 Megapixel ed una Webcam VGA, il tutto gestito dal sistema operativo Windows 7 Home Premium, per una durata della batteria che avrebbe garantito all'incirca 5 ore di operatività.

2.2 Microsoft Courier

Anche il Microsoft Courier, come l'HP Slate citato in precedenza, era nato come l'anti iPad, ma anch'esso si è trasformato in un nulla di fatto, questa volta in modo definitivo.

Frank Shaw, vicepresidente del dipartimento Comunicazione in Microsoft, ha annunciato la chiusura del progetto relativo al tablet in questi termini:

“Nel DNA di Microsoft vi è la volontà e la necessità di sviluppare e conservare continuamente nuove tecnologie per incoraggiare la produttività e la creatività. Il progetto Courier è un esempio di questo tipo di sforzo e le sue tecnologie potranno valutate in futuro e magari impiegate in altri prodotti Microsoft. Al momento però l'azienda di Redmond non ha alcuna intenzione di terminare e commercializzare questo tipo di prodotto”.

Secondo le fonti il Courier era stato pensato come un “giornale digitale” composto da due facciate multi tocco da 7 pollici di diagonale; un dispositivo sottile circa 2,5 cm, con

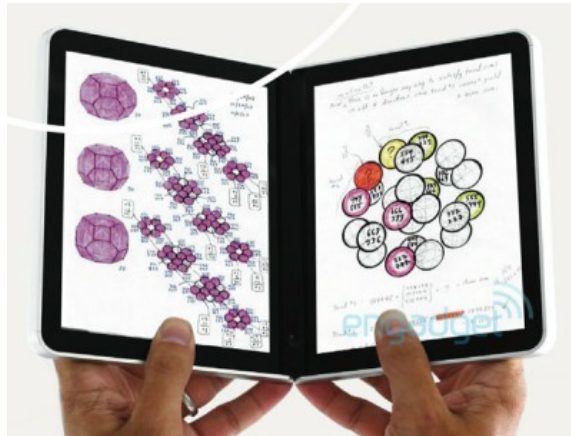


Figura 2.3: Microsoft Courier

un peso di poco meno di 500 gr che chiuso, sarebbe stato anche molto piccolo, circa 13x18 cm, la misura standard delle stampe fotografiche. La piattaforma hardware sarebbe stata dotata del SoC NVidia Tegra 2, e probabilmente avrebbe contato su un sistema operativo basato su Windows CE, condividendo alcuni elementi con quello dello Zune HD e quello degli ultimi smartphone Windows. L'interfaccia utente, a quanto pare, avrebbe contato sul pennino più che sulle dita, enfatizzando le operazioni di scrittura. Altre caratteristiche sarebbero dovute essere la connettività Wi-Fi ed una fotocamera da 3 Mega-pixel.

2.3 Archos 9 Tablet PC

Archos 9 Tablet PC è il primo dispositivo simil-iPad in analisi già disponibile sul mercato.



Figura 2.4: Archos 9 Tablet PC

Questa periferica (che misura 134*56*16 mm per 800 grammi di peso) è configurata con un processore Intel Atom Z515 a 1.2GHz con 1GB di memoria ed un hard disk da 1.8 pollici con capacità di 60GB (che dovrebbe incidere negativamente sulle prestazioni totali del sistema). La componente grafica controlla uno schermo da 8.9 pollici touchscreen con risoluzione di 1024*600 pixel. La dotazione è completata da una webcam da 1.3 megapixel con microfono, altoparlanti, un sintonizzatore TV e connettività Wi-Fi, Bluetooth, Ethernet con la disponibilità di porte USB 2.0. Secondo i recensori che l'hanno testato, l'Archos 9 avrebbe problemi di sensibilità del touch-screen che sembra alla volte non rispondere proprio al tocco. La dissipazione di calore è un altro punto critico del tablet che dopo un pò di tempo risulterebbe sgradevolmente caldo, insieme alla durata della batteria che garantisce solo 4 ore di autonomia. La periferica è dotata di sistema operativo Windows 7 Starter Edition, che però manca di alcune funzionalità touch presenti nella versione Home Premium, ed è disponibile ad un prezzo di 499,99 euro.

2.4 Notion Ink Adam

L'azienda indiana Notion ink è un'altra realtà pronta ad offrire il proprio prodotto e ad entrare a breve in questo nuovo mercato in rapida evoluzione.



Figura 2.5: Notion Ink Adam

Adam è un tablet di dimensioni 239*158*13 mm per un peso di circa 600-650 grammi dotato di uno schermo multi tocco da 10,1 pollici, in grado di lavorare ad una risoluzione nativa pari a 1.024*600 pixel. Caratteristica molto importante di questo componente è la tecnologia Pixel Qi in grado di lavorare a basso consumo di energia se utilizzato in ePaper mode, modalità in bianco e nero con retroilluminazione disattivata utile a risparmiare energia; questa funzionalità è prevista per mostrare gli e-book riflettendo la luce come un foglio di carta. Il dispositivo è basato sulla piattaforma SoC nVidia Tegra 2 con processore a 2.1 GHz e 512 Mb di RAM e sfrutta una memoria flash da 16 GB o 32 GB; dispone inoltre di uno slot per l'inserimento di schede di memoria di tipo SD/SDHC. Per quel che

riguarda le comunicazioni, oltre ad un modulo Bluetooth integrato, prevede anche la connettività Wi-Fi per la connessione a reti wireless e la possibilità di installare un modulo UMTS/HSDPA. Adam è dotato di una porta USB, una webcam integrata con rotazione 180 gradi e sensore da 3 megapixel, microfono, funzionalità A-GPS ed accelerometro. Secondo quanto dichiarato, la periferica dovrebbe raggiungere un'autonomia fino ad 8 ore ed il sistema operativo designato è Android di Google.

Notion Ink Adam dovrebbe essere disponibile ad un prezzo di circa 320 dollari.

2.5 Fusion Garage Joojoo

Joojoo è una periferica tablet prodotta a Singapore dall'azienda Fusion Garage, già lanciata nel mercato USA pochi giorni prima di Apple iPad e ad inizio Maggio nel resto del mondo.



Figura 2.6: Fusion Garage Joojoo

Il dispositivo misura 324,5*199*18,9 mm ed è dotato di un display multitouch da 12 pollici, con una risoluzione pari a 1366*768 pixel, per un peso di 1,1 kg. Joojoo è basato sulla piattaforma nVidia Ion con processore Intel Atom a 1,6 GHz, è dotato di 1GB di memoria RAM, ed ha una memoria a stato solido da 4GB. Questo tablet consente la connettività Wi-Fi ed è dotato di un modulo Bluetooth; fornisce una porta USB, una videocamera VGA e altoparlanti integrati oltre all'accelerometro. L'azienda produttrice ha dichiarato un'autonomia di circa 5 ore ed il tutto è controllato da un sistema operativo Linux basato su un semplice browser; per un prezzo di 359 euro.

Pochi giorni dopo la sua uscita un sito internet specializzato ha pubblicato la recensione del dispositivo, che ha evidenziato forti lacune software a dispetto di una buon risultato hardware. Questo lavoro ha dimostrato la mancanza di un media player e di un visualizzatore di immagini, l'impossibilità di lanciare il browser senza cliccare su collegamenti appositi e qualche imperfezione riguardo alla tastiera su schermo. Per contro uno dei punti di forza del tablet JooJoo, oltre allo schermo, è la sua natura x86, che permette di installare una serie di sistemi operativi classici fino ai sistemi Windows.

Al momento però le vendite non hanno raccolto un grande successo, almeno a sentire le voci da oltreoceano che parlano di soltanto 64 unità consegnate nel primo mese di disponibilità.

2.6 ExoPC Slate

ExoPC Slate è un interessante tablet che arriva dal Canada e disponibile sul sito dell'azienda in preordine.

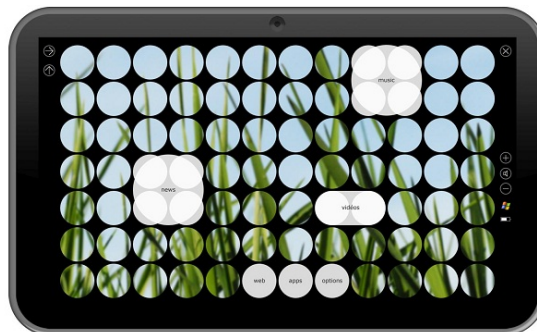


Figura 2.7: ExoPC Slate

Il dispositivo è un tablet multitouch da 8.9 pollici con risoluzione di 1024*600 pixel, che misura 25*16.8*2.1 cm e pesa 795 grammi. Slate è equipaggiato con un processore Intel Atom N270 a 1.6 GHz, 2 GB di RAM, SSD da 32 GB espandibile con schede di memoria SD, una webcam da 1.3 megapixel, 3 porte USB, connettività WiFi e bluetooth. La periferica dispone del sistema operativo Windows 7 pro ed offre un'autonomia di circa 4 ore. ExoPC Slate, come detto in precedenza, non è ancora disponibile sul mercato ma può essere ordinato ad un prezzo di 549 euro.

2.7 ICD Gemini

Prodotto dalla ICD, Gemini è l'ennesimo tablet (questa volta prodotto a metà tra Inghilterra e Stati Uniti) che andrà a concorrere, senza alcuna data precisa, con l'iPad di Apple.

Costruito sul chipset nVidia Tegra 2 da 1GHz, sarà dotato di un display touchscreen da 11.2 pollici con risoluzione di 1.366*768 pixel e dimensioni di 300*191*15,6 mm, ma non si hanno informazioni sul peso che la periferica potrà avere; in compenso è probabile che il display sia caratterizzato dalla tecnologia TN, che restituisce colori meno fedeli e soprattutto angoli di visione meno ampi rispetto ai pannelli LCD IPS, come quello montato sull'iPad. Secondo indiscrezioni, Gemini potrà fornire un lettore di schede SD fino a 32 GB, radio FM, GPS, accelerometro, bussola digitale, una porta microUSB e connettività WiFi e Bluetooth.



Figura 2.8: ICD Gemini

Altre peculiarità saranno gli altoparlanti stereo, e due webcam (una da 2 megapixel frontale, e una sul retro da 5 megapixel con autofocus), oltre al 3G integrato, alla registrazione video HD e ad un App Store opzionale.

Caratteristica ulteriore rispetto agli altri prodotti visti finora è la possibilità di effettuare telefonate, sfruttando il modulo 3G.

ICD Gemini adotterà il sistema operativo Google Android, ma non è ancora noto il prezzo di commercializzazione.

2.8 Samsung Galaxy Tab

L'azienda coreana Samsung ha lanciato ad inizio ottobre il suo primo minitabulet nominato Samsung Galaxy Tab. Questo dispositivo dalle dimensioni ridotte (190*120*12 mm) e dal peso contenuto di 380 grammi, è già disponibile anche sul mercato italiano. Il Galaxy Tab presenta uno schermo da 7 pollici TFT LED con una risoluzione di 1024*600 pixel, pieno supporto al multitouch, ed è configurato con un processore ARM Cortex A8 da 1 GB e 512 MB di memoria RAM. Per le connessioni sono supportate le tecnologie Wi-Fi, Bluetooth con il nuovo standard 3.0, 3G e A-GPS, mentre per l'archiviazione dati è disponibile la scelta tra 16 e 32 GB di memoria, espandibili con tecnologia microSD fino a 32GB. Il dispositivo sarà perfettamente in grado di fare e ricevere chiamate, utilizzando la SIM in esso inserita, anche se solo utilizzando un auricolare collegato al tablet. Samsung Galaxy Tab (che è anche dotato di fotocamera posteriore da 3 megapixel con flash e frontale da 1.3 megapixel per le video chiamate) garantisce, secondo l'azienda costruttrice, un'autonomia di 7 ore ed è gestito da un sistema operativo Android con supporto a Flash 10.1.

La periferica è commercializzata ad un prezzo di 699 euro per la versione base con memoria da 16 GB, forse un po' troppi per contrastare in modo efficace il predominio di iPad in questo mercato.



Figura 2.9: Samsung Galaxy Tab

2.9 Toshiba Folio 100

Anche l'azienda giapponese Toshiba ha presentato ufficialmente alla stampa, alla fine di quest'estate, la sua arma per guadagnarsi una fetta dell'ambito mercato dei tablet PC, che fino ad un anno fa era di scarso interesse per tutti: il nuovo Tablet Toshiba Folio 100. Questo è un dispositivo portatile dotato del sistema operativo Android 2.2 Froyo e che risulta molto interessante dal punto di vista prestazionale grazie alla presenza del chip nVidia Tegra 2 ad 1 GHz e 512 MB di memoria RAM.

Il tablet Toshiba, che ha dimensioni 281*181*14 mm ed un peso di 760 grammi, presenta uno schermo touchscreen capacitivo con una diagonale di 10,1" (25,7 cm con risoluzione 1024*600) ed è pensato per chi cerca un dispositivo portatile e versatile per la gestione dei contenuti digitali (compreso il supporto a Flash). Questa nuova periferica offre numerose opzioni di connessione come slot per schede di memoria SD, connessione HDMI, USB 2.0, Wi-Fi e Bluetooth. Inoltre sarà disponibile anche la versione Mobile Broadband (3G). La durata della batteria dichiarata dal costruttore supera le sette ore, mentre la funzionalità instant-on consente di accenderlo in meno di 30 secondi. Per quel che riguarda invece la memoria, i 16 GB disponibili nel dispositivo permettono di archiviare una buona quantità di



Figura 2.10: Toshiba Folio 100

dati, considerando la possibilità di ampliamento con SD card; altre caratteristiche rilevanti sono la presenza di una webcam da 1.3 megapixel e di accelerometro.

Il nuovo Folio 100 sarà disponibile, secondo il produttore, nel quarto trimestre del 2010 in Europa, Medio Oriente e Africa, ed il prezzo di commercializzazione dovrebbe essere di 399 euro (per la versione wi-fi).

2.10 Blackberry Playbook



Figura 2.11: Blackberry Playbook

Durante la conferenza annuale degli sviluppatori BlackBerry, svoltasi il 27 settembre, l'amministratore delegato Mike Lazaridis ha annunciato il nuovo tablet della società: il

BlackBerry Playbook. Tale dispositivo utilizzerà un sistema operativo chiamato BlackBerry Tablet OS (multitasking) che offrirà pieno supporto a nuove tecnologie come HTML5 e anche ai contenuti Flash tramite Flash 10.1.

Questo nuovo prodotto avrà una diagonale di 7 pollici, con risoluzione 1024*600, display multitouch capacitivo, basato su CPU dual-core Cortex A9 da 1 GHz, 1GB di RAM, e disporrà di una fotocamera da 3 megapixel frontale oltre ad una posteriore con un obiettivo da 5 megapixel, il tutto in circa 400 grammi di peso. Non si hanno certezze riguardo alla dimensione della memoria, anche se indiscrezioni parlano di 16 e 32 GB. Il Playbook sarà in grado di riprodurre video HD 1080p e, per quel che concerne le comunicazioni, oltre ad una porta HDMI ed un modulo Bluetooth integrato, prevede anche la connettività micro USB e Wi-Fi per la connessione a reti wireless.

BlackBerry ha indicato come periodo di lancio l'inizio del 2011 (per il modello wi-fi, quello 3G sarà sviluppato più avanti) e non c'è stata alcuna menzione del prezzo di vendita al dettaglio.

Come si vede da questa analisi, al momento non sono presenti sul mercato dei veri e propri antagonisti del celebre iPad di casa Apple.

A causa di vari motivi, che vanno dalle rinunce strategiche di alcuni costruttori, alle carenze tecnologiche di altri, oppure al ritardo nei progetti o ancora alla mancanza di tam-tam mediatico, nessuno è riuscito fino ad ora a scalare la sua posizione di riferimento nel settore.

I tablet prossimamente in arrivo non sembrano avere, per esempio, una durata della batteria paragonabile a quella della periferica Apple, anche dopo mesi e mesi che questa è già sul mercato.

Per i concorrenti rimarrà inoltre il problema che quando arriveranno sul mercato la clientela interessata a questo genere di prodotti sarà già stata in larga parte drenata dal successo di iPad. I nuovi modelli dovranno quindi offrire un sostanziale vantaggio per conquistarsi un posto al sole, ma a quel punto Apple non dovrà fare altro che fornire delle versioni della sua periferica con dei piccoli aggiornamenti, per mantenersi ai vertici del mercato tablet ancora per molto tempo, come è accaduto con iPhone.

iPad sta raccogliendo successo per la sapiente strategia di marketing e per l'effetto novità abbinato ad una estetica convincente ed all'accoppiamento tra piattaforma smartphone e grande schermo, nonostante la paradossale mancanza di funzioni tecniche importanti (esempio di questo fatto è l'introduzione del multitasking avvenuta con il rilascio del nuovo sistema operativo a novembre, e anche della possibilità di stampa in precedenza mancante). In qualche caso, addirittura, le restrizioni imposte da Apple potrebbero avere l'effetto di orientare il mercato: il fatto che il più diffuso e usato smartphone non supporti Adobe Flash, anziché danneggiare l'iPhone, ha spinto vari siti di primaria importanza a modificare le proprie pagine web realizzando delle versioni Flash free, affinché siano consultabili anche da questa periferica, senza lasciare antiestetiche zone vuote.

Una volta stabilita con iPad la testa di ponte sul mercato dei tablet, la proliferazione del mercato secondario di dock, connettori, cavi, custodie e altri accessori tutti pensati per questo dispositivo contribuiranno a consolidare l'asimmetria tra chi è arrivato primo e chi invece dovrà costruirsi da solo la struttura necessaria ed andarsi a cercare i clienti rimasti sulla piazza. Una corsa impegnativa, come dimostrato dagli esempi precedenti di iPod gli altri riproduttori multimediali e di iPhone contro smartphone generici.

Capitolo 3

Il sistema operativo

PHONE OS (recentemente ridenominato iOS) è il sistema operativo Unix-based sviluppato da Apple per l'iPhone, l'iPod touch e l'iPad. Come Mac OS X ha quattro livelli di astrazione: il Core OS layer, il Core Services layer, il Media layer ed il Cocoa Touch layer. Il sistema operativo occupa meno di mezzo gigabyte della memoria interna del dispositivo e, fino al rilascio della prima beta dell'iPhone SDK agli inizi di marzo 2008, non aveva una denominazione ufficiale.

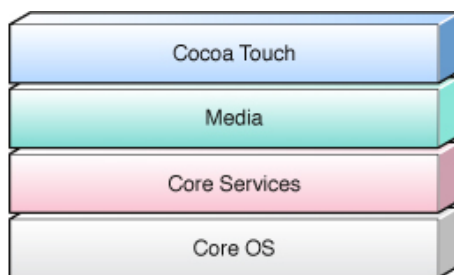


Figura 3.1: Gli strati del sistema operativo iOS

Lo strato Core OS si trova nel livello più basso del sistema operativo e rappresenta le funzioni più primitive, le operazioni più vicine al cuore del sistema (e solitamente quindi le più complesse): gestisce l'allocazione di memoria, il sistema di input e output, il networking, il threading, l'accesso ai file di sistema ed i drivers.

Core Services è lo strato che rappresenta tutti i servizi fondamentali che le applicazioni necessitano di gestire, come Address Book (la gestione dei contatti), il supporto e la gestione dei tipi di dati, dell'ora e delle stringhe, anche delle preferenze di sistema. Inoltre gestisce i servizi di localizzazione con longitudine e latitudine. Questi due livelli sono quasi esclusivamente scritti in linguaggio C.

Allontanandosi dal nucleo del sistema operativo si trova lo strato Media, che permette di amministrare tutto quello che rappresenta la grafica, media, audio e video. Media contiene

principalmente api C con l'eccezione di Core Animation, un motore avanzato per la gestione di animazioni sviluppato in Objective-C.

Il livello più esterno è denominato Cocoa Touch, ed è implementato esclusivamente in Objective-C. Questo è il livello più importante, e viene gestito da due framework: UIKit e Foundation. Questi frameworks coordinano aspetti molto importanti come la gestione delle applicazioni, della grafica e delle finestre, del multitouch e degli eventi, dell'interfaccia e degli oggetti che controllano il sistema. Inoltre gestiscono l'accelerometro, la libreria fotografica, la fotocamera, e le informazioni sul dispositivo.

Mac OS X e iPhone OS sono stati mescolati nel sistema operativo iPad diventando una versione del sistema operativo iPhone che è adattata alle nuove dimensioni dello schermo (attualmente il nuovo dispositivo monta lo stesso OS del telefonino, ma nella versione 3.2). Non si trovano molte modifiche estetiche ad eccezione di un dock simil Mac OS X e la possibilità di utilizzare sfondi personalizzati. Con questa nuova variante del sistema operativo Apple è possibile eseguire la maggior parte delle applicazioni ad esso dedicate, nella loro risoluzione originale o estesa per vederle a schermo intero, se previsto (tramite un tasto "2x"). Si trovano quindi tutte le applicazioni di default, ma soprattutto è possibile installare ogni programma già presente su AppStore: questo significa che, al lancio, l'iPad aveva già un parco software di oltre 100.000 applicazioni! Gli accelerometri interni alla periferica sono utilizzati da alcune applicazioni per rispondere allo scuotimento del dispositivo (solitamente usato come comando di annullamento) o la rotazione in tre dimensioni (passaggio da modo verticale a orizzontale).

Il primo problema di un sistema operativo tradizionale portato su tablet riguarda la risoluzione: tutta l'interfaccia è pensata per essere utilizzata tramite puntatore del mouse e una sua implementazione su schermi piccoli avrebbe sensibilmente ridotto l'area utilizzabile dall'utente a favore delle icone e delle barre menù, che sarebbero state più grandi rispetto alla versione desktop. Anziché rendere l'iPad un dispositivo più lento e con meno superficie utilizzabile per i contenuti, Apple ha quindi deciso di partire con quanto di buono fatto su iPhone.

Su questo tablet la risoluzione è bassa ma densa, perfetta per terminali di ridotte dimensioni. Anche per la gestione del multitouch è stata usata la stessa tecnologia dell'iPhone, che male si sarebbe comportata su un sistema operativo completamente diverso come Mac OS X. Mentre sull'interfaccia Apple avrebbe potuto lavorare e creare qualcosa di compatibile, tutte le applicazioni presenti su Mac avrebbero dovuto essere riscritte per poi poter essere utilizzate al meglio tramite multitouch. A livello pratico ci sarebbero stati pure dei problemi: l'utente che usa il mouse è abituato a puntare verso l'obiettivo, ma se si utilizza il touchscreen è difficile usare le dita come sue sostitute. Ecco perché Apple ha ideato Cocoa Touch, per rendere l'esperienza utente il più naturale possibile e con un sistema desktop questo sarebbe stato impossibile.

Pensando alla maggior parte degli altri dispositivi mobile ci sono diverse filosofie di pensiero: c'è chi usa il trackpad, chi il pennino, tutte scelte implementate proprio per imitare l'uso del mouse. Apple ha pensato diversamente e scegliere Mac OS X per iPad avrebbe

significato fare un passo indietro in quella che vuole essere la strada avviata dalla società di Steve Jobs: facilità di utilizzo, esperienza utente il più naturale possibile e touchscreen perfettamente funzionale.

Dal punto di vista software, come detto prima, la scelta sarebbe risultata errata: gli sviluppatori sarebbero stati costretti a riscrivere le loro applicazioni e senza poter utilizzare una piattaforma di sviluppo semplice e di successo come quella iPhone.

Il firmware 3.2 per iPad porta anche novità importanti, come il supporto ai display esterni tramite apposito cavo e che consente di visualizzare tutte le applicazioni lanciate direttamente su un monitor esterno, o come il file sharing, con una cartella condivisa nella quale poter gestire file e documenti da Mac/Pc ad iPad e viceversa.

Come affermato in precedenza mancherà il supporto a Flash Player, per cui tutti i siti che fanno uso del plug-in di Adobe non potranno essere visualizzati su iPad. Apple ha più volte spiegato che Flash è ormai una tecnologia passata, che penalizza le prestazioni e che presto verrà soppiantata da HTML5, il nuovo standard che sta prendendo piede su molti siti internet.

3.1 La critica più grande: il lancio senza multitasking

Una delle critiche più aspre rivolte al nuovo prodotto di Cupertino nel momento del lancio sul mercato, è stata quella di non poter far girare più applicazioni (che non siano quelle preinstallate da Apple) contemporaneamente.

In realtà l'iPad (come del resto l'iPhone e l'iPod touch, visto che montano tutti lo stesso sistema operativo) supporta nativamente il multitasking, ovvero è capace di far girare più processi contemporaneamente. Questo per il semplice motivo che il kernel dell'iPhone OS, ovvero il cuore del sistema operativo, è esattamente lo stesso kernel di Mac OS X, semplicemente ricompilato per girare su processori ARM (quelli che troviamo dentro iPhone, iPod touch e iPad).

Ovviamente tra Mac OS X e iPhone OS ci sono delle differenze sostanziali a livello di interfaccia utente ma entrambi sono capaci di far girare più processi, e quindi più applicazioni, contemporaneamente. Questa cosa risulta evidente nel momento in cui, utilizzando iPhone, arriva una telefonata mentre si sta ascoltando la musica dell'iPod o si naviga sul web con il browser Safari.

Nonostante il sistema operativo sia in grado di gestire più processi contemporaneamente, però, ci sono delle limitazioni che non sono tecniche ma imposte dalla casa produttrice. È la Apple che non permette all'utente di far girare più applicazioni allo stesso tempo; le poche in grado di girare in background (ovvero in secondo piano mentre l'utente è impegnato su un altro applicativo) sono quelle fornite dalla stessa Apple e sono quelle fondamentali per un iPhone/iPod/iPad: Telefono, Mail, SMS e iPod.

Le ragioni di questa scelta sono diverse e c'è da considerare che l'introduzione di una nuova funzionalità ha quasi sempre pro e contro. In questo caso dare la possibilità ai

programmi di girare in background vorrebbe dire anzitutto una riduzione sensibile dell'autonomia del dispositivo, ed in secondo luogo minori risorse hardware a disposizione per le applicazioni, visto che ovviamente dovranno dividerle. Infine c'è anche da considerare la questione sicurezza.

L'azienda non è rimasta con le mani in mano di fronte alle lamentele degli utenti su questo aspetto ed ha quindi cercato delle soluzioni alternative per poter quantomeno simulare il multitasking per il software non preinstallato. Gran parte delle applicazioni ha la necessità di rimanere attiva in background semplicemente per poter rimanere aggiornata e sincronizzata con un server centrale: è molto comodo, per esempio, poter ricevere messaggi via chat senza dover per forza tenere la relativa applicazione perennemente aperta, perché questo vorrebbe dire non poter fare altro durante il collegamento.

Apple ha pensato di risolvere questo problema introducendo su iPhone OS il concetto di notifiche push, cioè dei messaggi di notifica che le applicazioni possono inviare tramite server all'utente, il quale potrà quindi leggerli anche se l'applicazione stessa non è aperta. Questo sistema funziona piuttosto bene e soddisfa gran parte delle esigenze degli sviluppatori; esistono però applicazioni che non possono accontentarsi delle notifiche push, come per esempio una radio. Ascoltare la musica con iPod mentre si usa un'altra applicazione è molto comodo, e sarebbe altrettanto comodo poterlo fare con un'applicazione che fa streaming audio di una radio, ma in questo momento ciò non è possibile.

Molte volte si è sentito dire che la mancanza del multitasking su iPhone OS è dovuta anche ad una questione di sicurezza. Qualcuno sostiene che se le applicazioni fossero libere di girare in background potrebbero anche occuparsi di “affari loschi” senza che l'utente si accorga di nulla. In realtà sembra non essere così infatti il sistema di sicurezza di iPhone OS è piuttosto robusto e si basa principalmente sul fatto che, a meno di jailbreak e/o modifiche al firmware, la Apple ha il totale controllo su ciò che può o non può essere installato sul dispositivo. Un'applicazione che gira in background sarebbe comunque proveniente da AppStore, e quindi testata ed approvata dall'azienda.

Nonostante non sia possibile garantire al 100% un comportamento impeccabile da parte del software installato, resta il fatto che anche girando in background può fare esattamente le stesse cose che farebbe una qualsiasi altra applicazione, con l'unica differenza che l'utente non avrebbe di fronte un'interfaccia grafica.

Installare un virus su iPhone OS è quanto meno difficile, visto che ogni applicativo gira all'interno di una sandbox, ovvero può disporre di file e risorse che gli appartengono e non può andare a richiederli ad altre applicazioni, nè tantomeno al sistema operativo. Allo stesso modo il concetto di applicazione spyware è poco sensato, visto che una qualsiasi applicazione può fornire dati relativi all'utente senza che quest'ultimo se ne renda conto, sebbene il fatto di poter girare in background in questo caso potrebbe rappresentare un vantaggio.

3.2 Le novità del sistema operativo 4.2

Il nuovo sistema operativo per iPad è stato preceduto dal lancio della versione numero 4.0 di quello per iPhone, avvenuto insieme alla presentazione della quarta generazione di tale dispositivo nel mese di aprile.

Questo OS ha introdotto numerosi cambiamenti e migliorie riguardanti il telefonino (novità che poi saranno introdotte anche nel tablet) a partire dalla ridenominazione del sistema operativo stesso: da iPhone OS si è passati alla nuova terminologia iOS. La versione 4.2 dell'iOS (il primo upgrade per iPad) sarà rilasciata in versione stabile nel corso di novembre 2010. L'aggiornamento sarà molto significativo poiché rappresenterà il punto di incontro tra i dispositivi touch di Apple che a partire da questo aggiornamento infatti utilizzeranno tutti la stessa versione di iOS.

Queste saranno le principali nuove caratteristiche per iPad:

- introduzione di AirPrint, funzionalità di stampa tramite Wi-Fi;
- introduzione del multitasking;
- possibilità di creare ed organizzare cartelle, contenenti fino a 2160 applicazioni;

Va specificato che il multitasking secondo Apple per il momento deve "restare controllato e limitato per motivi di sicurezza, performances e consumi"; in pratica consiste in una serie di servizi dell'applicazione che possono rimanere in esecuzione in background. Questa tecnica, infatti, permette un risparmio energetico elevato tale da consentire l'esecuzione, seppur parziale, di circa 30 applicazioni, che in questa modalità inoltre non causano un rallentamento percepibile dall'utente, e soprattutto impedisce che programmi o tasks non autorizzati continuino a funzionare invisibilmente all'insaputa dell'utilizzatore.

3.3 Applicazioni preinstallate

Ecco una breve carrellata delle applicazioni che sono preinstallate nel nuovo dispositivo Apple, e quindi possono essere utilizzate già dalla prima accensione.

Tutte le applicazioni integrate sono state ridisegnate rispetto alle versioni precedenti per sfruttare appieno le funzionalità evolute di iPad ed il suo ampio schermo.

Contatti

I contatti vengono visualizzati in modo innovativo, ed il loro design li fa assomigliare ad una vera e propria agenda. iPad permette di scorrere i contatti sfiorando lo schermo con un dito, oppure usando la ricerca integrata per trovare velocemente un nome. Se servono indicazioni per localizzare un contatto, toccando l'indirizzo viene aperta automaticamente l'applicazione Mappe.

Calendario

Calendario su iPad permette di gestire gli impegni, organizzandoli per giorno, settimana o mese. In verticale è possibile vedere tutto il mese o un singolo giorno, mentre in orizzontale apparirà un elenco con le attività della giornata da un lato, ed il programma dettagliato dall'altro.

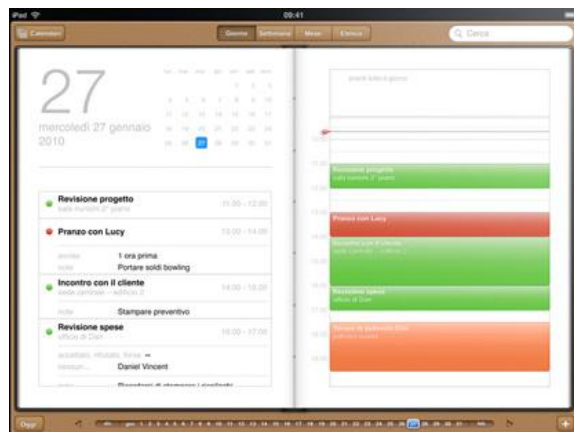


Figura 3.2: Calendario

Mappe

Mappe offre un'esperienza più fluida e soddisfacente che nella sua versione iPhone: la cartografia è fornita da Google come servizio, ma l'applicazione è realizzata dagli ingegneri di Cupertino; è possibile accedere anche alla vista Street View in maniera molto fluida.

Video

Anche per la visualizzazione dei filmati l'interfaccia è semplice e facile da gestire con una applicazione per l'accesso diretto che si lancia dalla schermata principale, denominata Video, o attraverso la sezione video dell'applicazione iPod. I formati supportati arrivano al 720p (1280x720) corrispondenti a quelli dei filmati predisposti per il media player Apple. Il full HD non ha senso su un apparecchio con uno schermo così piccolo: l'occhio umano non riuscirebbe a percepire la differenza rispetto a quello che iPad permette di vedere normalmente.

YouTube

YouTube è molto potente e consente di gestire a tutto campo il video in maniera più ricca e articolata che non con iPhone: un buon indicatore del fatto che la maggior parte del

web non usa Flash neanche in questo settore. YouTube è integrato comunque anche dentro Safari, ed è possibile quindi visualizzare un filmato all'interno di una pagina web nel suo box di partenza premendo play oppure portarlo alle massime dimensioni dello schermo con un semplice gesto a pinza in allargamento.

App Store

App Store offre oltre 250.000 app ed iPad le supporta praticamente tutte, dai giochi alle applicazioni per il lavoro e tanto altro; le differenze rispetto all'iPhone sono poche e legate alla sola organizzazione del layout.

Safari

Safari, il web browser Apple, su iPad è un incrocio tra la versione desktop e l'iPhone, ma è più simile alla versione del secondo poichè non supporta flash; consente di vedere intere pagine in orizzontale o in verticale sull'ampio schermo Multi-Touch.

Mail e Note

Mail e note sono il miglior esempio dei cambiamenti effettuati rispetto ad iPhone: per avere uno schermo più grande, l'applicazione viene eseguita in una modalità a due colonne con menu a discesa ed una grande video-tastiera virtuale per digitare i testi. Grazie all'ampio display con iPad si prendono appunti facilmente e si mantengono le informazioni necessarie. Si può anche spedire un promemoria via e-mail.

iPod

Per quanto riguarda la modalità iPod, viste le maggiori dimensioni dello schermo Apple ha creato un mix tra gli elenchi ridotti a lunghe liste che è possibile utilizzare su iPhone e iPod touch e la ricchezza di informazioni disponibile sulle versioni di iTunes Mac e PC. Come sulle versioni desktop, anche in iPad è presente la comoda opportunità di consultare i testi delle canzoni.

Immagini

E' possibile gestire le foto per album, eventi, volti e luoghi: i primi tre sono "mucchietti" di immagini tra le quali si può curiosare in punta di dita e che possono essere gestite con eleganza. I trasferimenti possono avvenire da Mac e PC o da una fotocamera tramite gli accessori per il collegamento USB o di lettura SD CARD.

Una novità è rappresentata dall'interazione con i luoghi che consente di combinare le coordinate geografiche con gli scatti e permette, premendo sui singoli pin, di mostrare le foto acquisite in quel preciso punto.

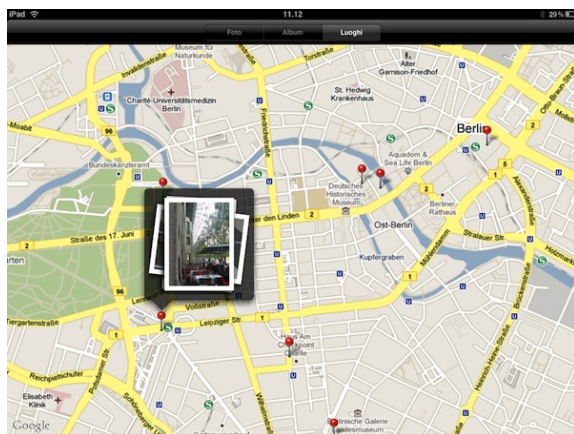


Figura 3.3: Immagini in modalità “luoghi”

iBooks

iBooks, l'applicazione che converte l'iPad in un lettore eBook, è creata da Apple stessa. Il suo design emula un libro e permette di interagire cliccando su qualsiasi angolo per cambiare pagina o muovere un dito per sfogiarla, modificare le dimensioni del testo, passare direttamente da un capitolo ad un altro dall'indice, e così via. Come detto in precedenza esso utilizza epub, il formato standard per i lettori di libri elettronici (ad eccezione del Kindle), si è così in grado di importare i libri in aggiunta a quelli che si acquistano direttamente nel nuovo negozio di Apple, l'iBookstore, nel quale si trovano, almeno negli USA, alcuni degli editori più importanti tra cui Penguin, Harper Collins, Macmillan e Hachette.

L'applicazione non è pre-installata ma è scaricabile gratuitamente.

Capitolo 4

Lo sviluppo di applicazioni

PER POTER SVILUPPARE APPLICAZIONI per iPad si ha la necessità di avere a disposizione tutta una serie di strumenti atti a questo obiettivo. L'SDK (Software Developer Kit) è il pacchetto che Apple mette a disposizione dei programmatori per fare questo lavoro e può essere scaricato, previa registrazione all'iPhone Developer Program sul sito dell'azienda, avendo un ID ed una password Apple.



Figura 4.1: SDK Apple

Il download dell'SDK è gratuito e contiene l'ultima versione del tool di sviluppo Xcode insieme ad una suite di strumenti aggiuntivi per lo sviluppo di applicazioni per iPhone, iPad e iPod touch, come l'iPhone Simulator.

Xcode consente di collaudare le applicazioni sviluppate sul simulatore, ma il developer kit è disponibile solo per Mac, quindi un PC con sistema operativo Windows o Linux non può essere utilizzato.

Per testare sull'iPhone/iPad e commercializzare (anche se in forma gratuita) le applicazioni sviluppate si deve acquistare una licenza particolare disponibile in due versioni distinte:

Standard Program dal costo di 99\$ (fornisce la possibilità di installare il software su 100 dispositivi Apple diversi e pubblicarla anche nell'App Store);

Enterprise Program dal costo di 299\$ (con funzionalità più evolute per le aziende).

Come accennato precedentemente per sviluppare software destinato all'iPad con iPhone SDK bisogna avere a disposizione un Mac basato su Intel, con sistema operativo Mac OS X Snow Leopard o versioni successive.

Le applicazioni create non possono avere una dimensione superiore ai 2 GB, possono essere rilasciate in forma gratuita o a pagamento e, in quest'ultimo caso, il prezzo deve essere compreso tra i 0.99\$ ed i 999\$. Ogni applicativo necessita di una firma digitale con un sistema di chiavi fornite da Apple, che provvederà anche a firmare ulteriormente il software prima di renderlo scaricabile dell'App Store.

Per tutta la documentazione necessaria è disponibile la iOS Reference Library, in cui è possibile trovare oltre 700 documenti utili al lavoro di sviluppo raggruppati in categorie, in modo da poter trovare rapidamente le informazioni cercate. Queste riguardano indicazioni sugli strumenti, sulle migliori pratiche di lavoro con una serie di linee guida per l'interfaccia umana e sui metodi di progettazione per la creazione di applicazioni innovative, incorporando una raccolta approfondita di documentazione tecnica, codici di esempio, guide e articoli per lo sviluppo. È possibile inoltre vedere gli esperti Apple discutere di tutto, da come iniziare con l'iPhone SDK, agli strumenti e alle tecnologie utilizzate per creare del nuovo software potendo includere tali nuove funzionalità in ciò che si sta creando.

È utile ricordare che la grande maggioranza delle applicazioni create per iPhone e iPod touch sono compatibili con iPad, ma anche che questo ha delle funzionalità e un hardware diversi da quelli dei dispositivi più piccoli, ed in quanto tale permette di realizzare programmi molto più complessi e sofisticati.

Per diventare sviluppatori Apple, e avere quindi la possibilità di sfruttare appieno le caratteristiche della periferica, è necessario registrarsi sul sito dell'azienda, venendo guidati attraverso il processo per la compilazione della richiesta e presentare l'iscrizione al programma selezionato, basato sulla piattaforma per cui si desidera sviluppare, scegliendo tra i sistemi operativi iOS e Mac OS X oppure il browser Safari.

Dopo aver acquistato il programma, si riceverà una mail da Apple Developer Support con informazioni sulle modalità per accedere alle risorse dedicate.

Riepilogando, il ciclo di sviluppo di un'applicazione per iPad può essere riassunto brevemente in tre fasi:

Sviluppo : con l'iPhone SDK si crea l'applicazione desiderata, avendo la possibilità di accedere al sito Apple per avere un efficace supporto comprendente video, codice di esempio, documentazione tecnica, con la possibilità di discutere con gli sviluppatori e altri ingegneri Apple;

Test : grazie all'iPhone Simulator è consentito controllare e collaudare il comportamento reale del software direttamente sul computer, con la possibilità di verificarne le

prestazioni. L'iOS Developer Program prevede due modalità di supporto tecnico, dove gli ingegneri Apple forniscono assistenza a livello di codice e guide utili, oppure indicano la documentazione tecnica adeguata per velocizzare il processo di sviluppo;

Distribuzione : L'iOS Developer Program Standard fornisce l'accesso ad App Store in cui è possibile distribuire le proprie applicazioni gratuitamente o a pagamento a milioni di iPad, iPhone e iPod touch. L'App Store è accessibile tramite Wi-Fi e reti cellulari dalle periferiche, ed i clienti possono conoscere e scaricare nuovi software ovunque si trovino ed essere avvisati ogni volta che è disponibile un aggiornamento, direttamente sui loro dispositivi. Distribuendo la propria applicazione sullo store Apple è possibile sceglierne il prezzo, ottenere il 70% dei ricavi delle vendite, ricevere dei controlli sulle entrate mensili, senza avere nessun costo per le applicazioni gratis, per l'hosting e per il marketing.

Un'importante novità arrivata con l'introduzione dell'iPad, è la possibilità fornita dall'SDK Apple di sviluppare tre tipi diversi di applicazioni, a seconda del target per cui sono pensate:

iPhone Applications : sono ottimizzate per girare su iPhone e iPod Touch. Queste possono essere avviate anche su iPad nella loro risoluzione originale o eventualmente possono essere impostate durante l'esecuzione in visualizzazione raddoppiata, adatta al display più grande del nuovo dispositivo Apple.

iPad Applications : queste sono ottimizzate per sfruttare i vantaggi offerti dalle nuove funzionalità del tablet Apple; tali software saranno però ovviamente compatibili solo con iPad.

Universal Applications : il kit di sviluppo supporta infine lo sviluppo di applicazioni universali, compatibili con tutti i dispositivi su cui è montato iOS.

Un'applicazione universale è in grado di riconoscere su quale dispositivo viene eseguita e predisporre automaticamente le migliori impostazioni per l'utilizzo in tale periferica. Questo significa che codesti software, per essere progettati correttamente, devono tener conto delle caratteristiche e delle funzionalità hardware di un unico ipotetico dispositivo, obbligando lo sviluppatore ad utilizzare le sole presenti su tutti i dispositivi iOS.

Capitolo 5

L'ambiente di sviluppo Apple

IL SISTEMA OPERATIVO Mac OS X Snow Leopard include una suite completa di strumenti di sviluppo che sono predisposti per permettere la creazione di applicazioni di alta qualità, tra cui:

- Xcode: il cuore degli strumenti per sviluppatori.
- Interface Builder: per creare l'interfaccia utente.
- Strumenti: utile a visualizzare le prestazioni dell'applicazione.

5.1 Xcode

Xcode è un ambiente di sviluppo integrato (Integrated Development Environment, IDE) sviluppato da Apple per agevolare la creazione di programmi per Mac OS X e iPhone OS.

L'ambiente offre un'interfaccia elegante e potente per creare e gestire progetti di sviluppo software in Mac OS X: si possono organizzare e modificare i file sorgente, visualizzare la documentazione, eseguire il debugging del codice ed ottimizzare le prestazioni del prodotto. Xcode è altamente personalizzabile con funzionalità che permettono di lavorare in modo intuitivo ed efficiente. L'IDE consente di gestire in modo automatico tutti i dispositivi di prova, le applicazioni iPhone, iPad ed iPod Touch con i certificati corretti, e l'installazione dei programmi sulle stesse periferiche.

Per sviluppare una nuova applicazione iOS si deve iniziare con la creazione di un nuovo progetto, che gestisce tutte le informazioni associate al programma come i file sorgente ed i file di interfaccia. Il cuore di ogni progetto è la finestra (vedi figura) che consente di accedere rapidamente a tutti gli elementi chiave dell'applicazione.

L'ambiente di sviluppo è fornito di un editor di testo avanzato, che supporta svariate funzionalità come ad esempio il completamento del codice, il suo ripiegamento (per nascondere temporaneamente alcuni blocchi) e la colorazione della sintassi. Nel caso in cui sia

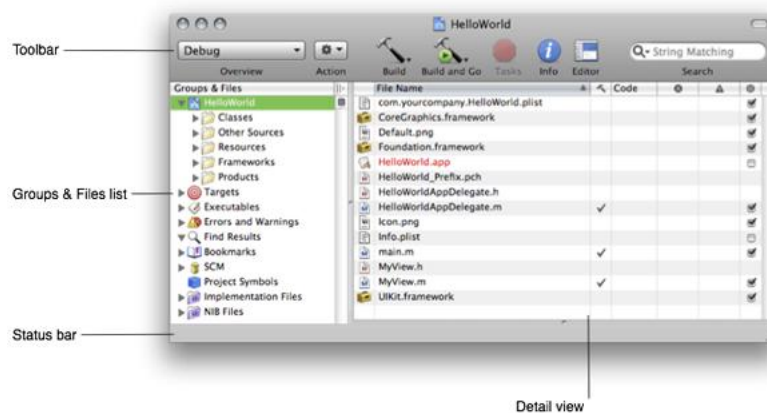


Figura 5.1: Finestra del progetto in Xcode

necessaria della documentazione, l'assistente di ricerca fornisce quella attinente al contesto, fornendo la possibilità di navigare e cercare ulteriori informazioni.

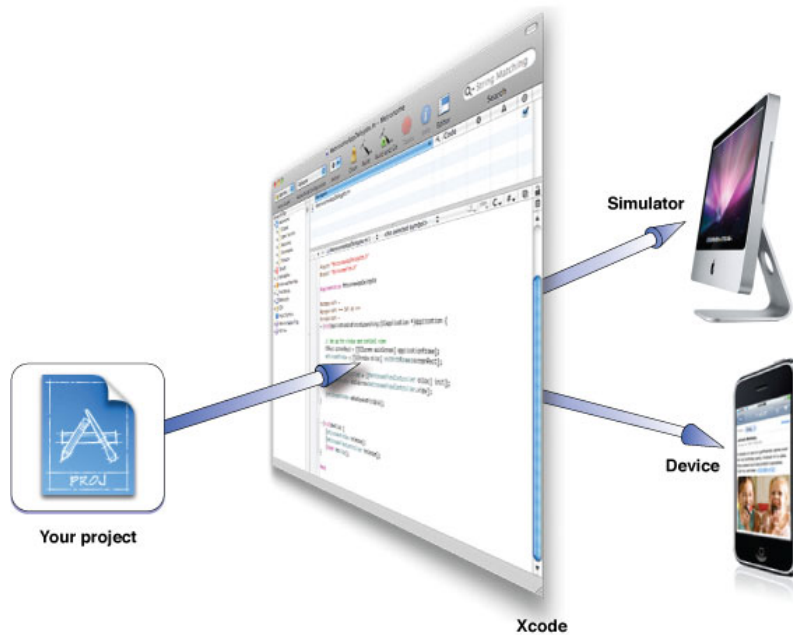


Figura 5.2: L'esecuzione di un progetto da Xcode

Quando si genera l'applicazione in Xcode è possibile decidere se eseguirla sull'iPhone Simulator o in un dispositivo (se questo è disponibile). Il simulatore fornisce un ambiente locale per testare il software ed essere certi che si comporti nel modo desiderato, per esempio che il punto di vista cambi in modo corretto quando si ruota il dispositivo; inoltre si possono simulare i gesti touch utilizzando il mouse. Quando si è soddisfatti del comportamento di base del software provato al simulatore, si può farlo funzionare su un dispositivo reale basato su iOS e collegato al computer.

L'ambiente di sviluppo Xcode include GCC, che è in grado di compilare codice C, C++, Java, Javascript, Applescript, Gnu e Free Pascal e anche Ada ma soprattutto Objective C, il linguaggio adoperato per scrivere tutte le applicazioni per dispositivi portatili Apple. Una delle caratteristiche tecnologicamente più avanzate di questo editor è il supporto alla distribuzione in rete del lavoro di compilazione: usando Bonjour e Xgrid è in grado di effettuare le operazioni su più computer riducendo i tempi. Supporta inoltre la compilazione incrementale ed è in grado di analizzare il codice mentre viene scritto.

Questo ambiente di sviluppo semplifica la comprensione degli eventuali errori di programmazione con la finestra dei risultati di costruzione. In una unica vista di log, gli errori vengono filtrati per mostrare le informazioni più importanti del compilatore: facendo clic su una voce all'interno dell'elenco si passa rapidamente al codice incriminato e le informazioni necessarie per risolvere il problema sono facili da trovare e da capire.

5.1.1 Xcode: il debugger

Il debugger consente di vedere che cosa succede quando il programma viene eseguito ed è di grande aiuto quando si sta cercando di scoprire cosa c'è che non va con il codice: è possibile mettere in pausa l'esecuzione in qualsiasi momento, analizzarla riga per riga, ed esaminare i valori delle variabili del programma.

Xcode è impostato in modo da non dover fare molto lavoro per eseguire il debugger, infatti per far comparire la finestra dedicata ed eseguire l'operazione è sufficiente premere un pulsante dedicato nella barra degli strumenti dell'editor.

La finestra del debugger ha quattro aree. Nella parte superiore è situata la barra degli strumenti, che dispone di pulsanti per svolgere le funzioni di debug più comuni. Sotto la barra degli strumenti a sinistra è presente lo stack di chiamate effettuate dal programma in esecuzione, a destra invece si trova il visualizzatore delle variabili: questo consente di vedere i valori che assumono in un preciso momento le variabili locali alla funzione corrente ed in generale di tutte quelle che hanno a che fare con il codice in esecuzione in quell'istante. Nella parte inferiore della finestra del debugger è presente l'editor, che consente di vedere quale riga di codice del programma è attualmente in esecuzione.

L'avvio del debugger non fa nulla se manca l'impostazione dei punti in cui deve essere sospesa l'esecuzione: i breakpoints servono proprio a questo scopo, ed è importante sistemarli nelle zone in cui il programma deve essere messo in pausa. Un'altra funzionalità

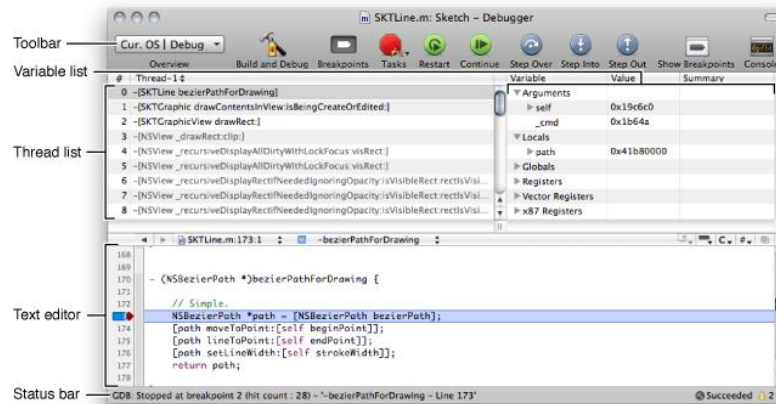


Figura 5.3: Il debugger di Xcode

molto importante consiste nell'esecuzione di una riga di codice alla volta, per analizzare gli effetti che ognuna di esse produce.

5.2 Interface Builder

Interface Builder è un software che aiuta a progettare le interfacce utente delle applicazioni in corso di sviluppo, sincronizzando automaticamente il codice sorgente delle modifiche apportate in Xcode. Si possono scegliere i componenti grafici desiderati da una libreria di elementi configurabili ed organizzarli nella finestra in fase di creazione servendosi dell'aiuto di guide di layout, con la possibilità di aggiungere sofisticate animazioni, creare transizioni ed effetti fra i vari stati dell'interfaccia oppure ancora ombre 3D che sottolineano i controlli.

Tale applicativo semplifica anche la risoluzione dei problemi, perché il posizionamento dei componenti ed il loro collegamento ai relativi metodi nel codice sorgente si svolge tramite un'interfaccia utente che permette di svolgere le operazioni necessarie con pochi click del mouse.

I componenti sopra citati includono i controlli standard del sistema, come i bottoni, i campi di testo ed i pulsant, oltre a vari tipi di viste, anche su misura, per rappresentare il modo in cui l'interfaccia apparirà a video. Altri controlli usati sovente in applicazioni per iOS sono i UIPickerView ed i DatePicker: questi sono essenziali per permettere all'utente la scelta tra varie possibilità, ed hanno il ruolo che nelle applicazioni per periferiche dotate di mouse o di altri dispositivi di puntamento è ricoperto dalle combo-box. Dopo aver posizionato i componenti sulla superficie della finestra, è possibile trascinarli nella posizione desiderata, configurare i loro attributi utilizzando la finestra di ispezione e stabilire le relazioni tra questi oggetti ed il codice sorgente.

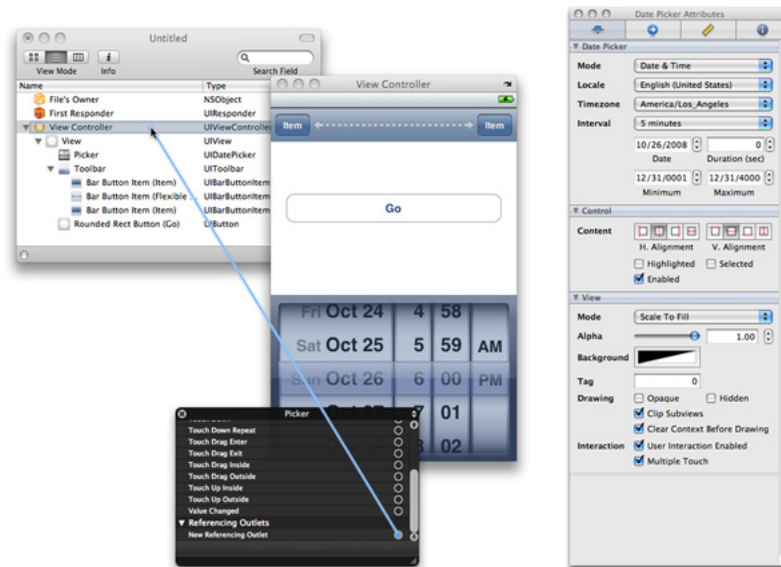


Figura 5.4: Interface Builder

Nel complesso l'utilizzo di Interface Builder consente di risparmiare diverso tempo nel creare l'interfaccia utente dell'applicazione; dato che è un editor visuale permette di osservare esattamente come le schermate appariranno a runtime, ed inoltre non è necessario implementare il codice per creare, configurare e posizionare gli oggetti che compongono l'interfaccia.

5.3 Strumenti

Da tempo gli strumenti per il monitoraggio delle prestazioni svolgono un ruolo essenziale nei processi di sviluppo. Il loro limite, però, è che forniscono solo un'immagine parziale del funzionamento delle applicazioni. Scegliere il tool migliore può essere difficile, e non c'è modo di confrontare i dati fra le varie analisi per avere una panoramica completa sul comportamento del software.

Strumenti in Mac OS X invece, permette di visualizzare più aspetti delle performance durante l'esecuzione sul simulatore o su un dispositivo, ed offre grafici temporali per monitorare uso della CPU, I/O del disco, utilizzo della memoria e vedere come interagiscono: tutto per avere un'immagine integrale del comportamento dell'applicativo che si sta sviluppando. La vista timeline può raffigurare tutti i diversi tipi di informazioni fianco a fianco, cosa che permette di correlare il comportamento generale dell'applicazione, senza limitarlo ad una determinata area.

L'utility consente inoltre di memorizzare i dati provenienti da cicli multipli in modo che sia possibile notare se il comportamento dell'applicativo è effettivamente migliorato o se necessita di ulteriore lavoro.

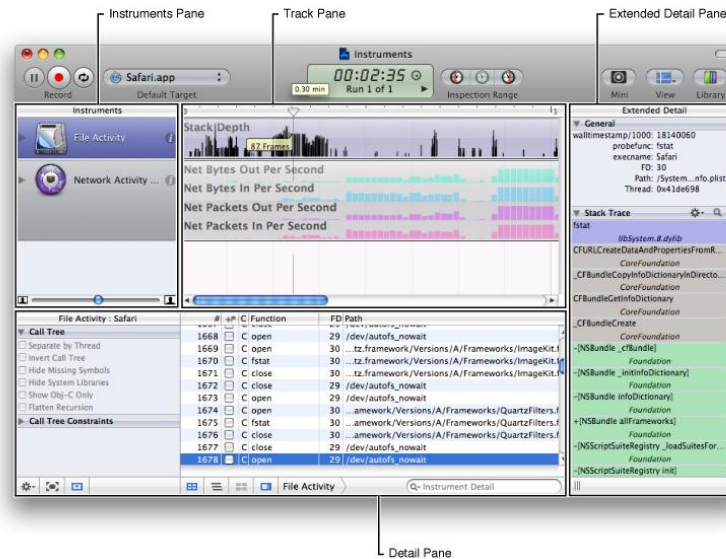


Figura 5.5: Strumenti

5.4 Clausole per lo sviluppo software

Oltre all'ambiente di sviluppo integrato Apple, visto il successo riscontrato dalle periferiche dell'azienda, sono stati creati altri ambienti per permettere lo sviluppo di applicazioni anche su sistemi operativi diversi da Mac OS X e con altri linguaggi di programmazione rispetto ad Objective C. Per cercare di aggirare gli standard, le restrizioni e le procedure che Apple richiede a chi voglia scrivere software da eseguire sulle sue macchine, sono state sviluppate varie suite comprendenti compilatori e API private che danno la possibilità di sviluppare in diversi linguaggi, traducendo poi il codice stesso oppure gli eseguibili e le librerie in applicazioni native.

Lo scorso aprile però, in concomitanza con il lancio del suo primo tablet, l'azienda ha modificato una clausola della licenza per l'utilizzo del Software Development Kit (necessario per portare i programmi sui propri dispositivi) e tra le novità è stato introdotto il divieto nell'utilizzo di software alternativi per lo sviluppo, o meglio, la conversione di applicazioni iPhone partendo da linguaggi diversi dall'Objective C.

“Le applicazioni possono utilizzare solamente le API documentate nella maniera prescritta da Apple e non devono utilizzare o chiamare altre API private. Le applicazioni devono

essere originariamente scritte in linguaggio Objective-C, C, C++, o JavaScript quando eseguite dall'engine iPhone OS WebKit, e solo il codice scritto in C, C++, e Objective-C può essere compilato e collegato direttamente alle API documentate (per esempio applicazioni che si collegano ad API documentate attraverso una traduzione intermediaria, o un livello di compatibilità o un tool sono vietate)."

L'amministratore delegato di Apple, Steve Jobs, a seguito delle numerose polemiche sorte in seguito a tale decisione, ha motivato la scelta, rispondendo all'e-mail di un utente:

"Apple ha creato una piattaforma di sviluppo senza precedenti, in grado di offrire agli sviluppatori una certa immediatezza ed una semplicità assoluta nella costruzione del codice e dell'interfaccia grafica dell'applicazione. Creare piattaforme alternative in grado di appoggiarsi alla nostra non fa altro che confondere gli sviluppatori e sicuramente non fa bene a noi dato che blocca i progressi e i possibili miglioramenti di tutto ciò che abbiamo creato."

In previsione del lancio del nuovo sistema operativo per iPad in novembre, Apple ha proceduto però ad ammorbidire la sua politica riguardo allo sviluppo delle app per iOS.

Questo cambiamento sostanziale dell'azienda potrebbe essere legato alle accuse ed indagini riguardo a presunte violazioni delle leggi anti-trust.

Tornando in parte sulle decisioni prese l'8 aprile 2010, l'azienda ha eliminato dal suo Developer Program License Agreement le restrizioni sull'uso di strumenti di terze parti al posto di quelli del proprio SDK: essi potranno essere impiegati purché le applicazioni risultanti non scarichino codice.

Inoltre, per la prima volta, sono state pubblicate le linee guida delle procedure di verifica e approvazione del software in vista della pubblicazione sullo store on-line.

Questo il comunicato della Apple:

"App Store ha rivoluzionato il modo di sviluppare e distribuire le applicazioni mobili. Con oltre 250.000 app e 6,5 miliardi di download, l'App Store è diventato la più grande piattaforma per applicazioni mobili del mondo ed ha permesso agli sviluppatori di guadagnare oltre un miliardo di dollari dalla vendita delle loro app. Il nostro desiderio è di continuare a rendere l'App Store un'esperienza sempre migliore. Per questo abbiamo ascoltato i nostri sviluppatori, prendendo in considerazione le loro opinioni e il loro feedback. Sulla base dei loro suggerimenti, abbiamo deciso di apportare alcuni importanti cambiamenti alla nostra licenza iOS Developer Program per quanto concerne le sezioni 3.3.1, 3.3.2 e 3.3.9, al fine di mitigare alcune limitazioni che avevamo introdotto all'inizio dell'anno. In particolare, abbiamo deciso di mitigare tutte le limitazioni relative agli strumenti di sviluppo usati per creare app iOS, purché tali applicazioni non comportino il download di codice. Ciò dovrebbe dare agli sviluppatori la flessibilità che desiderano, pur mantenendo un elevato livello di sicurezza, come da noi richiesto. Inoltre, per la prima volta, pubblicheremo le Linee guida sulle modalità di revisione adottate su App Store per aiutare gli sviluppatori a compren-

dere meglio in che modo rivediamo e analizziamo le app che riceviamo. Ci auguriamo che questa operazione aumenti ulteriormente la nostra trasparenza, contribuendo allo sviluppo di applicazioni ancora migliori per l'App Store. L'App Store è probabilmente una delle più importanti pietre miliari nella storia delle applicazioni mobili. Collaborando con i nostri sviluppatori, continueremo a sorprendere e deliziare i nostri utenti con app mobili sempre più innovative."

Tale cambiamento nella politica Apple molto probabilmente farà tornare agli onori della cronaca nomi come MonoTouch, Corona SDK, Titanium, RevMobile, e non da ultimo Adobe Flash Professional CS5, tutti ambienti di sviluppo alternativi che in primavera sembravano destinati ad avere vita breve, e senza trascurare quelli che verranno presentati in futuro.

Questi software hanno la caratteristica di consentire all'utente di scrivere il proprio codice in qualche linguaggio di programmazione diverso dallo standard Objective C, come ad esempio .NET, Java, revTalk o ancora Lua, HTML e CSS, trasformando poi gli eseguibili e le librerie in applicazioni native per iPad, iPhone e iPod Touch. Tali kit promettono agli sviluppatori di utilizzare vari framework e linguaggi più efficienti, che richiedono l'implementazione di meno righe di codice, per creare applicazioni funzionanti su dispositivi Apple traendo vantaggio delle API native iPhone.

Il costo di questi applicativi però risulta essere frequentemente molto elevato e la qualità del prodotto difficilmente può essere quella ottenuta seguendo in modo fedele le indicazioni dettagliate dei documenti e delle librerie Apple, anche se certamente lasciano una maggiore libertà di scelta agli sviluppatori.

L'ambiente di sviluppo fornito con l'SDK Apple, unitamente alle librerie a disposizione, alle convenzioni sul design che contraddistingue il software presente sui dispositivi dell'azienda, e ad anni di esperienza e miglioramenti continui nell'assistenza agli utenti, fanno di questo il prodotto migliore per sviluppare applicazioni per le periferiche portatili Apple ed iPad in particolare. Primaria importanza ricopre il fatto che questo è un pacchetto completo, senza costi aggiuntivi, sviluppato dall'azienda stessa che produce il dispositivo e senza la necessità di aggiunte o controlli di compatibilità con altri software che gli ambienti alternativi richiedono: per questo il programmatore è facilitato nel seguire le linee guida e le regole da rispettare perché la propria applicazione venga approvata e resa disponibile sull'App Store.

Capitolo 6

Il linguaggio 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).

Come lo stesso nome suggerisce, questo linguaggio è un'estensione ad oggetti del celeberrimo 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 oggetti, classi ed istanze.

6.1 Le classi

In questo linguaggio di programmazione ogni oggetto deriva in maniera diretta o indiretta da una classe base, così chiamata perché non deriva da nessun'altra: questa è denominata NSObject, e da essa derivano quasi tutte le classi di Objective C; le altre invece implementano un protocollo (una sorta di interfaccia Java) che le riconduce come comportamento ad NSObject e quindi, praticamente, qualsiasi oggetto deriva da questa classe.

La definizione di una classe in Objective C prevede due parti:

interfaccia (interface) : è pubblica, viene codificata in un file che ha estensione .h ed include tutte le dichiarazioni delle variabili di istanza della classe, delle proprietà e delle firme dei metodi;

implementazione (implementation) : è privata, viene definita in un file con estensione .m e contiene il codice di ogni metodo ed il completamento delle proprietà dichiarate nell'interfaccia.

I due file devono essere collegati e questa operazione viene svolta nel file di implementazione con l'istruzione `#import`, molto simile ad `#include` presente nel linguaggio C: la differenza sta nel fatto che la prima fa in modo che lo stesso file non sia mai incluso più di una volta.

L'interfaccia di una classe ha una sintassi standard: inizia sempre con la parola chiave `interface` e termina con la parola chiave `end`.

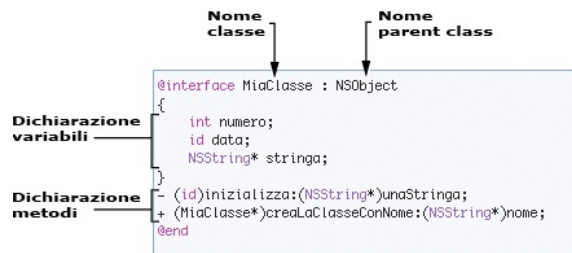


Figura 6.1: Esempio di interfaccia

Subito dopo `interface` seguono nell'ordine il nome della classe, i due punti ed il nome della classe base, quella da cui l'oggetto deriva. Per convenzione il nome delle classi viene scritto secondo lo standard "CamelCase", quindi le iniziali delle parole che compongono la classe sono in maiuscolo.

All'interno delle parentesi graffe si definiscono le variabili di istanza ed al di fuori, prima della parola chiave `end`, vengono inserite le eventuali dichiarazioni dei metodi e delle proprietà (metodi setter/getter).

Il file di implementazione invece inizia sempre con una riga di codice che importa il file di interfaccia:

```
#import "Interfaccia.h"
```

Come nel file d'interfaccia, anche in questo caso tutto il codice viene scritto all'interno delle parole chiave `implementation` `NomeClasse` e `end`.

Ogni metodo implementato inizia con la definizione scritta nel file di interfaccia ed il suo corpo è delimitato da parentesi graffe.

```

@implementation MiaClasse

- (id) inizializza:(NSString*)unaStringa{
    if (self == [super init]){
        numero = 0;
        data = nil;
        stringa = [unaStringa copy];
    }
}

+ (MiaClasse)creaLaClasseConNome:(NSString*)nome{
    return [[[self alloc] initWithString:nome ] autorelease];
}

@end

```

Figura 6.2: Esempio di implementation

6.2 I metodi

In Objective C possono essere dichiarati due tipi di metodi: di istanza o di classe. Il primo si riferisce ad una particolare istanza, il secondo invece viene invocato direttamente dalla classe e va ad operare su alcune proprietà generali.

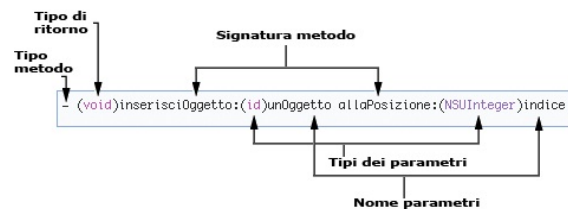


Figura 6.3: Esempio di metodo

La signature, ovvero il nome del metodo, in questo esempio è “inserisciOggetto: allaPosizione:”.

Come prima cosa è presente l'identificativo del tipo di metodo, in questo caso d'istanza: se come nell'esempio un metodo inizia con un meno (`-`) sarà d'istanza, se invece è preceduto da un simbolo (`+`) esso sarà di classe. Il (`void`) è il tipo di ritorno e quindi il tipo di oggetto ritornato dal metodo. I nomi dei parametri, invece, sono i termini con cui si fa riferimento ad essi all'interno del metodo; seguendo l'esempio, nel metodo “inserisciOggetto:” sarà presente una variabile “unOggetto”, non definita all'interno, ma che costituisce proprio il parametro.

Se il metodo ne accetta più d'uno è possibile inserire delle parole arbitrarie che spiegano il significato dei parametri che le seguono.

Quindi la struttura generale di un metodo è così definita:

```
tipo_metodo (tipo_di_ritorno) nome_del_metodo: (tipo_parametro) nome_parametro
```

Come detto in precedenza il modello di programmazione dell'Objective C è basato sullo scambio di messaggi tra oggetti: non si chiama un metodo, ma si invia un messaggio.

A differenza di molti linguaggi che utilizzano la dot notation (oggetto.metodo) per la chiamata ad un metodo, Objective C utilizza la sintassi:

```
[oggetto metodo];
```

Tale operazione è considerata non tanto come la chiamata ad una “funzione”, quanto l’invio di un messaggio ad un’entità. In quest’ottica è possibile reinterpretare la sintassi in questo modo:

```
[ricevente messaggio];
```

Il ricevente può essere un’istanza di una classe o la classe stessa, nel primo caso il metodo che si vuole invocare è un “metodo di istanza” nel secondo caso un “metodo di classe”.

Per esempio:

```
[mioArray inserisciOggetto:obj allaPosizione:3];
```

“mioArray” è l’oggetto su cui viene eseguito il metodo “inserisciOggetto”; “obj” è l’oggetto da inserire all’interno dell’array e “allaPosizione” è la posizione in cui deve essere inserito l’oggetto, in questo caso la 3.

L’Objective C consente inoltre di nidificare più chiamate.

Utilizzando tale tipo di approccio è possibile inviare messaggi ad un oggetto anche se esso non è capace di rispondere. Questo differisce dai linguaggi tipizzati staticamente come C++ e Java nei quali tutte le chiamate devono essere di metodi predefiniti.

Dato che questo linguaggio permette l’invio di un messaggio ad un oggetto che potrebbe non rispondere ad esso, quest’ultimo può gestirlo in altri modi, uno dei quali potrebbe consistere nel suo inoltro (forwarding) ad un’altra entità che sia in grado di rispondere.

In realtà la classica notazione con il punto può essere utilizzata per accedere alle proprietà. Ad esempio, queste righe di codice:

```
int valore = [oggetto proprieta];
```

```
[oggetto setProprieta:valore];
```

sono equivalenti a queste:

```
int valore = oggetto.proprieta;
```

```
oggetto.proprieta = valore;
```

La dot notation facilita il passaggio alla programmazione Objective C da chi proviene da linguaggi “C like”.

6.3 Le variabili di istanza

Ogni oggetto ha un suo stato interno rappresentato dalle variabili di istanza che per default non possono essere modificate né lette da oggetti esterni.

Le variabili di istanza possono essere dei tipi scalari del linguaggio C (int, float, decimal ecc.) oppure altri oggetti, anche classi definite dal programmatore.

La sintassi per la dichiarazione è uguale a quella usata in C:

```
tipo nomeVariabile;
```

Inoltre, allo stesso modo linguaggio C, anche in Objective C tutti gli oggetti sono riferiti tramite puntatori e quindi per definirli si antepone al loro nome un asterisco (*).

Per realizzare l'incapsulamento nei linguaggi orientati agli oggetti sono stati introdotti i qualificatori, che determinano la visibilità delle variabili definite all'interno delle classi.

I livelli di visibilità in Objective C sono 3:

public : rende la variabile visibile da qualunque oggetto;

private : rende la variabile utilizzabile solo all'interno dei metodi della stessa classe;

protected : alla variabile possono accedere solo i metodi della stessa classe e quelli delle classi da essa derivati.

Le variabili di istanza per default sono protected, quindi accessibili dalle sottoclassi; per indicare la visibilità è sufficiente far precedere la dichiarazione della variabile dalla relativa parola chiave, anticipata a sua volta da un carattere chiocciola ().

6.4 Protocolli e delegati

Un protocollo dichiara metodi che possono essere implementati da ogni classe, ma non possono essere considerati come tali, infatti essi definiscono semplicemente un'interfaccia che altri oggetti devono poi sviluppare, anche senza dichiararlo esplicitamente.

Quando un oggetto od una classe implementano i metodi di un protocollo, si dice che sono conformi a tale protocollo.

La dichiarazione di un protocollo è molto simile a quella di una classe:

```
@protocol Protocollo1
- (void) mioProtocollo;
@end
```

Un delegato invece è un oggetto che agisce in coordinamento o per conto di un altro: un'entità invia messaggi ad un altro oggetto specificato come suo delegato per gestire l'evolversi dell'esecuzione.

I delegati sono molto utilizzati nella gestione delle interfacce grafiche delle varie applicazioni: si crea un'istanza di un controller delegato a gestire una particolare vista, quindi con il compito ad esempio di informare l'applicativo che durante l'esecuzione l'utente utilizzatore ha generato un particolare evento.

6.5 Le proprietà

L'uscita di Objective C 2.0, nel 2006, ha portato delle novità, la più importante delle quali, oltre alla garbage collection, è stata l'introduzione delle proprietà (property), già citate nel documento.

Mentre precedentemente le variabili d'istanza richiedevano metodi espliciti di lettura e scrittura per essere utilizzate dal mondo esterno (incapsulamento), detti getters e setters, con questa nuova funzionalità è possibile procedere in modo diverso: la sintassi si compone di due parti, una da inserire nel file di interfaccia l'altra da scrivere nel file di implementazione.

Sarà il compilatore poi a sostituire le dichiarazioni con i metodi getters e setters secondo le indicazioni del programmatore.

Nel file di interfaccia le proprietà di una classe vanno dichiarate ad esempio nel seguente modo:

```
@property (assign, readwrite) int nomeVariabile;
```

Nella parte tra parentesi tonde vanno inserite le informazioni necessarie al compilatore per scrivere correttamente i metodi della proprietà, ad esempio per garantire i permessi di lettura, scrittura od entrambe.

Per completare la definizione dei parametri è necessario specificare nel file di implementazione l'istruzione:

```
@synthesize nomeVariabile;
```

Se viene omessa questa parte nel file di implementazione, il compilatore non genera automaticamente i metodi get e set.

6.6 Il costruttore

In Objective C il costruttore della classe è in realtà composto dalla chiamata di due diversi metodi annidati, dei quali il primo serve per l'allocazione della memoria alla classe, ed il secondo per l'inizializzazione delle variabili della stessa.

L'istanza di una classe generica avviene nel seguente modo:

```
Classe *istanza = [[Classe alloc] init];
```

Il metodo `init`, della classe `NSObject`, è quello che più somiglia al concetto di costruttore di Java, in esso si inizializzano tutte le variabili che utilizzerà la classe.

6.7 Super e self

`Super` è un riferimento alla superclasse dalla quale l'oggetto deriva ed è indispensabile quando è necessario creare un metodo di inizializzazione diverso da quello standard `init` o quando serve fare l'override di un metodo della superclasse.

Oltre a `super` ogni metodo riceve un parametro implicito detto `self`, che rappresenta un riferimento all'oggetto corrente.

L'oggetto `self` è utilizzato quando si vuole mandare un messaggio (invocare un metodo) interno alla classe stessa oppure quando si intende modificare il valore di una variabile di istanza.

6.8 Override ed overload

In Objective C è possibile effettuare l'override di un metodo definito nella superclasse, semplicemente creandone uno con lo stesso nome e gli stessi parametri: all'interno di questo metodo è possibile riferirsi a quello della superclasse attraverso l'oggetto `super`.

Non è possibile invece effettuare l'overload dei metodi, ma è consentito agire in questo modo:

```
-(id) init;
```

```
-(id) init:(tipo1) variabile1 andAltraVariabile:(tipo2) variabile2;
```

Anche se questi due metodi sembrano uguali, in realtà non lo sono: il primo ha un selettore (`init:`), il secondo invece un altro (`init: andAltraVariabile:`).

Qualora la classe implementi più metodi per l'inizializzazione, ad esempio, occorre sempre effettuare l'override di quello senza parametri, per evitare che l'utente possa creare la nostra classe in modo anomalo utilizzando il metodo `init` della superclasse. Quando viene effettuato l'override di tale metodo è possibile anche richiamare uno dei metodi `init` più specifico, passando dei valori di default.

6.9 L'oggetto ID

Utilizzando questo linguaggio di programmazione non è possibile dichiarare un oggetto come istanza di una interfaccia ma si può utilizzare l'oggetto generico 'id' di Objective C: a quest'ultimo può essere assegnata un'istanza di qualsiasi classe e su questo può essere richiamato un qualsiasi metodo (o inviato qualunque messaggio). Il metodo chiamato viene determinato dal runtime di Objective C in fase di esecuzione: se l'oggetto lo implementa non ci saranno problemi, altrimenti il programma terminerà inaspettatamente.

Le proprietà dell'oggetto universale id (assegnare all'oggetto qualsiasi istanza e richiamare da esso un qualsiasi metodo), rendono superfluo l'utilizzo contemporaneo di interfacce o ereditarietà, l'unico rischio è quello di perdere il controllo dei tipi del compilatore in quanto il tipo verrà controllato solo a runtime.

6.10 Gestione della memoria

La gestione della memoria è di fondamentale importanza per applicazioni dedicate a dispositivi mobili come iPad: è essenziale cercare di evitare qualsiasi spreco di risorse, dato che in questo tipo di periferiche sono limitate.

Ogni variabile, puntatore od oggetto di un programma scritto in Objective C occupa memoria, e questa deve essere allocata in quantità opportuna: come visto in precedenza, per creare ed inizializzare ad esempio un'istanza di una classe si utilizzano i due metodi `alloc` ed `init` annidati.

Il problema non si limita però alla sola allocazione della memoria, infatti è di fondamentale importanza liberarla non appena possibile, affinché possa essere adoperata per altri scopi.

Esiste un metodo che, come indicato dal nome (`dealloc`), serve proprio a tale scopo: questo però non deve mai essere invocato in modo diretto. Infatti agendo in questo modo l'oggetto diventerebbe inutilizzabile, anche nel caso in cui esso serva ancora durante l'esecuzione del programma, ad esempio se è riferito da qualche altra entità.

Nasce l'esigenza quindi di avere un contatore che indichi al software se l'oggetto è ancora utilizzato: tale valore deve essere posto ad uno non appena questo viene allocato in memoria, dovrà quindi essere incrementato di uno ogni volta che un'altra entità ha interesse a mantenerlo attivo, e diminuito di una unità invece quando un oggetto ne ha esaurito l'utilizzo.

Il contatore viene mantenuto in una proprietà chiamata `retainCount`, presente in ogni oggetto derivante dalla classe `NSObject`: nel momento in cui `retainCount` assume il valore zero, il metodo `dealloc` viene invocato in automatico e la memoria liberata.

Per diminuire il valore del contatore deve essere invocato nel codice il metodo `release`.

Parte II

Sviluppo di un'applicazione gestionale: iOrder

L'obiettivo della seconda parte del tirocinio è stato quello di sviluppare un'applicazione per piattaforma Apple iPad destinata ad agenti di vendita di una qualsiasi azienda.

Questo software è stato pensato per rendere più semplici, efficaci e gestibili le operazioni di esposizione degli articoli, acquisizione e conferma degli ordini presso i clienti, ed agevolare la sincronizzazione dei risultati della giornata lavorativa nel database aziendale una volta rientrati in sede.

Si tratta di un applicativo completamente nuovo, sviluppato, a seguito di una fase di raccolta dei requisiti e progettazione, seguendo le indicazioni e le richieste impartite dal tutore aziendale che è stato il committente dell'applicativo.

Il lavoro si è articolato quindi in quattro sezioni principali:

Raccolta ed analisi dei requisiti : necessaria per capire e studiare con attenzione tutte le caratteristiche e le funzionalità da integrare nel programma;

Progettazione della base di dati : indispensabile per gestire le collezioni di dati che l'applicativo andrà ad utilizzare;

Progettazione dell'interfaccia utente : cruciale per garantire all'utilizzatore funzionalità e facilità d'uso, ed allo stesso tempo sfruttare nel miglior modo possibile le dimensioni dello schermo del dispositivo per rappresentare le informazioni;

Implementazione del software : stesura del codice e realizzazione concreta dell'intero lavoro di progettazione.

Capitolo 7

Raccolta ed analisi dei requisiti

7.1 Specifiche di massima

IL SOFTWARE IN QUESTIONE è pensato per un'azienda che ha un certo numero di agenti di vendita, i quali devono recarsi presso i clienti per acquisire degli ordini di acquisto.

L'agente in questione deve avere la possibilità di consultare il catalogo degli articoli disponibili con i prezzi del cliente specifico e di selezionare i prodotti eventualmente desiderati dal cliente con le relative quantità.

Nel momento in cui l'acquirente ha terminato di ordinare la merce e viene chiuso l'ordine, l'agente può stampare o inviare via e-mail la conferma dell'operazione al cliente stesso ed anche all'azienda.

Una volta ritornato in sede, all'agente deve essere consentito di scaricare gli ordini raccolti nel gestionale aziendale.

7.2 Analisi in dettaglio: requisiti dell'applicazione

Il primo fattore di importanza fondamentale, entrando più in profondità nelle specifiche, riguarda l'utilizzatore del software: questo è destinato esclusivamente all'agente di vendita, il cliente può infatti solamente visionare gli articoli disponibili nel catalogo.

Una volta giunto dal compratore, l'incaricato dell'azienda ha il compito di aprire per esso un nuovo ordine di acquisto: può essere attivo un solo ordine alla volta e per iniziarne un altro deve prima essere chiuso quello corrente.

Gli articoli disponibili nel catalogo vengono sincronizzati dal database aziendale, non esiste l'opportunità di inserimento manuale; possono essere mostrati al cliente tramite una ricerca per vari criteri, tra cui categoria, marca e descrizione degli stessi.

Quando l'ordine in corso viene chiuso, è salvato nello storico: si richiede la creazione inoltre di un fincato dello stesso in formato PDF, da inviare oppure stampare in duplice copia, per l'azienda e l'acquirente.

L'applicazione deve consentire al suo utilizzatore di visionare tutte le righe d'ordine ed il dettaglio delle stesse, sia per l'ordine in corso che per quelli già chiusi.

Gli ordini raccolti, con tutti gli articoli al loro interno, vanno aggiornati nel database aziendale.

I valori di iva, dei tipi di trasporto e di pagamento da integrare nell'ordine devono essere visibili all'agente, che li deve impostare prima della chiusura dello stesso; è richiesta inoltre la presenza di un'anagrafica degli agenti che lavorano per la ditta; anche tutti questi dati vengono inseriti nella periferica tramite sincronizzazione con il gestionale dell'impresa.

I clienti generalmente vengono sincronizzati dal gestionale aziendale, ma un agente ha la facoltà di aggiungere un nuovo cliente in caso di bisogno.

Gli aggiornamenti del database presente in locale su iPad ed in remoto in quello dell'azienda, possono essere effettuati per ogni singola tabella, oppure per tutte contemporaneamente.

Viene presentata la logica principale di funzionamento dell'applicazione a diversi livelli di astrazione, tramite rappresentazione con diagrammi UML:

Diagramma di sequenza: descrive le interazioni tra le entità del progetto mettendo in evidenza l'avanzamento temporale

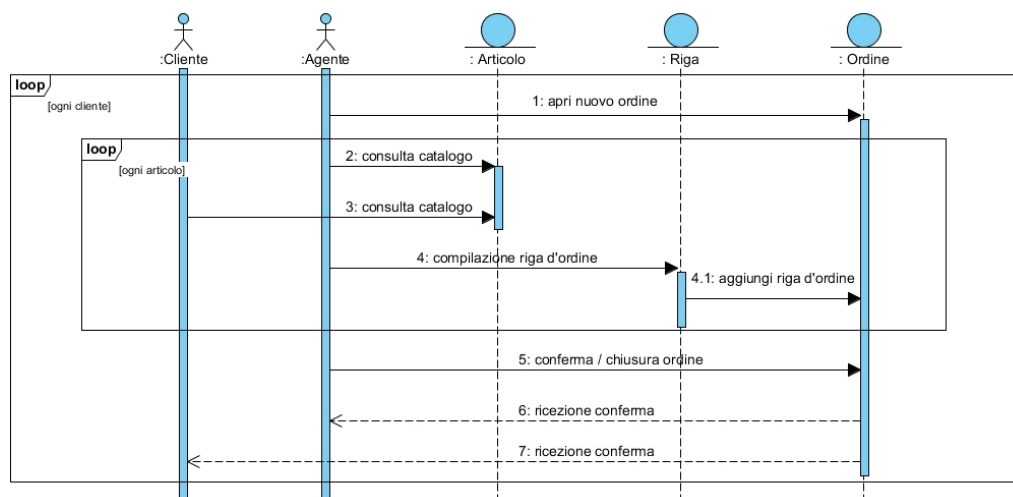


Figura 7.1: Diagramma di sequenza

Diagramma dei casi d'uso: rappresenta le modalità d'uso dell'applicativo da parte dei fruitori, esprime quindi le interazioni tra il sistema e la componente umana evidenziando quali funzionalità gli attori possono richiedere al software

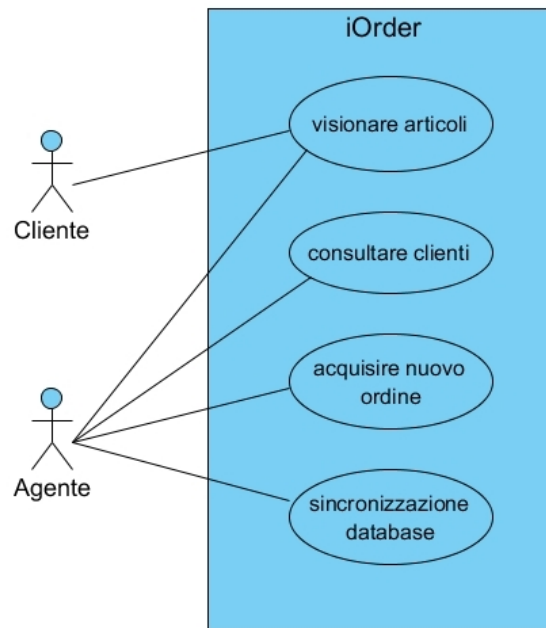


Figura 7.2: Diagramma dei casi d'uso

Diagramma delle attività (con swimlane): rappresenta la logica procedurale dell'esecuzione del software, mettendo in evidenza, oltre all'andamento temporale, anche chi compie determinate azioni

Requisiti dei dati dell'applicazione

Dopo una presentazione dei requisiti, vengono descritte tutte le proprietà che devono essere rappresentate per ogni particolare entità che andrà a far parte dell'applicazione.

- Per un cliente, identificato univocamente da un codice, si rappresentano la ragione sociale, la partita iva, il codice fiscale, l'indirizzo, il CAP, la località, la provincia ed il paese di residenza, il telefono, il fax, l'e-mail, uno sconto cliente, il referente ed il codice di pagamento. Un cliente può essere associato ad una modalità di pagamento. Un cliente può effettuare più ordini.
- Per un articolo, identificato univocamente da un codice, si rappresentano il codice a barre, la descrizione, le specifiche, la marca, la categoria, l'unità di misura, la valuta, il prezzo netto, la quantità disponibile, la data e l'ora a cui tale numero si riferisce e la foto che lo rappresenta. La quantità degli articoli disponibile è riferita ad una precisa data e ora. Uno stesso articolo può essere presente in più righe d'ordine. Un articolo è associato ad una sola categoria. Un articolo è associato ad una sola marca.

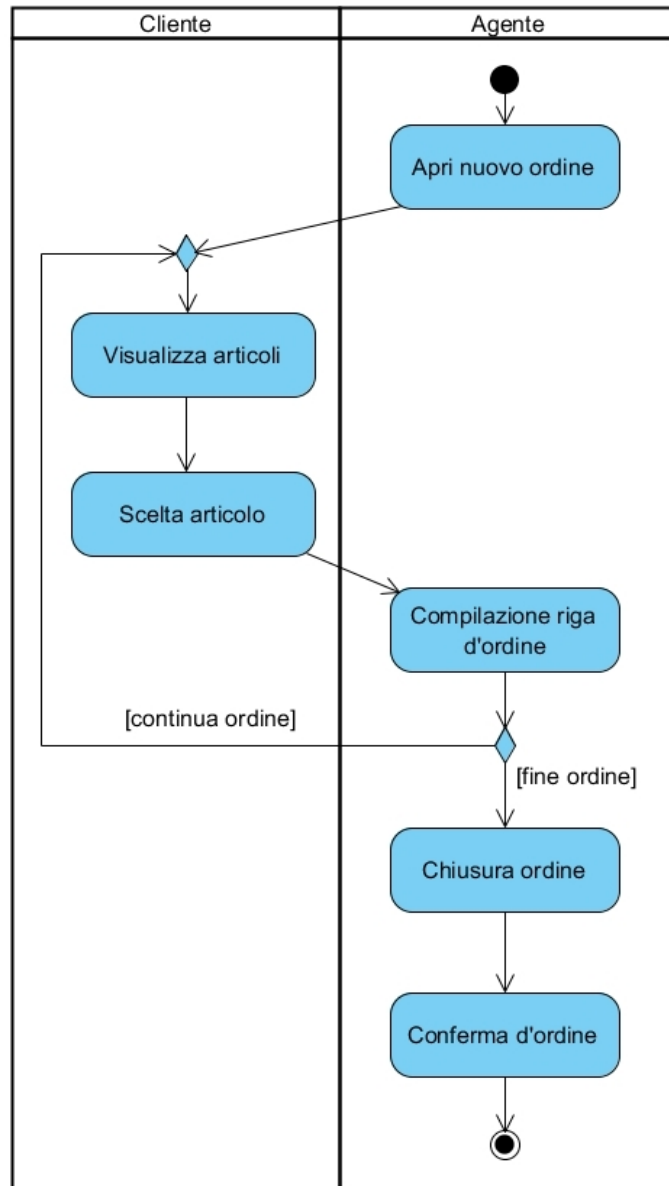


Figura 7.3: Diagramma delle attività (con swimlane)

- Per un ordine, identificato univocamente da tipo, anno e numero dello stesso, si rappresentano la data, il referente, lo sconto del cliente, l'agente, il totale netto, il valore totale dell'iva, il prezzo totale, i dati anagrafici del cliente, la descrizione del pagamento e la modalità di trasporto. Un ordine può contenere più righe d'ordine.

Uno specifico ordine è ricevuto da un solo agente. Ad un ordine sono associate una sola modalità di trasporto ed una sola modalità di pagamento.

- Per una riga d'ordine, identificata univocamente da tipo, anno e numero dell'ordine a cui appartiene e dal suo numero all'interno dello stesso, si rappresentano il codice dell'articolo, la descrizione, l'unità di misura, la quantità, la data di consegna prevista, la valuta, il prezzo unitario, la percentuale di sconto, il prezzo netto, e l'iva. Una riga d'ordine può avere uno sconto indipendente da quello standard di un particolare cliente. Ad una riga d'ordine è associato un solo valore di iva. Una riga d'ordine può contenere un solo articolo. Una specifica riga d'ordine è contenuta all'interno di un solo ordine.
- Per un agente identificato univocamente da un codice, si rappresentano il cognome, il nome, il codice fiscale, l'indirizzo, il CAP, la località, la provincia ed il paese di residenza, il telefono e l'e-mail. Un agente può ricevere più ordini.
- Per un tipo di pagamento, identificato univocamente da un codice, si rappresenta una descrizione. Un particolare pagamento può essere associato a più clienti.
- Per un tipo di trasporto, identificato univocamente da un codice, si rappresenta una descrizione. Un particolare trasporto può essere associato a più ordini.
- Per un'iva, identificata univocamente da un codice, si rappresentano una descrizione e il valore dell'iva (in percentuale). Una particolare iva può essere associata a più righe d'ordine.
- Per una categoria di articolo, identificata univocamente da un codice, si rappresenta una descrizione. Una particolare categoria può essere associata a più articoli.
- Per una marca di articolo, identificata da un codice, si rappresenta un nome ed una descrizione. Una particolare marca può essere associata a più articoli.

Operazioni richieste

Vengono di seguito specificate le operazioni da implementare durante lo sviluppo del software: di fianco ad ognuna viene precisata la frequenza con la quale verranno effettuate secondo le richieste del committente.

Operazione	Frequenza
Visualizzare anagrafica articoli	(500 volte al giorno)
Visualizzare anagrafica clienti	(10 volte al giorno)
Visualizzare storico ordini	(1 volta al giorno)
Visualizzare composizione di un ordine (tutte le righe d'ordine)	(50 volte al giorno)
Ricerca di un articolo data la categoria	(300 volte al giorno)
Ricerca di un articolo data la descrizione	(50 volte al giorno)
Ricerca di un articolo data la marca	(150 volte al giorno)
Inserimento nuovo cliente	(1 volta al mese)
Inserimento nuovo ordine	(10 volte al giorno)
Sincronizzazione articoli con il database aziendale	(1 volta alla settimana)
Sincronizzazione ordini con il database aziendale	(1 volta al giorno)
Sincronizzazione agenti con il database aziendale	(1 volta all'anno)
Sincronizzazione clienti con il database aziendale	(5 volte al mese)
Sincronizzazione tipi di pagamento con il database aziendale	(1 volta all'anno)
Sincronizzazione tipi di trasporto con il database aziendale	(1 volta all'anno)
Sincronizzazione valori iva con il database aziendale	(1 volta all'anno)

Tabella 7.1: Lista delle operazioni con frequenza

Capitolo 8

Progettazione della base di dati

IL PASSO SUCCESSIVO all'analisi dettagliata dei requisiti dell'applicazione riguarda l'elaborazione della base di dati che verrà utilizzata, e quindi l'organizzazione della struttura, delle caratteristiche ed del contenuto.

Una metodologia di progettazione per un database consiste nella decomposizione del piano di sviluppo in fasi successive ed indipendenti tra loro, in strategie da usare nelle varie fasi e dei criteri da adottare in caso di scelta oltre ai modelli di riferimento per descrivere i dati utilizzati nei vari stadi.

Devono essere garantite anche delle proprietà, le quali dimostrino la correttezza del lavoro svolto, come ad esempio la generalità rispetto all'applicazione usata, la qualità del prodotto ottenuto per quel che riguarda completezza, correttezza ed efficienza ed anche la facilità d'uso dei modelli di riferimento.

La metodologia di progetto che meglio soddisfa questi requisiti consiste in tre fasi principali, da eseguire una dopo l'altra e necessarie per distinguere nettamente che cosa rappresentare nella base di dati da come farlo nella realtà: queste fasi sono denominate nell'ordine progettazione concettuale, progettazione logica e progettazione fisica.

Le prime due fasi vengono sviluppate di seguito, mentre la progettazione fisica non fa parte della trattazione.

8.1 Progettazione concettuale

La progettazione concettuale di una base di dati ha lo scopo di rappresentare le specifiche del minimondo di interesse e l'organizzazione dei dati ad un alto livello di astrazione, molto utile a scopo documentativo.

Questa fase è indipendente dagli aspetti implementativi e di efficienza dello specifico sistema di gestione di basi di dati scelto; il prodotto ottenuto fa riferimento ad un modello concettuale dei dati ed è chiamato schema concettuale (schema Entità-Relazione).

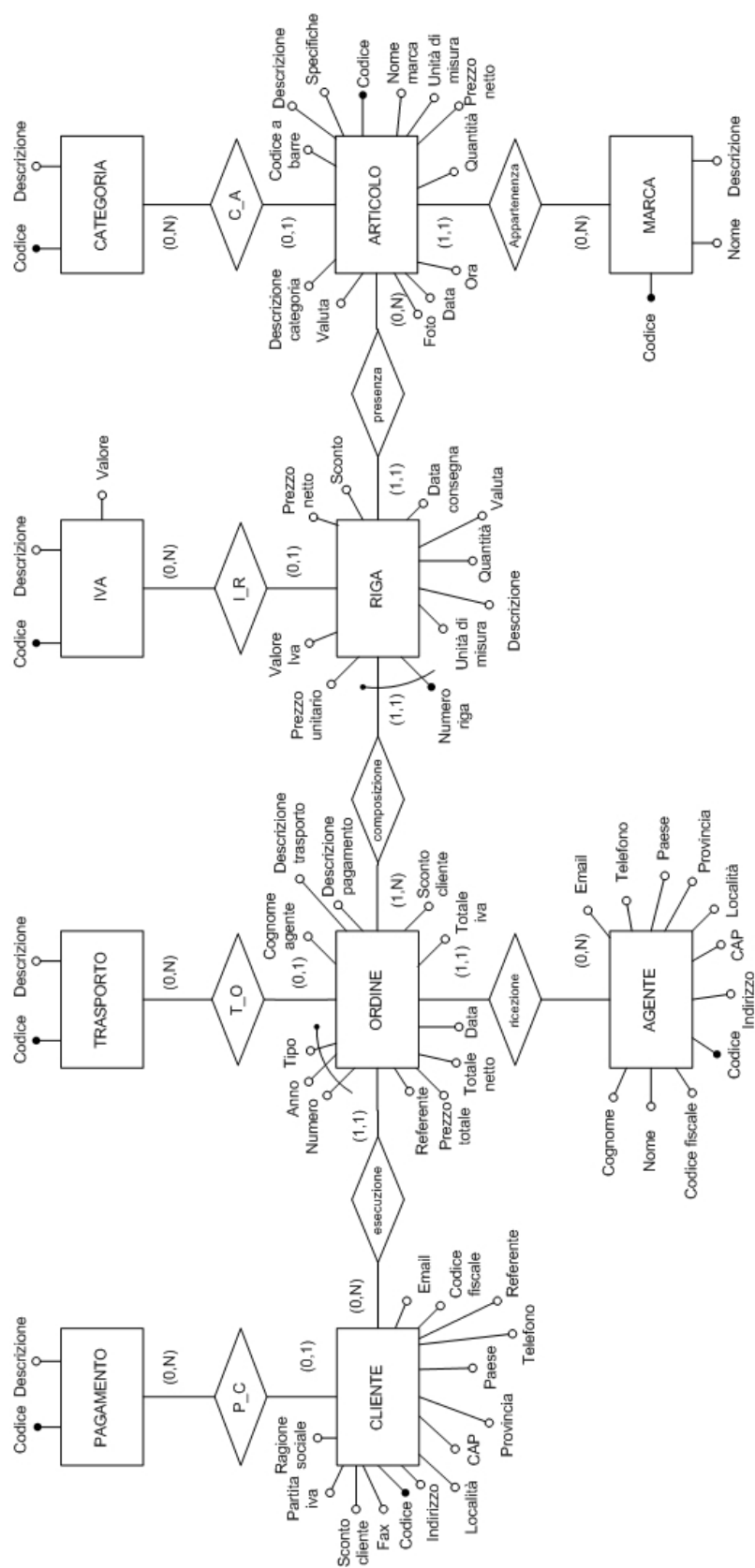


Figura 8.1: Schema concettuale Entità-Relazione

8.1.1 Dizionario dei dati

Entità

Nel modello concettuale sono presenti vari attributi che potrebbero cambiare nel tempo e quindi servono per mantenere i dati di una particolare entità come storici: sono validi nel momento in cui vengono inseriti nella base di dati ma possono subire variazioni nell'occorrenza originaria.

Tali attributi a cui si fa riferimento sono:

- ScontoCliente, Referente, DescrizionePagamento, DescrizioneTrasporto, CognomeAgente, PartitaIva, RagioneSociale, Indirizzo, CAP, Località, Provincia e Paese nell'entità Ordine;
- UnitàMisura, Descrizione, Valuta, PrezzoNetto e ValoreIva nell'entità Riga.
- DescrizioneCategoria e NomeMarca nell'entità Articolo.

Associazioni

(Vedi tabella 8.2)

8.1.2 Regole di vincolo

Rv1 – l'entità Ordine DEVE rappresentare anche i dati anagrafici del Cliente: oltre ai dati descritti nello schema E-R sono presenti anche gli attributi PartitaIva, RagioneSociale, Indirizzo, CAP, Località, Provincia e Paese

Rv2 - l'attributo Valore dell'entità Iva NON DEVE essere nullo.

8.1.3 Regole di derivazione

Rd1 - L'attributo PrezzoUnitario dell'entità Riga SI OTTIENE dall'attributo PrezzoNetto, a cui viene sommato il prodotto tra gli attributi PrezzoNetto e ValoreIva fratto 100 (il valore effettivo dell'iva) e poi sottratto il prodotto tra gli attributi PrezzoNetto e Sconto fratto 100 (il valore effettivo dello sconto). Tutti questi attributi sono appartenenti all'entità Riga.

Rd2 - L'attributo TotaleNetto dell'entità Ordine SI OTTIENE dal prodotto tra gli attributi PrezzoNetto e Quantità, a cui vengono sottratti il prodotto tra gli attributi PrezzoNetto e Sconto fratto 100 (il valore effettivo dello sconto della particolare riga) ed ancora il prodotto tra gli attributi PrezzoNetto e ScontoCliente fratto 100 (il valore effettivo dello sconto del cliente specifico); tutti questi valori calcolati e quindi sommati per ogni riga d'ordine.

Entita'	Descrizione	Attributi	Identificatore
PAGAMENTO	Informazioni relative ad un tipo di pagamento	Codice, Descrizione	Codice
CLIENTE	Informazioni relative ad un cliente	Codice, RagioneSociale, PartitaIva, CodiceFiscale, Indirizzo, CAP, Localita, Provincia, Paese, Telefono, Fax, Email, ScontoCliente, Referente	Codice
ORDINE	Informazioni relative ad un ordine	Tipo, Anno, Numero, Data, Referente, TotaleNetto, TotaleIva, PrezzoTotale, DescrizionePagamento, DescrizioneTrasporto, ScontoCliente, PartitaIva, RagioneSociale, Indirizzo, CAP, Località, Provincia, Paese, CognomeAgente	Tipo, Anno, Numero
AGENTE	Informazioni relative ad un agente	Codice, CodiceFiscale, Cognome, Nome, Indirizzo, CAP, Localita, Provincia, Paese, Telefono, Email	Codice
TRASPORTO	Informazioni relative ad un tipo di trasporto	Codice, Descrizione	Codice
RIGA	Informazioni relative ad una riga d'ordine	TipoOrdine, AnnoOrdine, NumeroOrdine, NumeroRiga, Descrizione, UnitàMisura, Quantità, Valuta, PrezzoUnitario, DataConsegna, Sconto, PrezzoNetto, ValoreIva	TipoOrdine, AnnoOrdine, NumeroOrdine, NumeroRiga
IVA	Informazioni relative ad un valore di iva	Codice, Descrizione, Valore	Codice
ARTICOLO	Informazioni relative ad un articolo in vendita	Codice, CodiceBarre, Descrizione, Specifiche, UnitàMisura, Valuta, PrezzoNetto, Quantità, Data, Ora, Foto, DescrizioneCategoria, NomeMarca	Codice
CATEGORIA	Informazioni relative ad una categoria di articolo	Codice, Descrizione	Codice
MARCA	Informazioni relative alla marca di un articolo	Codice, Nome, Descrizione	Codice

Tabella 8.1: Lista delle entità

Associazione	Descrizione	Attributi	Entità collegate
P_C	Collega un tipo di pagamento ad un cliente	-	Pagamento(0,N), Cliente(0,1)
Esecuzione	Collega un cliente ad un ordine	-	Cliente(0,N), Ordine(1,1)
T_O	Collega un tipo di trasporto ad un ordine	-	Trasporto(0,N), Ordine(0,1)
Ricezione	Collega un agente ad un ordine	-	Ordine(1,1), Agente(0,N)
Composizione	Collega un ordine ad una riga d'ordine	-	Ordine(1,N), Riga(1,1)
I_R	Collega un valore d'iva ad una riga d'ordine	-	Iva(0,N), Riga(0,1)
Presenza	Collega una riga d'ordine ad un articolo	-	Riga(1,1), Articolo(0,N)
C_A	Collega una categoria ad un articolo	-	Categoria(0,N), Articolo(0,1)
Appartenenza	Collega una marca ad un articolo	-	Articolo(1,1), Marca(0,N)

Tabella 8.2: Lista delle associazioni

Rd3 - L'attributo TotaleIva dell'entità Ordine SI OTTIENE dallo stesso calcolo necessario a calcolare l'attributo TotaleNetto, con la differenza che in questo caso si sostituisce il valore PrezzoNetto utilizzato in precedenza con il prodotto tra gli attributi PrezzoNetto e ValoreIva fratto 100 (il valore effettivo dell'iva) per ogni riga d'ordine.

Rd4 - L'attributo PrezzoTotale dell'entità Ordine SI OTTIENE come somma tra l'attributo TotaleNetto e TotaleIva, sempre dell'entità Ordine.

8.2 Progettazione logica

La progettazione logica di un database consiste nella traduzione dello schema Entità-Relazione ottenuto dalla progettazione concettuale nel modello di rappresentazione dei dati adottato dal sistema di gestione di basi di dati scelto.

A questo livello i dati vengono descritti in modo indipendente da dettagli fisici, ma condizionato dallo specifico DBMS: il prodotto di questa fase di progettazione fa riferimento ad un modello logico dei dati (modello relazionale) ed è denominato schema logico.

Si procede prima ristrutturando lo schema E-R derivato dalla progettazione concettuale, analizzando eventuali ridondanze e generalizzazioni presenti oltre alla possibilità di dividere od unire alcuni concetti, e poi traducendo lo stesso per ottenere lo schema logico.

8.2.1 Volumi dei dati

Seguendo le indicazioni del committente si fa riferimento ad una situazione media giornaliera, nella quale si suppone che un agente raccolga circa dieci ordini comprendenti ognuno

Concetto	Tipo	Volume
Cliente	E	2000
Ordine	E	10
Articolo	E	50000
Riga	E	500
Agente	E	10
Pagamento	E	10
Trasporto	E	10
Iva	E	10
Categoria	E	10
Marca	E	100
P_C	R	2000
Esecuzione	R	10
T_O	R	10
Ricezione	R	10
Composizione	R	500
I_R	R	500
Presenza	R	500
C_A	R	50000
Appartenenza	R	50000

Tabella 8.3: Volumi dei dati

cinquanta righe d'ordine.

8.2.2 Analisi delle ridondanze

Nello schema E-R sono presenti vari attributi ridondanti: TotaleNetto, TotaleIva e Prezzo-Totale dell'entità Ordine e PrezzoUnitario dell'entità Riga.

Questi sono dati calcolati a partire da altri attributi della stessa o di altre entità, nel modo specificato nelle regole di derivazione della progettazione concettuale, ma dato l'uso di tale base di dati principalmente come storico, su precisa indicazione del committente si è deciso per utilità e convenienza di mantenerli all'interno del database.

Nello schema E-R originale non sono presenti generalizzazioni e non si ritengono necessari partizionamenti o accorpamenti di concetti, per cui lo schema E-R ristrutturato non subisce alcuna variazione.

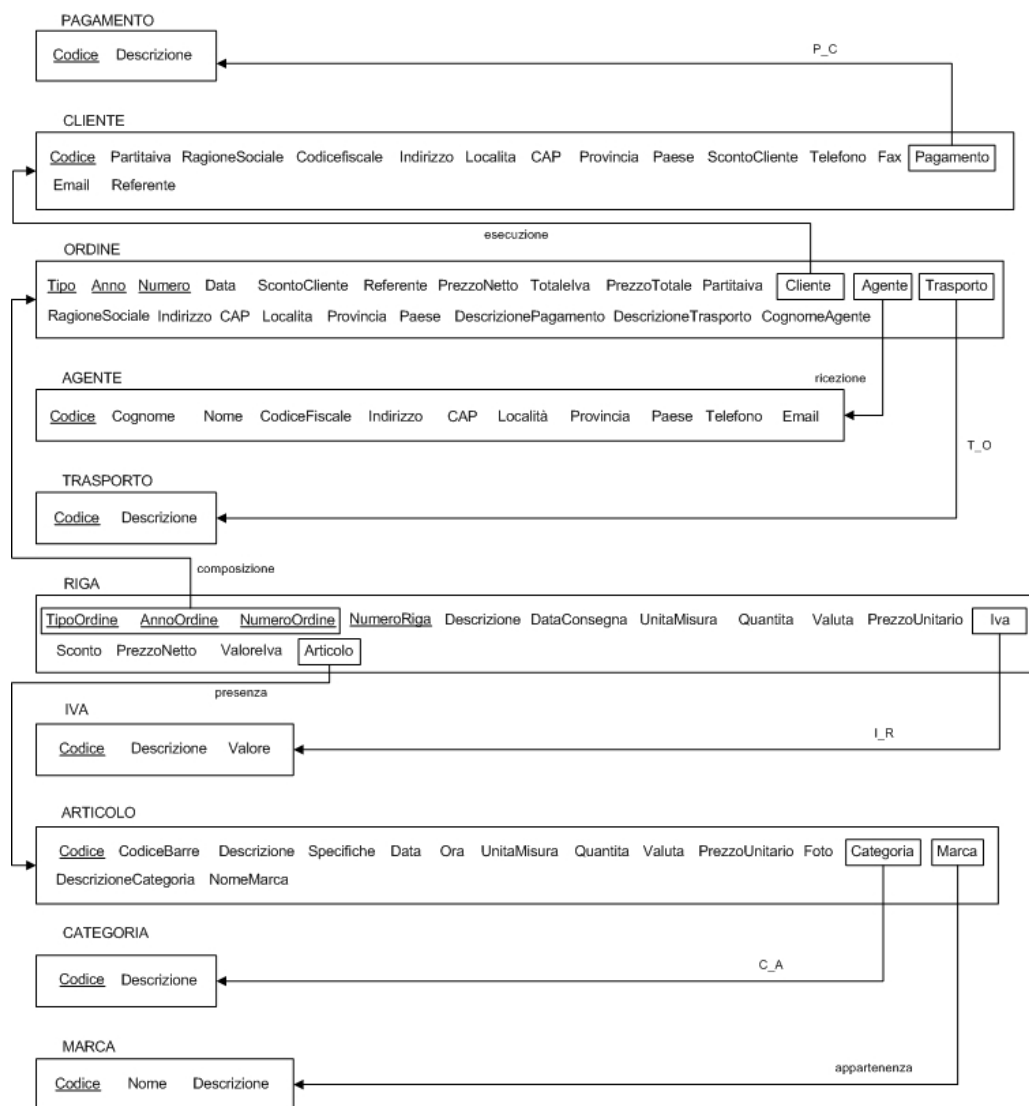


Figura 8.2: Schema logico

Capitolo 9

Progettazione dell'interfaccia utente

IN UN'APPLICAZIONE DESTINATA ad un dispositivo portatile come iPad, il modo in cui le informazioni vengono esibite all'utilizzatore ricopre un ruolo fondamentale, e cercare di presentare i dati gestiti dal software nel miglior modo possibile e completo è stato l'obiettivo prefissato per il progetto iOrder.

Il kit di sviluppo software destinato a questa periferica ha introdotto dei componenti grafici di nuova concezione per sfruttare al meglio l'aumento di superficie dello schermo rispetto a quanto fosse precedentemente possibile con iPhone.

Grazie ad una di queste novità, chiamata *split view*, è stato possibile progettare un'interfaccia su diversi livelli, mantenendo a sinistra una sezione statica con l'indicazione della particolare funzionalità in uso, ed invece a destra una porzione dinamica del contenuto con il dettaglio della selezione effettuata nella prima parte oppure con altre finestre da esso derivate.

Mantenere una tale disposizione delle informazioni garantisce all'utilizzatore del software una maggiore usabilità, efficacia ed efficienza rispetto ad una rappresentazione tradizionale a singola finestra: consente infatti di avere sempre la cognizione precisa del punto in cui si trova, e permette un passaggio rapido tra viste che gestiscono diverse funzionalità, evitando perdite di tempo con la navigazione tra finestre intermedie, irrilevanti in quel momento.

Questo tipo di interfaccia permette inoltre all'utente facilità di apprendimento e di ricordo, consentendo il raggiungimento di buone prestazioni in tempi brevi e la capacità di interagire con il programma in modo semplice anche dopo un lungo periodo di inutilizzo.

Oltre alla *split view* che è il componente più in vista dell'interfaccia grafica, ne sono stati utilizzati altri per lo sviluppo del software, insieme alle linee guida per gli sviluppatori suggerite da Apple.

L'azienda infatti invita chi desidera sviluppare applicazioni per i propri dispositivi portatili a seguire delle convenzioni, in modo da rappresentare le informazioni e gestirle in stile tipicamente Apple: uno standard che facilita il passaggio da un programma ad un altro

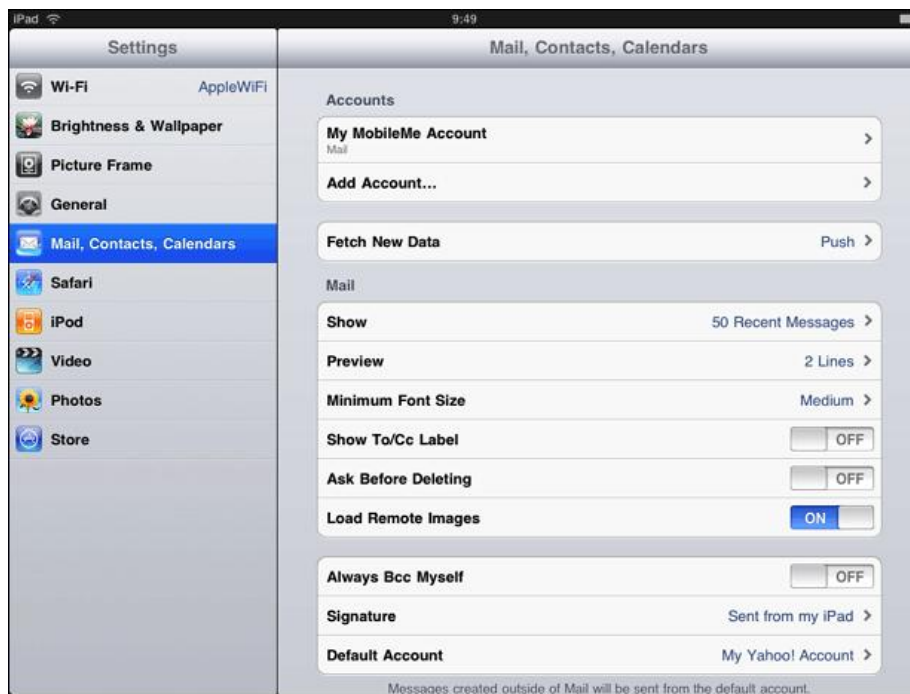


Figura 9.1: Split view

senza alcuna difficoltà, data l'omogenea sistemazione dei dati e del significato delle icone che sono via via presentate all'utente.

Viene consigliato allo sviluppatore quali sono i componenti grafici che meglio si adattano ad una certa situazione, oppure come personalizzare meticolosamente il modo di presentare le informazioni per dare un particolare significato a ciò che viene scritto.

Un esempio di questo fatto è il modo in cui si consiglia di presentare degli avvisi all'utilizzatore usando gli alert view.



Figura 9.2: Alert view

Questo componente è il modo più rapido e semplice per dare importanti informazioni durante l'esecuzione dell'applicazione: può essere completamente personalizzato, e proprio per questo vengono spiegate tutte le modifiche possibili per rappresentare ciò che il pro-

grammatore vuole comunicare, e nel modo più intuitivo per l'utente. Sono presenti precise indicazioni sulla formulazione del titolo, del testo, sul numero di alternative concesse al destinatario del messaggio, in che ordine rappresentarle a video e che colori utilizzare per rappresentare diversamente ad esempio un avviso potenzialmente pericoloso da uno senza rischi.

Altro componente molto importante nella gestione della logica del software è la barra di navigazione, o navigation bar, posta in alto sullo schermo dell'iPad in uso.



Figura 9.3: Barra di navigazione

La navigation bar è utilizzata quando viene permesso all'utente di scendere di livello in una gerarchia di informazioni: all'inizio contiene il titolo della vista corrente al centro, ma quando l'utilizzatore ne fa apparire una nuova, "figlia" della precedente, la barra cambia il suo aspetto. Infatti il titolo al centro diventa quello della nuova finestra, e a sinistra compare un bottone di ritorno con il titolo della vista "padre", per poter risalire velocemente nella gerarchia.

Oltre al tasto di ritorno, una barra di navigazione può contenere un secondo bottone a destra del titolo e, se il primo non è necessario, se ne può aggiungere a sinistra un altro con diversa funzionalità.

Le linee guida vengono fornite da Apple anche per uniformare i bottoni che il programmatore può inserire per effettuare precise azioni, ed anche la particolare posizione che alcuni devono avere all'interno di una finestra.

Ad esempio nella barra di navigazione i bottoni di conferma vanno posti a destra, per tornare indietro tra le schermate o per annullare l'azione in corso devono invece essere situati a sinistra.

Anche la barra di ricerca è un componente grafico che è stato utilizzato nell'applicativo e specificatamente quella dotata di scope bar.

Nel programma è implementata la ricerca degli articoli in base a vari attributi, e proprio in base a questi può essere orientata: toccando la barra in questione appare a video la tastiera virtuale di iPad che consente di digitare il testo desiderato, la funzionalità aggiuntiva consente inoltre di scegliere lo specifico campo da ricercare cliccandolo tra quelli disponibili.

L'ultimo componente tra i principali utilizzati nell'applicazione, ma non il meno importante, è indispensabile per consentire all'utilizzatore la scelta tra un range limitato di possibilità fornite, senza permettere la digitazione di valori tramite tastiera: il ruolo che nei programmi per periferiche dotate di mouse è ricoperto dalle combobox, le quali in dispositivi touch come iPad non possono essere utilizzate data la minor precisione con cui gli utenti selezionano gli oggetti.



Figura 9.4: Barra di ricerca con scope bar

In realtà sono disponibili due diverse alternative di implementazione, il picker ed il datePicker: quest'ultimo permette di scegliere una precisa data od ora, mentre il primo consente la decisione più generale tra un insieme qualsiasi di possibilità fornite.

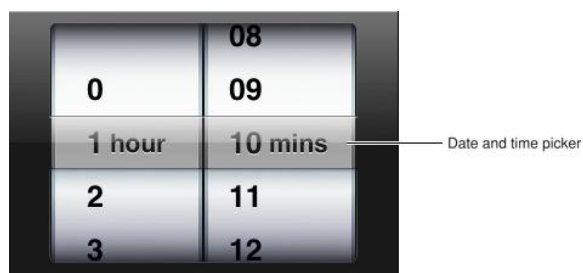


Figura 9.5: DatePicker

Per effettuare le proprie scelte l'utente deve far scorrere le rotelline (fino a quattro affiancate ed indipendenti) che contengono i valori in questione: quello evidenziato al centro è il dato scelto.

Capitolo 10

Sviluppo del progetto

NELLA SECONDA PARTE del tirocinio è stata messa in pratica tutta la fase di progettazione e studio svolta nel primo periodo per chiudere il ciclo di lavoro svolto presso l'azienda con lo scopo finale di questo progetto formativo: creare l'applicazione iOrder.

Per prima cosa è stata implementata la base di dati destinata a contenere le informazioni utili al corretto funzionamento del programma, il lavoro è poi continuato con la stesura del codice per la realizzazione del software.

10.1 Implementazione della base di dati

Lo sviluppo della base di dati utilizzata nel programma iOrder ha tenuto conto del dispositivo su cui avrebbe effettivamente funzionato e per questo, dato che iPad è una periferica portatile con risorse limitate, è stato scelto un DBMS relazionale che obbedisse agli standard SQL e che fosse il più possibile compatto, veloce e di semplice interpretazione.

Tali preferenze hanno portato alla scelta di SQLite, un RDBMS particolarmente adatto ai sistemi portatili, tanto da essere già utilizzato nel sistema operativo dell'iPhone per la gestione di SMS e MMS, per il calendario ed altre funzionalità.

SQLite è una libreria software scritta in linguaggio C che implementa un DBMS SQL, permette di creare una base di dati incorporata in un unico file ed è di dominio pubblico e quindi utilizzabile senza restrizioni.

Il database è stato creato utilizzando l'utility SQLite Manager, disponibile come componente aggiuntivo del browser Mozilla Firefox: questo ha permesso l'inserimento dei comandi SQL per creare le tabelle in modo semplice e con la visione immediata dei risultati.

Dopo la realizzazione il file con la base di dati è stato inserito in una cartella appositamente predisposta nel dispositivo, per poter essere utilizzato all'interno dell'applicazione.

La logica di lavoro, supponendo che le informazioni siano già presenti nel database locale, consiste nel prelevare i dati nel momento dell'avvio del programma e salvarli in array (uno

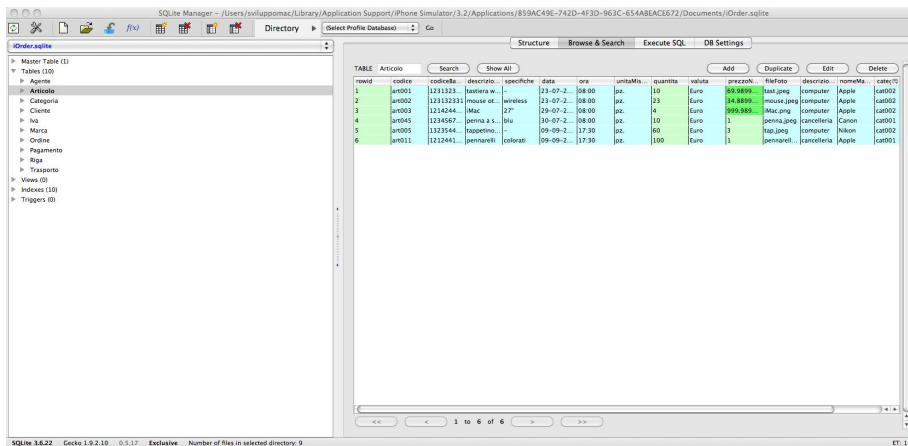


Figura 10.1: SQLite Manager

per ogni tabella) con dei metodi sviluppati appositamente, per poi gestire l'esecuzione con queste strutture dati senza più effettuare accessi alla base di dati.

Ad esempio per trasferire le informazioni dalla tabella “Cliente” del database all'array “listaClienti” utilizzato nel codice si usa il seguente metodo:

```
-(void)caricaClientiDaDb{
    if (sqlite3_open([writablePath UTF8String], &database) == SQLITE_OK) {
        // query che ricava i valori
        const char *sql = "select * from Cliente";
        sqlite3_stmt *caricaClienti;
        if(sqlite3_prepare_v2(database, sql, -1, &caricaClienti, NULL) ==
        SQLITE_OK) {
            while(sqlite3_step(caricaClienti) == SQLITE_ROW) {
                Cliente *cli = [[Cliente alloc] init];
                cli.codice= [NSString stringWithUTF8String:(char *)sqlite3_column
                _text(caricaClienti, 0)];
                cli.partitaIva=[NSString stringWithUTF8String:(char *)sqlite3_column
                _text(caricaClienti, 1)];
                ...
                cli.pagamento=[NSString stringWithUTF8String:(char *)sqlite3_column
                _text(caricaClienti, 14)];
                [listaClienti addObject:cli];
            }
        }
        else NSLog(@"Errore nel database: %s", sqlite3_errmsg(database));
        sqlite3_finalize(caricaClienti);
    }
}
```

```
    }  
    else  
        sqlite3_close(database);  
}
```

Nel caso in cui, durante l'esecuzione del programma, sorga la necessità di includere dei nuovi dati all'interno degli array, come avviene con l'inserimento di un nuovo cliente o di un nuovo ordine, è evidente il bisogno di aggiornare immediatamente anche la relativa tabella nel database e per questo un metodo simile a quello citato sopra, con le modifiche necessarie, viene richiamato per adempiere allo scopo.

10.2 Implementazione del codice

Come detto in precedenza iOrder è un applicativo nuovo, sviluppato a partire dai requisiti richiesti dal tutore aziendale, utilizzando una tecnologia nuova sia per il tirocinante che per la società.

Per prendere confidenza con gli strumenti di sviluppo ed il linguaggio di programmazione, anch'essi usati per la prima volta durante questo tirocinio, come attività iniziale sono stati sviluppati vari programmi di esempio, all'interno dei quali è stato possibile provare i componenti grafici utilizzati in seguito nell'applicativo principale.

Molto importante durante questa fase di prova è stato capire le funzionalità principali dell'ambiente di sviluppo Xcode e del software Interface Builder, riuscendo a farli interagire efficacemente nell'implementazione delle prime interfacce grafiche.

Fondamentale si è dimostrato il graduale apprendimento dell'utilizzo del debugger per trovare e risolvere gli errori che via via si manifestavano: con il passare del tempo si sono ridotti i tempi per la ricerca di lacune nella sintassi e della sistemazione dei difetti nella logica dei programmi.

Passo successivo a questa fase preparatoria, è stato lo studio delle varie viste da implementare e delle loro gerarchie: quali finestre rappresentare come principali, e quindi accessibili direttamente, e quali invece come loro figlie o nipoti, accessibili solamente dal relativo genitore.

La parte fissa dell'interfaccia utente, situata alla sinistra dello schermo, contiene 10 voci che rappresentano le viste non dipendenti da nessun'altra, ma che possono essere padre di qualche altra finestra.

L'architettura consente di capire la struttura generale dell'applicazione iOrder, che viste sono disponibili durante l'esecuzione del software, quali finestre dipendono da altre e sono accessibili tramite di esse.

In linea generale lo sviluppo è avvenuto finestra per finestra a partire dalla vista principale, seguendo un modello di processo evolutivo a spirale: ognuna di queste componenti è stata creata a partire da una fase di pianificazione e modellazione, seguita dalla costruzione e dal perfezionamento del risultato ottenuto. Il programma è andato formandosi con delle

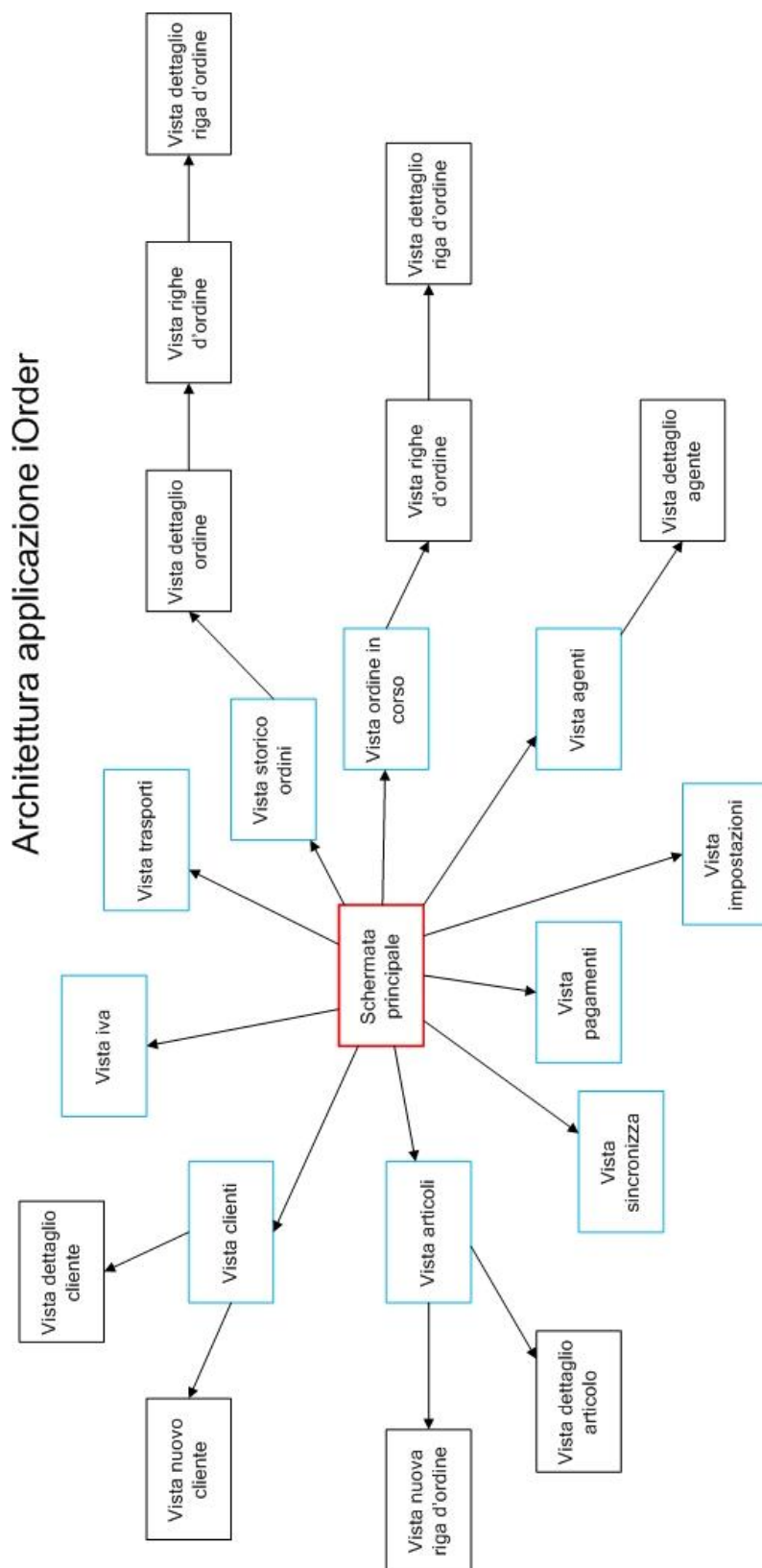


Figura 10.2: Architettura dell'applicazione iOrder

versioni via via più complete, arricchite dalle funzionalità che mano a mano sono state integrate.

Da tenere presente il fatto che, nonostante la possibilità di ruotare la presentazione dei contenuti dell'applicazione a seconda dell'orientamento dell'iPad, si è deciso per una precisa scelta progettuale di abilitare i cambiamenti in base ai valori dell'accelerometro solo nelle due modalità landscape, allineando cioè l'interfaccia utente sempre con il dispositivo in posizione orizzontale, e ruotando la visualizzazione solo nel caso in cui questo venga capovolto.

Per ogni vista è stata sviluppata una classe con la funzione di controller destinato a gestirne il comportamento, ed inoltre per ognuna delle entità da rappresentare sono stati creati degli oggetti per racchiudere e manipolare in modo migliore tutti gli attributi che le compongono.

Nella finestra di base, che è la prima a comparire al lancio del programma, la sezione situata a sinistra nello schermo, detta "master", viene presentata con le dieci voci principali, le funzionalità fondamentali messe a disposizione dell'utente.

Queste sezioni riguardano le anagrafiche degli articoli disponibili, dei clienti, degli ordini archiviati, degli agenti di vendita, delle modalità di trasporto e pagamento e dei possibili valori di iva; inoltre sono presenti le voci per accedere all'ordine in corso, se presente, alle sincronizzazioni con il database aziendale ed infine alle impostazioni del software.

All'avviamento dell'applicazione nessuna voce tra le precedenti risulta selezionata, e nell'area a destra sullo schermo, la vista "detail", non è rappresentata alcuna informazione.

Il settore master della split view governa il funzionamento dell'applicazione, è la base che coordina l'avvicendamento tra le finestre: il nome delle funzionalità principali è inserito in un array e, tramite un costrutto switch basato sul loro indice, ad ognuna viene associata una diversa vista con le informazioni specifiche destinata ad apparire nell'area principale dello schermo.

La prima finestra di dettaglio creata per l'applicativo è stata quella riguardante l'anagrafica dei clienti aziendali.

La vista è costituita da una tabella nella quale ogni riga contiene uno dei clienti presenti nel database; è importante notare come la voce "Clienti" alla sinistra dello schermo resti evidenziata per localizzare rapidamente l'area d'interesse.

Nel codice di esempio si evidenzia come il metodo destinato a gestire la voce selezionata nella sezione master agisce per far comparire la vista desiderata, impostando l'opportuno controller:

```
- (void)tableView:(UITableView *)aTableView didSelectRowAtIndexPath:
(NSIndexPath *)indexPath {
    switch (indexPath.row) {
        case 1:{
            self.viewClientiViewController = [[ViewClientiViewController alloc]
initWithNibName:@"ViewClienti" bundle:[NSBundle mainBundle]];
```

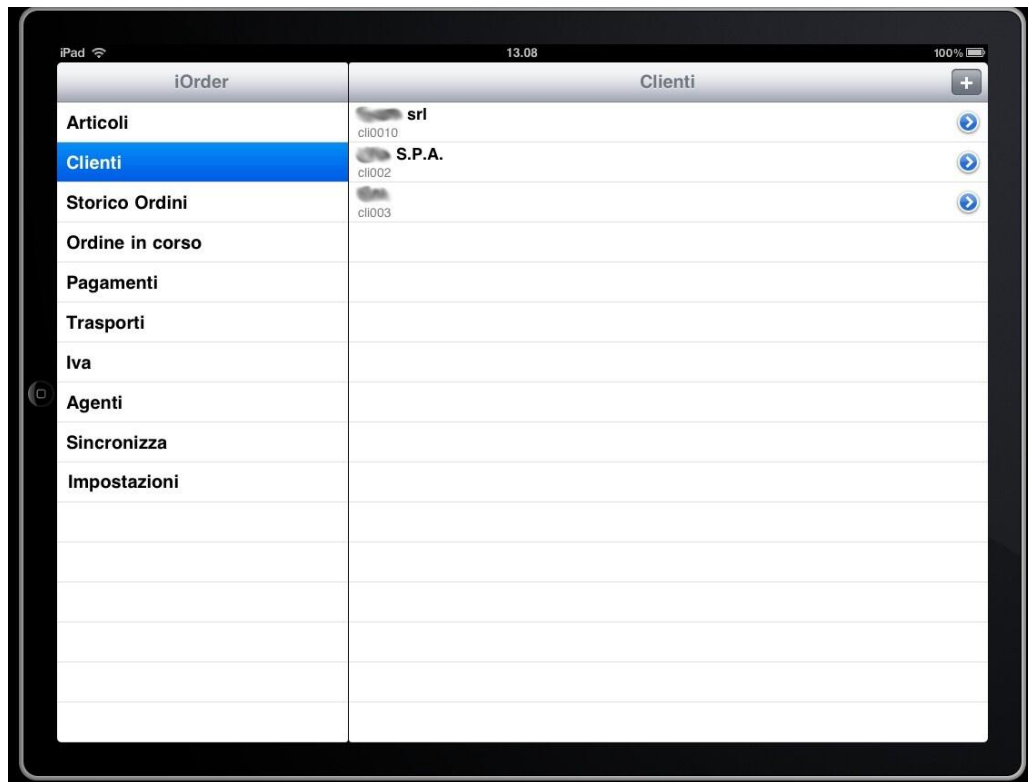


Figura 10.3: Vista clienti

```

self.viewClientiViewController.navigationItem.title =
[menu objectAtIndex:indexPath.row];
    NSArray *viewControllers = [[NSArray alloc] initWithObjects:
mainDelegate.detailViewController, viewClientiViewController, nil]
    [mainDelegate.detailViewController.navigationController setView
Controllers:viewControllers animated:NO];
    [viewControllers release];
}
break;

```

A questo punto l'utilizzatore del software può visualizzare il dettaglio di uno specifico cliente, avviare un nuovo ordine oppure inserire un nuovo acquirente in caso di necessità: queste infatti sono le azioni messe a disposizione dell'utente tramite la gestione di tre eventi differenti.

Per rendere visibile la descrizione di un cliente è necessario premere il bottone blu presente alla destra di ognuno di essi: in questo modo la vista corrente viene sostituita

dal dettaglio che contiene tutte le informazioni desiderate; da notare che nel caso in cui le informazioni non compaiano completamente a video è sufficiente far scorrere con le dita la superficie nelle varie direzioni fino a visualizzare i dati voluti.

Si nota un ulteriore cambiamento riguardante la barra di navigazione in alto sullo schermo la quale è stata modificata per contenere il pulsante di ritorno che permette di risalire nella vasta padre della gerarchia.

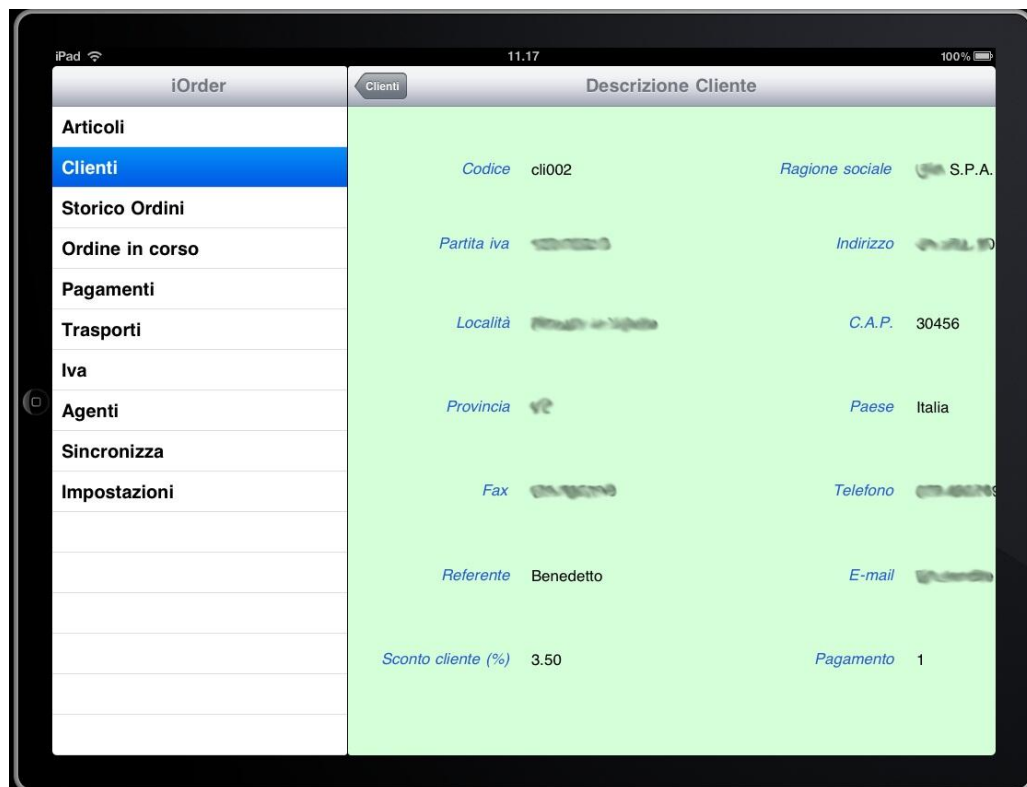


Figura 10.4: Vista dettaglio cliente

Nel caso in cui l'agente desideri aprire un nuovo ordine per un cliente nella lista, questi deve toccare la riga della tabella corrispondente: apparirà a video un avviso, sotto forma di alert view, che lo inviterà a scegliere se confermare la creazione oppure no.

Nell'ipotesi invece che l'agente si trovi presso un nuovo acquirente, e che quindi debba inserirlo e memorizzarlo nel dispositivo, viene richiamata la finestra finalizzata a questo scopo tramite la pressione del pulsante con il simbolo “+” presente nella navigation bar in alto: così facendo ha la possibilità di inserire tutti gli attributi che lo caratterizzano, completando i campi di testo predisposti con la tastiera che compare a video.

Nel momento in cui questa appare in primo piano però occupa all'incirca metà schermo: ecco perché è stato predisposto lo spostamento automatico della casella di testo se-



Figura 10.5: Alert view nuovo ordine

lezionata nel caso in cui si trovi in un'area non visibile durante la digitazione; al termine dell'operazione viene riportata nella posizione originaria.

La selezione del tipo di pagamento utilizzato dal cliente viene effettuata usando il componente grafico denominato picker view, che permette la scelta tra le varie modalità possibili facendo scorrere le opzioni con le dita.

Se alcuni dati inseriti sono errati l'applicativo fa comparire a video un avviso adeguato e vengono colorate di rosso le caselle dove il testo digitato non corrisponde ai requisiti richiesti per facilitarne l'individuazione.

Quando tutti i dati sono stati immessi correttamente è possibile completare l'inserimento sfiorando il tasto "Fine" evidenziato a destra nella barra di navigazione e tornare all'anagrafica dei clienti.

Dall'immagine si rileva anche come, alla comparsa di un alert view, questo venga messo in evidenza, oscurando in modo sfumato il resto dello schermo.

Terminata la gestione delle viste riguardanti i clienti dell'azienda, il lavoro è proseguito allestendo alcune finestre dalla struttura simile tra loro, con informazioni sui tipi di trasposto e pagamento e dei valori di iva.

Queste schermate hanno il compito di esporre le anagrafiche delle varie entità, rappresentando i dati in una tabella, un'occorrenza per riga, all'interno della quale sono presenti tutti gli attributi dell'oggetto con in evidenza la descrizione delle modalità di trasporto e di pagamento ed il valore percentuale dell'iva.

Le tabelle in questione hanno puramente uno scopo informativo, possono essere scorse nel caso in cui sia presente un numero elevato di valori, ma non permettono alcuna azione all'agente che utilizza l'applicazione.

Una vista simile a quelle appena descritte è stata progettata per visualizzare gli agenti di vendita dell'azienda: siccome gli attributi in questo caso sono molto numerosi è risultato più chiaro implementare un comportamento analogo a quello creato nella finestra dei clienti. Infatti la tabella contiene solo il codice ed il nominativo dei dipendenti e, per visualizzare il dettaglio completo delle caratteristiche, è richiesta la selezione della riga desiderata in un punto qualsiasi della sua area, non utilizzando un bottone come nell'altro caso.

Agendo in questo modo compare una finestra del tutto somigliante a quella vista precedentemente per rappresentare il dettaglio di un cliente.

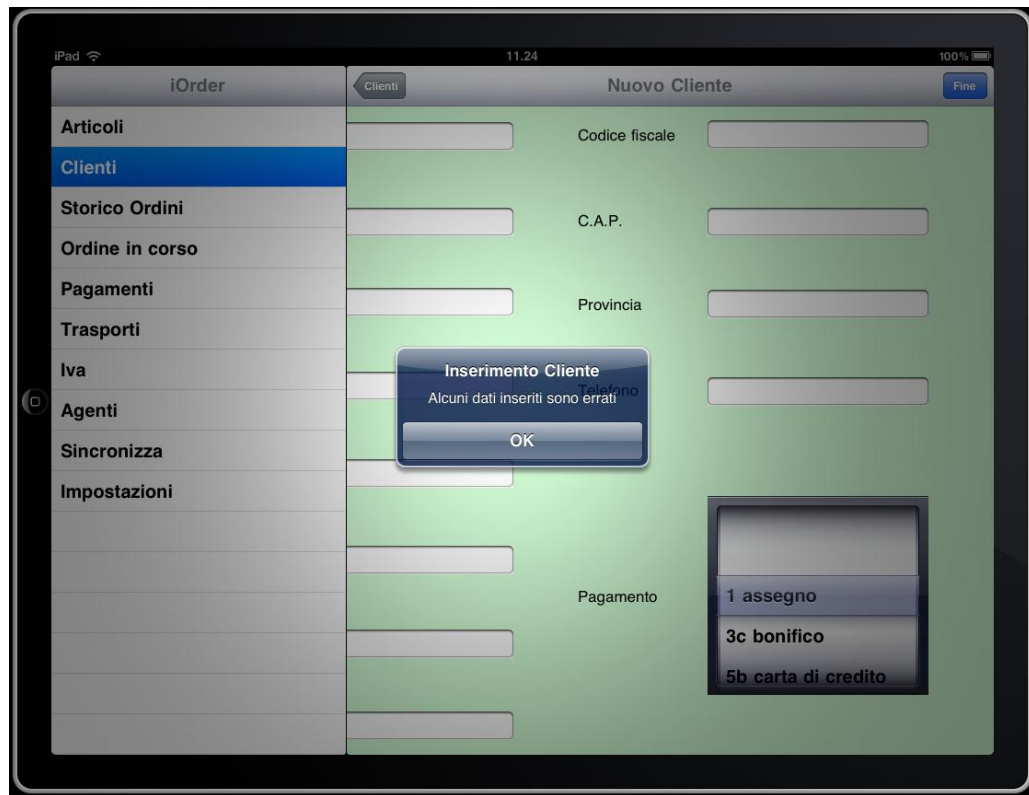


Figura 10.6: Vista inserimento nuovo cliente

Più complessa si è rivelata la gestione delle funzionalità del software iOrder riguardanti gli articoli disponibili alla vendita. La presentazione del catalogo è costituita, come negli altri casi, da una tabella, ma in questo caso lo spazio standard disponibile riga per riga non è risultato sufficiente a contenere anche le sole informazioni fondamentali per ogni singolo oggetto, come la foto, la descrizione, il codice ed il prezzo: è stato necessario quindi progettare una nuova struttura che contenga tutti i dati necessari ad una sintetica esposizione dell'oggetto.

Nel momento in cui l'agente deve trovare un articolo nel listino, può farlo scorrendo la tabella contenente l'intero catalogo, oppure utilizzando la funzione di ricerca appositamente studiata, utile per risparmiare tempo quando sono già note alcune caratteristiche dell'oggetto voluto.

Selezionando il bottone con la lente di ingrandimento nella barra di navigazione, appaiono sullo schermo la tastiera virtuale di iPad ed una piccola finestra, denominata popover view, che contiene la barra di ricerca con scope bar; questi due componenti scompaiono non appena si tocca un'area dello schermo esterna alla loro superficie.

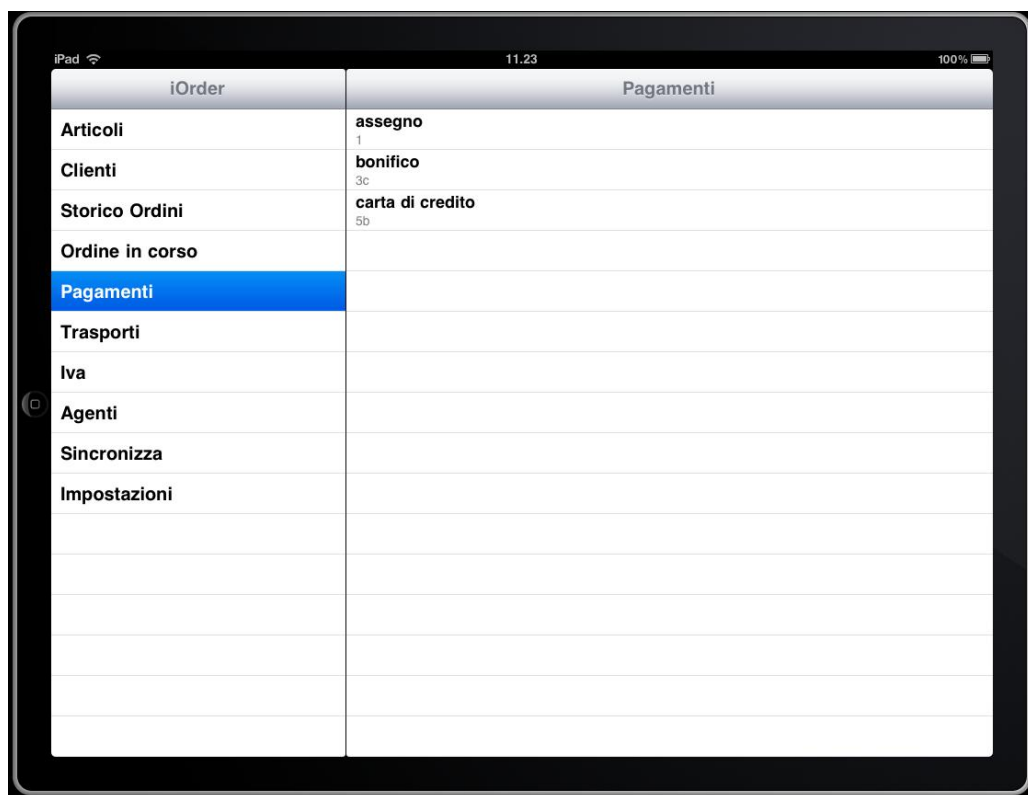


Figura 10.7: Vista pagamenti

	Codice	art002	Quantità	23 pz.	
	Marca	Apple	Prezzo netto	34.89 Euro	
	Descrizione			mouse ottico	

Figura 10.8: Struttura riga nella tabella degli articoli

Ogni lettera digitata o la selezione di un pulsante provocano una variazione negli articoli presenti nella tabella: in un array ausiliario vengono inseriti gli oggetti che mano a mano soddisfano i requisiti ricercati scremando il catalogo intero dei prodotti, in modo che l'utente possa vedere immediatamente i risultati della ricerca durante il progressivo affinamento della stessa.

Come accade nell'anagrafica dei clienti, premendo il tasto blu a destra di ogni riga è possibile richiamare la finestra che permette di visionare il dettaglio del particolare prodotto con tutti i dati che lo caratterizzano. La funzionalità principale di questa sezione è però



Figura 10.9: Ricerca degli articoli

la possibilità di selezionare un articolo dalla lista per aggiungerlo all'ordine d'acquisto. Selezionando uno specifico oggetto, nel caso in cui sia stato attivato un nuovo ordine per l'attuale cliente, appare a video la finestra che consente all'agente di compilare la riga d'ordine per la vendita; in caso contrario un avviso notificherà il mancato completamento dell'operazione.

Questa vista consente di inserire i dati necessari alla raccolta della riga d'ordine, con la quantità desiderata e lo sconto praticato, insieme alla selezione della data prevista di consegna ed al valore iva imposto con l'ausilio di due componenti picker e date picker; quando le informazioni immesse sono complete è possibile aggiungere definitivamente l'articolo all'ordine tramite l'apposito pulsante "Seleziona", e quindi ritornare al catalogo.

La gestione dei dati relativi ad un nuovo ordine di acquisto rappresenta il nocciolo dell'esecuzione del programma ed interessa tutti gli elementi descritti nelle viste sviluppate fino a questo momento; questa funzionalità contiene indicazioni sul cliente acquirente e sull'agente, sugli articoli scelti, con le caratteristiche di pagamento e trasporto preferite. La vista che contiene le informazioni dell'ordine attivo è progettata per contenere i suoi numerosi attributi, oltre a fornire all'utente le scelte necessarie e la possibilità di scendere nella ger-

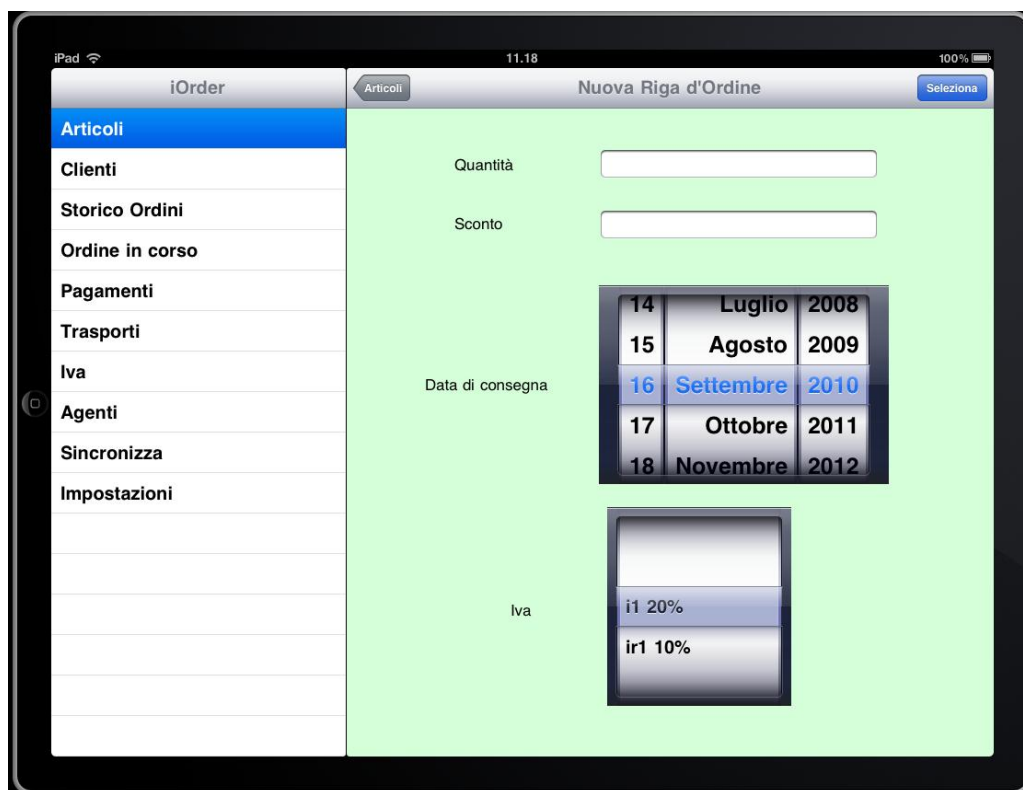


Figura 10.10: Vista nuova riga d'ordine

archia per vedere quali articoli lo compongono. Innanzitutto sono rappresentati i dettagli identificativi dell'ordine, composti dal suo tipo, anno e numero, seguiti dalle caratteristiche principali dell'acquirente e da un componente grafico per scegliere la data di sottoscrizione dell'operazione. In fondo alla vista è presente un picker view multiplo che permette, prima della chiusura del contratto di vendita, di scegliere le opzioni riguardanti le modalità di trasporto e pagamento ed il dipendente dell'azienda che raccoglie l'ordine, in questo caso se stesso: nel caso non vengano modificate queste opzioni si ritengono selezionate quelle evidenziate di default dal componente in questione. Oltre alle informazioni che riguardano le generalità dell'acquisto, il programma è stato progettato per fornire la possibilità di visualizzare tutti gli articoli che lo compongono ed infatti il bottone “dettaglio” è destinato a questo scopo. La sua pressione fa comparire una nuova vista con una tabella che contiene tutte le righe d'ordine inserite fino a quel momento con la descrizione dei dati più significativi: la pressione di una di queste, nello stesso modo visto in precedenza, pone in primo piano una finestra con il dettaglio e tutte le sue caratteristiche.

L'applicazione consente inoltre di eliminare le eventuali righe per le quali il cliente abbia avuto dei ripensamenti, utilizzando il pulsante “Edit” in alto sullo schermo. In questo modo,

Agente	Pagamento	Trasporto
	1 assegno	
	3c bonifico	
ag01 Rossi	5b carta di credito	tra01 corriere
ag05 Verdi		tra05 posta
ag07 Bond		

Figura 10.11: Vista ordine in corso

Numero riga	2	Data consegna	16-10-2010
Codice articolo	art002	Quantità	10
Descrizione	mouse ottico		

Figura 10.12: Struttura riga d'ordine

alla sinistra di ogni riga della tabella, appare un cerchietto rosso comprendente il simbolo “-“: nel momento in cui l’utente preme questo tasto esso ruota di 90 gradi e contemporaneamente compare alla destra delle informazioni sull’articolo ordinato il bottone rosso “Delete“. Nel caso in cui questo venga premuto la riga in questione viene cancellata dall’ordine e tutte quelle che la seguono scalano di una posizione verso l’alto. La procedura appena descritta può essere effettuata per tutte le righe della tabella e termina nel momento in cui viene premuto il tasto “Done“ nella barra di navigazione; a questo punto i bottoni rossi a sinistra degli articoli scompaiono e la finestra torna ad assumere le funzionalità originarie. Quando l’acquirente ha scelto gli articoli di cui aveva bisogno e le eventuali modifiche alle righe sono state apportate, l’agente può chiudere l’ordine ed inviarne una copia ciascuno al cliente ed

all'azienda con il tasto "Fine" appositamente creato. È importante notare che i prezzi totali contenuti nella vista principale di questa sezione sono aggiornati ogni qualvolta vi siano delle modifiche ai prodotti presenti, sia che si tratti di aggiunte oppure di cancellazioni.

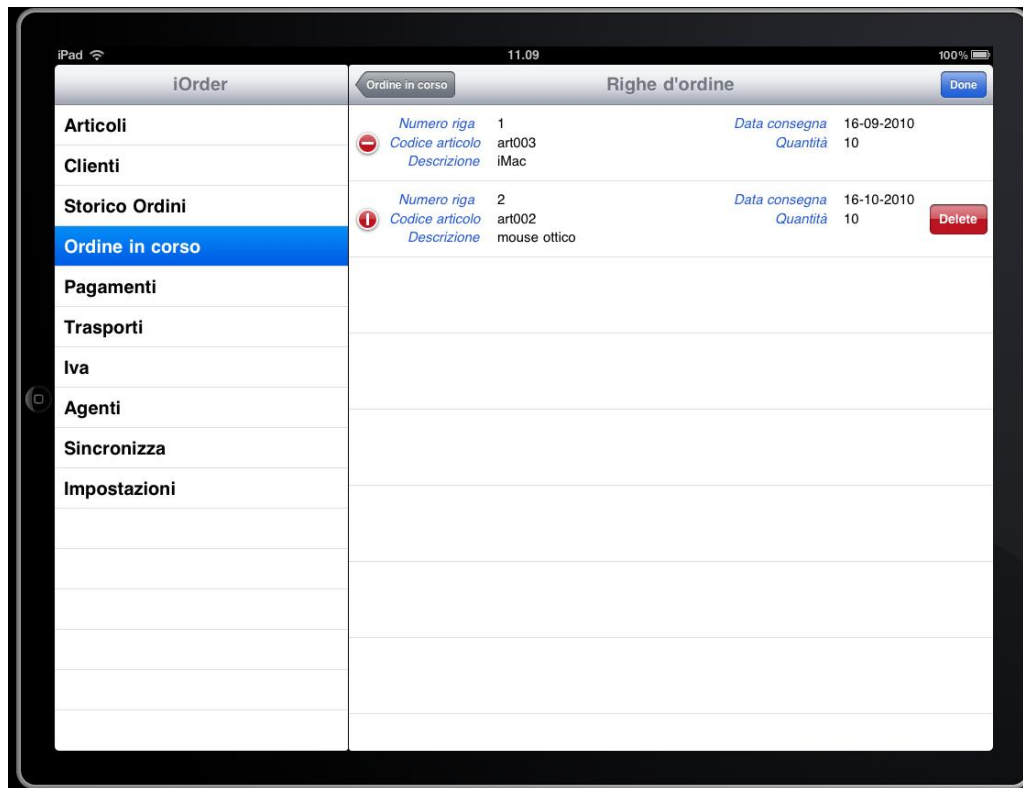


Figura 10.13: Cancellazione righe d'ordine

Proprio a questo punto è emersa la necessità di rinunciare alla stampa di copie cartacee dell'avvenuta operazione come richiesto nelle specifiche dell'applicazione, dato che iPad non supportava questa funzionalità al momento dello sviluppo del software e sarà integrata nel sistema operativo solo con il lancio della nuova versione prevista per novembre.

Al termine dell'ordinazione dunque appare a video una schermata per la composizione di una e-mail, con i campi già impostati ma modificabili, ed in allegato un file PDF contenente il fincato dell'ordine appena memorizzato.

Il codice sorgente che segue implementa l'operazione appena descritta:

```
MFMailComposeViewController *mail=[[MFMailComposeViewController alloc] init];
mail.mailComposeDelegate = self;
if([MFMailComposeViewController canSendMail]){
```

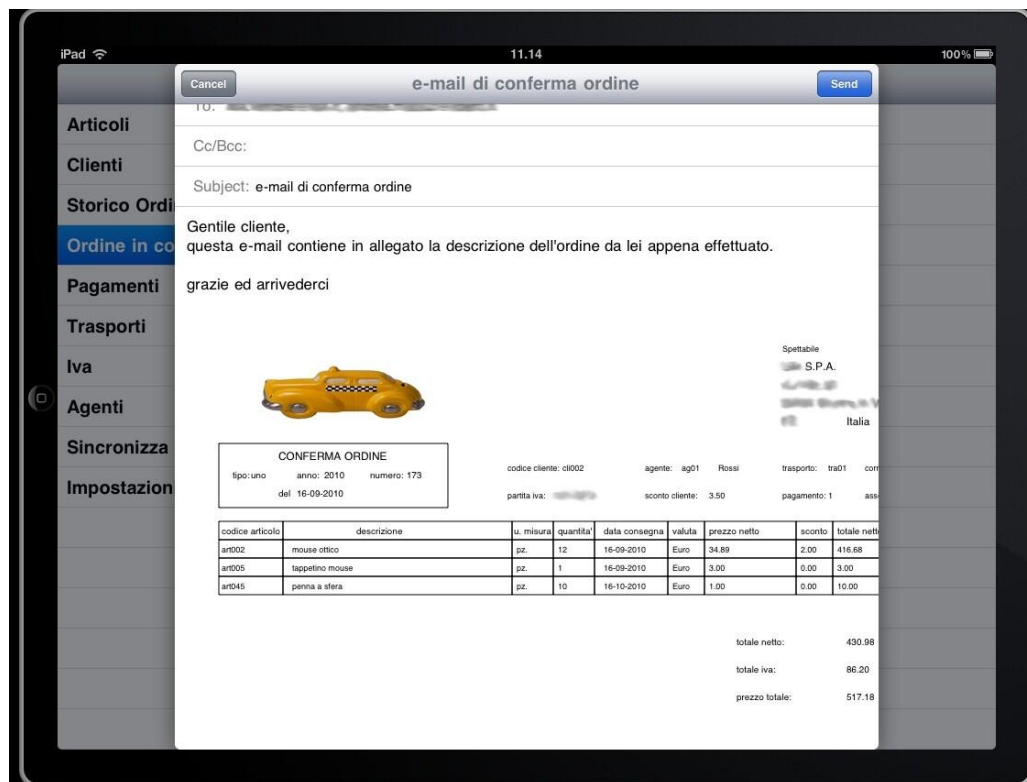


Figura 10.14: Invio e-mail di conferma

```
[mail setToRecipients:[NSArray arrayWithObjects: mainDelegate.cliente
Selezionato.email, nil]];

mail setSubject:@"e-mail di conferma ordine"];

mail setMessageBody:@"Gentile cliente, \n questa e-mail contiene in allegato
la descrizione dell'ordine da lei appena effettuato. \n\n grazie ed arrivederci"
isHTML:NO];

NSData *myData = [NSData dataWithContentsOfFile:newFilePath];

[mail addAttachmentData:myData mimeType:@"application/pdf" fileName
:pdfName];

[self presentViewController:mail animated:YES];
}
```

Il file sopra citato è creato utilizzando una libreria apposita (MessageUI) aggiungendo una ad una tutte le informazioni necessarie con la precisa indicazione della posizione occupata, e nominandolo con la sequenza di tipo, anno e numero dell'ordine in questione scritti senza spazi. Oltre ad essere inviato per posta elettronica, il documento viene anche salvato nella stessa cartella del dispositivo in cui si trova il file dalla base di dati, nel caso fosse necessario in un secondo momento. La numerazione dell'ordine avviene in maniera automatica, utilizzando la possibilità di salvare dei dati durante l'uso dell'applicazione per ritrovarli ai successivi avvii del programma, con la libreria NSUserDefaults:

```
//caricamento numero ordine

NSString *numeroSalvato = [[NSUserDefaults standardUserDefaults] objectForKey:@"numero"];

if (numeroSalvato == nil) {
    ordine.numero = 1;
}
else {
    ordine.numero = [[[NSString alloc] initWithFormat:@"%d", numeroSalvato]
    intValue] + 1;
}

//salvataggio numero ordine
NSUserDefaults *numSalvato = [NSUserDefaults standardUserDefaults];

[numSalvato setObject:[NSString stringWithFormat:@"%d", ordine.numero]
forKey:@"numero"];

[numSalvato synchronize];
```

In questo modo il numero viene incrementato di uno anche se il programma viene riavviato, per mantenere l'univocità del valore: nel momento in cui fosse necessario azzerare questa cifra, se ad esempio si desidera che all'inizio di un nuovo anno la numerazione riparta da uno, è sufficiente cancellare il file di impostazioni apposito contenuto nella cartella di iPad riservata al software.

Come specificato nei requisiti, lo scopo dell'applicativo non è solamente comporre degli ordini ed inviarli ai soggetti interessati, ma è richiesto anche di salvare quelli mano a mano raccolti in uno storico, da sincronizzare poi con il database dell'azienda. Ecco perché, nel momento in cui l'ordine in corso viene chiuso, questo è memorizzato e consultabile tramite la voce "Storico Ordini" presente nel menù master di iOrder per mezzo di una tabella che ricalca il design di quelle viste fino ad ora.

iOrder		Storico Ordini			
Articoli	Tipo	uno	Data	15-09-2010	>
Clienti	Anno	2010	Cliente	cli018	
	Numero	171	Ragione sociale		
Storico Ordini	Tipo	uno	Data	16-09-2010	>
Ordine in corso	Anno	2010	Cliente	cli002	
Pagamenti	Numero	172	Ragione sociale	S.P.A.	
Trasporti	Tipo	uno	Data	16-09-2010	>
Iva	Anno	2010	Cliente	cli002	
Agenti	Numero	173	Ragione sociale	S.P.A.	
Sincronizza					
Impostazioni					

Figura 10.15: Vista storico ordini

Come in quella contenente il catalogo degli articoli disponibili, anche in questo caso le righe in essa rappresentate sono diverse dal layout standard, in modo da rappresentare le informazioni principali senza dover visionare il dettaglio, e quindi gli identificativi dell'ordine stesso e del cliente che lo ha effettuato, insieme alla data di sottoscrizione. Allo stesso modo in cui sono gestite le altre viste simili, quando si seleziona una riga tra quelle presenti appare a video una finestra con il dettaglio dell'ordine desiderato; qui, a differenza di quando questo è ancora modificabile, sono presenti solo un insieme di label che espongono tutte le caratteristiche, anche dove in precedenza era possibile per l'utente effettuare delle scelte.

Le altre funzionalità accessibili da questa vista sono del tutto simili a quelle già viste per l'ordine in corso: con la pressione del pulsante “Dettaglio” è possibile accedere ad un elenco di tutti gli articoli acquistati, e la selezione di uno tra quelli disponibili fa scendere ancora nella gerarchia per visionare il dettaglio dello stesso. L'unica differenza è la mancanza del tasto “Edit” nella barra di navigazione in alto sullo schermo, data l'impossibilità a questo punto di modificare le righe d'ordine presenti.

Un'ultima necessità per il funzionamento completo dell'applicazione in locale non era specificata nei requisiti ma si è manifestata in corso d'opera, ed ha a che fare con l'occu-

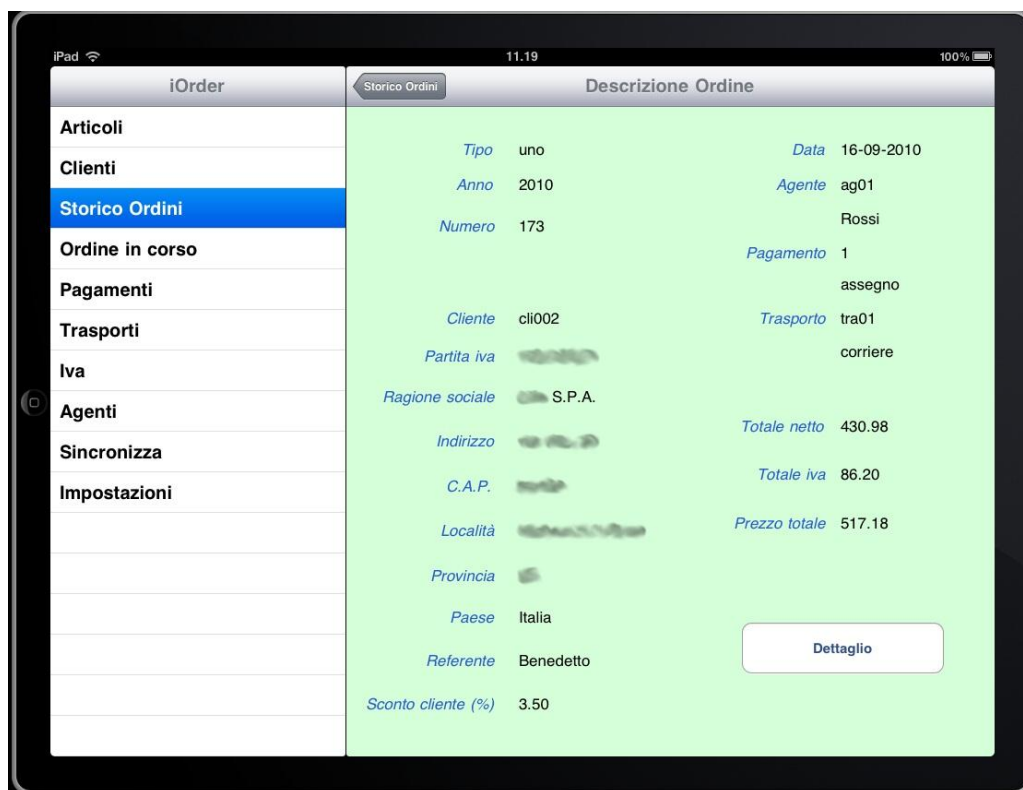


Figura 10.16: Vista ordine dello storico

pazione di memoria del programma iOrder. Infatti con il susseguirsi della raccolta degli ordini, e perciò del salvataggio dei loro file PDF nella cartella appositamente creata in iPad, questi con il passare del tempo vanno ad occupare una considerevole quantità di spazio nella memoria del dispositivo; è sorta quindi la necessità di fornire all'utente la possibilità di svuotare lo spazio interessato, nel momento in cui i file non sono più necessari. La vista che appare in primo piano premendo l'opzione "Impostazioni" è stata pensata proprio a questo scopo e si presenta in una forma particolare rispetto alle finestre analizzate fino a questo momento: anche se il layout ricorda più quello di un bottone da premere per ottenere l'operazione voluta, in realtà si tratta di una forma particolare di tabella, con delle caratteristiche diverse da quelle utilizzate fin qui.

Questo particolare effetto si ottiene utilizzando una "Grouped Table" che in realtà è una tabella nella tabella: ogni sotto-tabella è preceduta da un titolo e seguita da una breve descrizione, ma ogni riga che la compone può avere lo stesso comportamento visto finora. In questo caso selezionando l'unica riga disponibile appare a video un avviso che informa l'utente della cancellazione dei file PDF. La finestra in questione è stata pensata, oltre per il comportamento descritto, anche per un'evoluzione futura del software, rendendola più

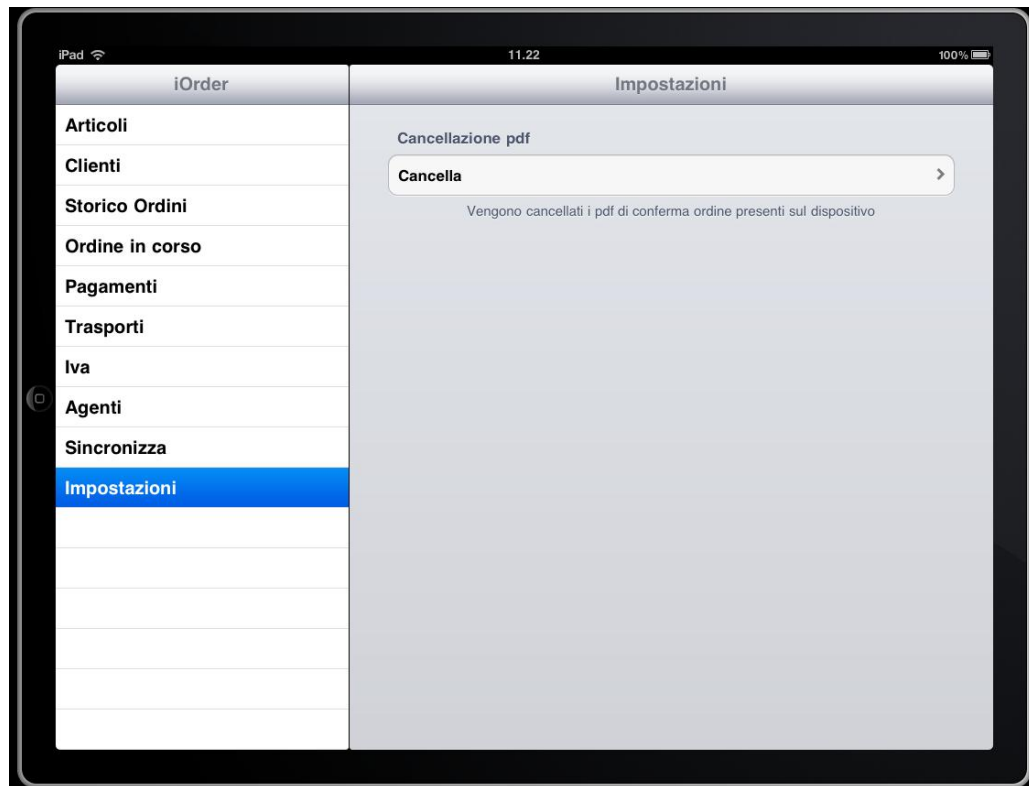


Figura 10.17: Vista impostazioni

completa con altre funzionalità e regolazioni da offrire all'utente.

A questo punto è stato descritto completamente il funzionamento dell'applicativo in locale, ma una delle caratteristiche principali richiesta nei requisiti riguarda la possibilità di sincronizzare i dati memorizzati nel dispositivo con il database del gestionale aziendale e viceversa, a seconda delle tabelle presenti.

La vista "Sincronizza" serve a questo fine ed è implementata usando lo stesso criterio appena visto per la finestra che gestisce le impostazioni: usando una tabella raggruppata vengono mostrate all'utente le varie possibilità di aggiornamento delle basi di dati, e questi può scegliere singolarmente quella che desidera adeguare da o verso remoto, oppure ha la possibilità di sincronizzare contemporaneamente tutte le tabelle dal gestionale dell'azienda. La pressione delle varie opzioni avvia il trasferimento dei dati richiesti, che termina con un diverso avviso all'utilizzatore a seconda se l'operazione ha avuto un esito positivo oppure no. Come si vede dall'immagine è possibile lo spostamento da remoto dei dati che riguardano gli articoli, i pagamenti, i trasporti, i valori di iva e gli agenti che lavorano per l'azienda, mentre il percorso inverso può essere compiuto dalle informazioni sugli ordini raccolti; i clienti invece sono aggiornabili in entrambe le direzioni dato che nuovi acquirenti possono

essere inseriti in tutte e due le basi di dati.

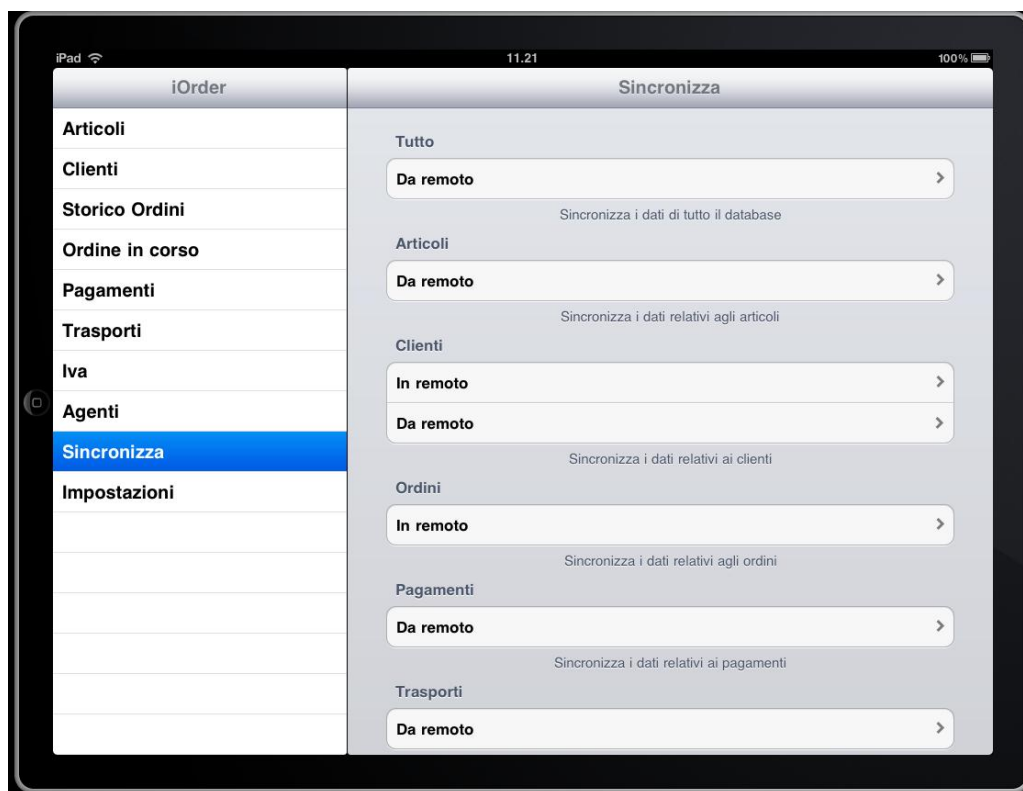


Figura 10.18: Vista sincronizza

10.3 Gestione dei trasferimenti

Per effettuare il passaggio delle tabelle tra database locale e remoto, si è scelto di utilizzare dei documenti XML da trasferire mediante il protocollo di trasferimento file standard: sono state caricate nel server FTP delle tabelle fittizie contenenti i dati da utilizzare come prova del corretto funzionamento del programma. Si intende infatti che al momento della sincronizzazione, siano presenti nel server FTP dei file XML ottenuti dal gestionale dell'azienda con delle funzioni di export delle tabelle richieste.

La creazione dei file XML all'interno dell'applicativo è stata implementata con un'apposita classe denominata XMLCreator, i cui metodi sono stati richiamati per inserire i tag necessari e tutti gli attributi delle varie entità.

Si passano come parametri ai metodi sotto descritti tutti gli attributi da inserire nel file per formare la struttura XML:

```
-(id)init{
    [super init];

    testoXml = [[NSMutableString alloc] initWithFormat:@"%<?xml version=
    \"1.0\" encoding=\"UTF-8\"?>"];

    return self;
}

-(void)aggiungiElementoXml:(NSString *)tag elemento:(NSString *)elemento{
    [testoXml appendFormat:@"%n<%>%</%>", tag, elemento, tag];
}

-(void)aggiungiTagAperturaXml:(NSString *)tag{
    [testoXml appendFormat:@"%n<%>", tag];
}

-(void)aggiungiTagChiusuraXml:(NSString *)tag{
    [testoXml appendFormat:@"%n</%>", tag];
}
```

Il documento creato viene salvato nella cartella dedicata sul dispositivo, come gli altri file gestiti dall'applicazione, dopo che i dati sono stati inseriti per mezzo di un ciclo for che itera tutti gli elementi presenti nella tabella richiesta.

Quando giunge il momento di trasferire il file dalla periferica al server remoto, si procede effettuando un collegamento FTP ed inviando tutti i dati necessari attraverso l'accesso ad un'apposita cartella sul server, con la condizione di avere a disposizione i permessi richiesti e l'indirizzo a cui il documento deve essere spedito. Dopo aver creato uno stream per inviare il file desiderato, questo, per mezzo di una classe opportuna, gestisce gli eventi che lo riguardano, soprattutto il caso in cui il flusso abbia dello spazio a disposizione per trasmettere dei dati che quindi vengono mano a mano spediti al documento nel server FTP.

Nel gestire il funzionamento inverso, ovvero il recupero delle tabelle dalla base di dati in remoto, si utilizza una classe simile alla precedente, seppure con le opportune modifiche. Questa operazione però presenta un nuovo problema, infatti è necessario recuperare gli attributi delle entità a partire dalla struttura del file XML.

Un oggetto che si occupa di recuperare i dati da un documento di questo tipo viene detto parser: quello utilizzato è un'istanza della classe NSXMLParser e consente di analizzare i file riga per riga. Invocando un apposito metodo è possibile recuperare uno ad uno i

componenti dalla struttura XML e per mezzo di un costrutto if con varie opportunità memorizzare ognuno di questi in apposite variabili, usate al momento della lettura del tag di fine elemento per creare un nuovo oggetto da salvare nello specifico array.

Conclusioni e sviluppi futuri

GIUNTI AL TERMINE DELLA realizzazione dell'applicativo iOrder si ripercorre il percorso compiuto raccogliendo sinteticamente le fasi di sviluppo ed i risultati raggiunti.

La scelta di rendere più completa la preparazione universitaria con un'esperienza nel mondo del lavoro assieme all'interesse suscitato dalla presentazione del nuovo iPad, ha permesso, grazie all'opportunità concessa dall'azienda ADL Ingegneria Informatica, di effettuare un'attività molto interessante e costruttiva.

La prima fase dell'esperienza di tirocinio è stata interamente dedicata allo studio approfondito della periferica Apple, dai suoi ambiti di utilizzo al linguaggio di programmazione richiesto per lo sviluppo, dalla ricerca ed analisi dei principali concorrenti sul mercato fino a giungere ad un percorso dettagliato per l'implementazione di applicazioni destinate a questo dispositivo.

Le conoscenze acquisite e l'apprendimento di un metodo di lavoro adeguato si sono dimostrate essenziali per una progettazione ben strutturata dell'applicativo iOrder, con i requisiti richiesti e l'interfaccia utente dedicata, insieme alla stesura del codice sorgente svolte nella seconda metà del periodo all'interno dell'azienda.

Si sono rivelati utili molti concetti appresi durante il corso di studi frequentato come la metodologia di progettazione di una base di dati e la modellazione UML, insieme a diverse nozioni trattate nei vari insegnamenti che hanno permesso di far fronte alle difficoltà emerse dall'incontro con un linguaggio di programmazione ed un ambiente di sviluppo completamente sconosciuti.

Il software è stato creato con l'ausilio del simulatore fornito con il kit di sviluppo Apple, ed è stato portato a termine nei tempi previsti, con i requisiti richiesti e migliorato con altre funzioni che hanno contribuito ad un risultato più completo; l'unico rammarico è di non aver potuto testare l'esecuzione su un dispositivo iPad reale e di provare in maniera concreta la resa finale.

Il progetto realizzato per l'azienda costituisce la prima versione di un software che sarà soggetto ad adeguamenti prima del rilascio, in base alle esigenze del mercato in cui verrà introdotto e a personalizzazioni su richiesta dei clienti a cui verrà offerto.

Per quanto concerne il futuro dell'applicazione iOrder potrà essere migliorata attraverso l'integrazione della funzionalità di stampa e dell'utilità di ricerca per gli ordini inseriti

nello storico, che potrebbe facilitare l'individuazione degli stessi all'interno di una tabella contenente una discreta quantità di righe.

Bibliografia

- R. A. Elmasri, S. B. Navathe, *Sistemi di basi di dati - Fondamenti*, Paravia Bruno Mondadori Editori, Milano, quinta edizione, 2007
- P. Atzeni, S. Ceri, S. Paraboschi e R. Torlone, *Basi di dati- Modelli e linguaggi di interrogazione*, McGraw-Hill, Milano, seconda edizione, 2006
- R.S. Pressman, *Principi di ingegneria del software*, McGraw-Hill, Milano, quinta edizione, 2007
- Martin Fowler, *UML Distilled*, Pearson, Milano, terza edizione, 2004
- <http://www.apple.com/it>
- <http://developer.apple.com>
- <http://www.devapp.it>

Ringraziamenti

Voglio innanzitutto ringraziare il professor Rumor, mio tutor universitario, sempre molto disponibile e cordiale sia a lezione che durante il periodo di stesura di questa tesi.

Ringrazio il mio tutor aziendale, l'ingegner Della Libera, per avermi dato l'opportunità di svolgere questa attività di tirocinio e per l'appoggio nel corso delle settimane.

Ringrazio inoltre tutti i membri di ADL Ingegneria Informatica per il periodo di collaborazione e soprattutto Luca, Daniele e Simone, compagni di ufficio simpatici e sempre pronti a dare una mano in caso di bisogno.

Ringrazio i miei genitori che mi hanno sempre incoraggiato durante gli studi anche nei momenti in cui dopo tanti sacrifici i risultati raggiunti non erano quelli sperati.

Ringrazio Michela, che è presente in ogni momento e cerca sempre di tirarmi su il morale e di farmi prendere il lato positivo delle cose.

Infine voglio ringraziare tutti i miei parenti, soprattutto quelli che da un po' di mesi non ci sono più e che purtroppo non possono assistere alla conclusione di questo percorso, ma che fino a quando hanno potuto mi hanno sempre voluto bene.

Andrea