



UNIVERSITÀ DEGLI STUDI DI PADOVA



FACOLTÀ DI INGEGNERIA
CORSO DI LAUREA IN INGEGNERIA DELL'INFORMAZIONE

Tesi di Laurea Triennale

Un'App in 48 ore

Developing an App in 48 Hours

Studente:
Mattia Scagno

Relatore:
Chia.mo Prof. Carlo Fantozzi

Anno accademico: 2012/2013 - Padova, 26 Settembre 2013

*Ai miei genitori,
che, sostenendomi in ogni situazione, mi hanno accompagnato fino ad ora.
Che possano essere sempre orgogliosi di me.*

Indice

1. Introduzione.	7
1.1 Introduzione del lavoro di tesi.	7
1.2 Cos'è un Hackathon.	8
1.3 Cosa sono le Universiadi.	8
1.4 Struttura della tesi.	9
2. Sviluppo del codice in un Hackathon.	9
2.1 Osservazioni tecniche sullo sviluppo del codice in generale.	9
2.2 Osservazioni tecniche sullo sviluppo del codice in un Hackathon e confronto con lo sviluppo "classico".	12
3. Presentazione dell'Hackathon "ICT Days 2013".	14
3.1 Piattaforme e servizi per l'Hackathon "ICT Days 2013".	14
3.2 Requisiti richiesti per applicazione sviluppata durante Hackathon "ICT Days 2013".	16
4. Interpretazione e implementazione dell'applicazione "Universiadi 2013".	18
4.1 Interpretazione del progetto "Universiadi 2013".	18
4.2 Implementazione del progetto "Universiadi 2013".	19
4.2.1 Implementazione del progetto "Universiadi 2013" durante l'Hackathon.	19
4.2.2 Implementazione del progetto "Universiadi 2013" durante stage in SmartCampus.	24
5. Conclusioni.	31
5.1 La fine dell'Hackathon: presentazione e sviluppi successivi.	31
5.2 Conclusione del lavoro di tesi.	32
6. Bibliografia.	33

1. Introduzione

1.1 Introduzione del lavoro di tesi

Le app (abbreviazione di “applicazione”) sono moduli software che vengono installati nei moderni dispositivi mobili, i così detti “smartphone”: essi sono dispositivi portatili con capacità di calcolo e di connettività molto superiori ai telefoni GSM. Possiamo tracciare una linea di confine tra telefoni cellulari e smartphone da quando i dispositivi mobili montano un SO (Sistema Operativo) e possiedono connettività Wi-Fi, 2G e 3G.

Le applicazioni sono a tutti gli effetti paragonabili al software che troviamo in un qualunque PC ma, a causa della ridotta potenza di calcolo, dello schermo più piccolo e di altre limitazioni dei device mobili, le app si poggiano sul dogma della semplicità e dell’eliminazione del superfluo, per rendere l’esperienza dell’utente intuitiva e veloce.

Le app si dividono in “native” e “cross-platform”: le prime vengono progettate appositamente per un sistema operativo (sia esso iOS, Android o Windows Phone), le seconde sono molto simili a siti Web e, per questo, possono essere visualizzate in qualunque dispositivo, indipendentemente dal SO montato, senza un’ottimizzazione specifica.

La “creazione” di un’applicazione per dispositivi mobili avviene come un qualunque applicativo software: è necessario un linguaggio di programmazione e un ambiente di sviluppo nel quale il linguaggio venga compreso ed interpretato.

Nello specifico, le applicazioni iOS (montate su dispositivi Apple quali iPhone, iPad e iPod) sono sviluppate in Objective-C all’interno dell’IDE¹ proprietario Apple denominato xCode.

Le applicazioni Android (montate su dispositivi Google ma non solo) sono sviluppate in Java² all’interno di un qualunque IDE che supporti il linguaggio stesso (Eclipse oppure NETBeans oppure Andorid Studio, appena lanciato proprio da Google).

Il seguente lavoro di tesi si prefigge di spiegare come sia possibile sviluppare un’applicazione per dispositivi smartphone all’interno di una maratona di programmazione (Hackathon, vedi § 1.2) e di come il progetto vincente di questa competizione venga poi terminato e reso fruibile agli utenti.

La maratona ha avuto come tema le Universiadi 2013 (vedi § 1.3).

¹ Un IDE (integrated design enviroment) è un programma software che, in fase di programmazione, aiuta il programmatore a sviluppare il codice sorgente.

Esso è composto, solitamente, da un editor di codice sorgente, un compilatore e/o un interprete, un debugger.

² Java è un linguaggio di programmazione sviluppato da Sun Microsystems.

Esso, grazie all’inserimento di una macchina virtuale JVM (Java Virtual Machine) si prefigge l’obiettivo di essere un linguaggio di programmazione indipendente dalla piattaforma su cui il codice sorgente viene lanciato.

Caratteristiche peculiari del Java sono l’orientazione agli oggetti e la portabilità (grazie appunto alla JVM)

1.2 Cos'è un Hackathon

Hackathon è un evento al quale partecipano a vario titolo esperti del settore informatico, sviluppatori di software, programmatori e web designer.

Il nome nasce dalla fusione di due termini, “hack”³ e “marathon”: esso, infatti, consiste in una vera e propria maratona, che può durare da 24 ore a una settimana, in cui i partecipanti, divisi in gruppi, risolvono un problema assegnato realizzando software e/o materiale hardware.

Un Hackathon si svolge, normalmente, in 3 fasi:

- Presentazione del “tema”, delle problematiche che gli hackers dovranno risolvere. Spesso queste problematiche sono portate dagli sponsor che organizzano e promuovono la maratona stessa.
- Sviluppo della soluzione attraverso varie piattaforme hardware e/o software, a discrezione dei partecipanti.
- Presentazione dei progetti alla giuria e decretazione dei vincitori.

La manifestazione nasce come momento di crescita e di miglioramento di un qualcosa che già esiste: a tutt'oggi gli Hackathon sono utilizzati dalle aziende non solo per la promozione del loro prodotto ma anche per ricevere feedback dagli hackers stimolando l'upgrade del prodotto stesso.

1.3 Cosa sono le Universiadi

Le Universiadi (o Olimpiadi Universitarie) sono una manifestazione sportivo-culturale che si svolge tutti gli anni dispari, ogni edizione in un luogo diverso.



Non è una manifestazione puramente agonistica ma coniuga spirito sportivo e spirito accademico, come dice il motto stesso della manifestazione: *“Excellence in Mind and Body”* (*“Eccellenza nel corpo e nella mente”*). La particolarità di questa competizione è che gli atleti partecipanti devono essere iscritti ad una università pubblica del paese che rappresentano.

Simbolo della manifestazione è la “U” circondata dalle 5 stelle (i 5 continenti come nello stemma olimpico) e l'inno è il “Gaudeamus igitur”, inno internazionale della goliardia e spesso associato all'Università come nel caso dei Goliardi universitari.

³ Hacker (derivazione del verbo “to hack”, accedere ad un sistema, rompere le barriere di sicurezza del sistema stesso) è una persona, di solito specializzata nel settore informatico o dell'ingegneria elettronica, che si impegna nell'aggirare creativamente le limitazioni che i prodotti che utilizza gli impongono creando upgrade e migliorie del prodotto stesso; possiamo vedere un “hacker” come un inventore che parte già con un prodotto finito e lo migliora, lo adatta alle sue esigenze.

La denominazione stessa del termine è fuorviante in quanto coloro che entrano illegalmente nei sistemi informatici per distruggerli sono definiti, anche dagli hackers stessi in maniera dispregiativa, “crackers”, cioè coloro che craccano, che rompono un sistema.

Le Universiadi si dividono in edizioni Estive (che constano di 10 discipline) e Invernali (che constano di 8 discipline) per un totale di più di 3000 partecipanti divisi in più di 50 nazioni.

Nell'edizione trentina ci saranno, invece, 13 discipline che si svolgeranno in 5 località trentine: Val di Fassa, Val di Fiemme, Pergine, Baselga di Pinè e Trento.

L'altra faccia della medaglia rispetto al risvolto sportivo delle Universiadi è l'organizzazione: è necessario un nutrito numero di persone dello staff organizzatore per coordinare e sostenere la manifestazione prima e durante i giorni delle gare. Sono messi a disposizione, quindi, servizi di hostess/stuart all'interno e all'esterno degli stadi e dei campi gara, parcheggiatori, driver e attaché⁴. All'interno di ogni gruppo, poi, i membri dello staff sono divisi secondo le proprie competenze, abilità e attitudini, in modo da formare un complesso organigramma a piramide in cima alla quale c'è il capo venue⁵.

1.4 Struttura della tesi

La seguente tesi si sviluppa attraverso tre macro argomenti, corrispondenti ad altrettanti capitoli: il capitolo 2 spiega le linee guida per la progettazione di un'applicazione in senso generico confrontando la realizzazione dell'applicativo dentro e fuori il contesto di una competizione come Hackathon.

Il capitolo 3 entra nello specifico considerando lo sviluppo del codice per l'applicazione "Universiadi 2013" all'interno dell'Hackathon ICT Days 2013. Vengono presentati i requisiti posti dagli organizzatori e una visione generale sull'interpretazione del mio team dei requisiti stessi.

Infine il capitolo 4 fornisce i dettagli implementativi dell'applicazione, entrando nello specifico delle singole funzionalità.

2. Sviluppo del codice in un Hackathon

2.1 Osservazioni tecniche sullo sviluppo del codice in generale

Come già anticipato nell'introduzione, lo sviluppo di un'applicazione prevede l'utilizzo di un linguaggio di programmazione e di un IDE. Ma questo è solo l'ultimo passo di una scala di attività che parte da ben prima.

Il problema che in questi ultimi anni è sorto intorno allo sviluppo delle applicazioni è il fatto che questo è un mondo nuovo: le regole, le cose da fare e non fare, gli schemi procedurali da seguire non sono ancora stati codificati e quindi gli sviluppatori "navigano a vista", guidati dall'esperienza e dagli errori passati, loro e dei colleghi.

⁴ Gli attaché sono un particolare gruppo di volontari altamente specializzati nelle lingue: essi vengono affiancati a personalità di spicco o delegazioni straniere durante tutto il corso della manifestazione.

⁵ Una venue è un campo gara o uno stadio ma rappresenta, in senso astratto, un settore dell'albero gerarchico. Ogni venue è organizzata in modo autonomo (a seconda delle peculiarità del campo gara) ed è strutturata gerarchicamente come un altro albero: la radice è il capo venue, sotto i responsabili dei vari settori (ingresso, parcheggio, campo gara ecc..) fino ad arrivare al singolo membro dello staff.

La mia personale esperienza in questo tipo di manifestazioni conta tre partecipazioni ad altrettanti Hackathon, uno di 48 ore (gli ICT Days, la competizione presa in esame in questa tesi) e altri due della durata di 24 ore. Questo bagaglio mi ha permesso di stilare un prontuario di procedure da seguire durante questo tipo di competizioni.

La sequenza di attività che ora esporrò, quindi, non è da prendere come uno schema fisso e studiato ma più come un sunto di ciò che la mia esperienza e quella di altri colleghi sviluppatori ha prodotto.

Il primo step parte dal cliente: se il cliente è un'entità terza la cosa fondamentale è comprendere i requisiti e far comprendere a chi commissiona cosa è possibile e cosa non è possibile fare.

Lavorare per un cliente esterno è sempre più complesso perché ogni successivo step dovrà essere validato e approvato dal committente stesso e non mancano "marce indietro" e rivisitazioni di ciò che è già stato fatto.

Se il cliente e il programmatore coincidono, quindi se si sviluppa un'applicazione da vendere direttamente negli Store⁶, il primo passo fondamentale è trovare una propria nicchia di mercato in cui collocare l'applicazione.

Il mercato dei dispositivi mobili è cresciuto esponenzialmente nel giro di una decina di anni⁷ e ciò ha fatto sì che praticamente chiunque fosse un possibile fruitore di applicazioni. Questo ha portato le software house e i programmatori singoli a sovraccaricare gli Store di applicazioni per qualsiasi utilizzo.

Oggi giorno è difficile ritagliarsi una fetta di mercato in un mondo che sembra già saturo e il consiglio che Chad Mureta⁸ dà a tutti gli sviluppatori è proprio quello di innovare qualcosa che già è stato creato: "Non basta cambiare il nome e il colore" dice Chad ma innovare significa capire perché quella determinata applicazione ha avuto o sta avendo successo e cercare di "essere diversi", mettere nella propria app qualche caratteristica che nessuno ha ancora inserito oppure inserire qualcosa che già c'è ma in una veste nuova, migliore e più intuitiva. In una parola, innovare.

Molto importante in questa fase è, infine, la possibilità di avere dei feedback da un target il più ampio e variegato possibile: il rischio è quello di creare un'applicazione ricalcando le proprie necessità e aspettative; un'applicazione che deve essere venduta, si spera, al maggior numero di utenti possibili deve invece rispondere alle richieste dell'utente finale.

Il fatto che il mercato dei dispositivi mobili e delle app è cresciuto unito alle peculiarità del dispositivo quali ridotta potenza di calcolo, ridotta superficie dello schermo e altro, ha costretto gli sviluppatori a seguire il primo vero "dogma" della programmazione di app: la semplicità. Dunque il secondo gradino della nostra scala è proprio lo studio di come

⁶ Le aziende produttrici di SO per smartphone (Google, iOS e Microsoft) mettono a disposizione degli Store ovvero delle applicazioni che vendono le app prodotte dagli sviluppatori.

Il numero di applicativi presenti nei due maggiori Store, App Store e Google Play, conta, a oggi, circa 700mila app ciascuno con un giro d'affari che toccherà i 25 miliardi di dollari entro fine anno facendo segnare un +62% nei guadagni nel settore della telefonia mobile (stime di Marzo 2013 effettuate da Gartner, azienda leader mondiale nella consulenza strategica, ricerca e analisi nel campo dell'Information Technology).

⁷ Basti pensare che per il 2013 le stime parlano di un numero complessivo di telefoni cellulari venduti nei 12 mesi pari a oltre 1,8 miliardi di pezzi; per il 2017 le previsioni vedono una ulteriore crescita sino a oltre 2,3 miliardi di unità vendute.

Solo due anni fa il mercato degli smartphone aveva totalizzato vendite per mezzo miliardo di pezzi in 12 mesi.

⁸ Chad Mureta è fondatore di Empire Apps e co-fondatore di T3 Apps e Best Apps, tre applicazioni che lui stesso ha creato e venduto, poi, per milioni di dollari.

rendere il nostro software semplice: non semplicemente intuitivo, facile da capire, ma anche semplice nella grafica e pulito nei contenuti e in come essi vengono mostrati. Il metodo migliore per raggiungere lo scopo è nel disegnare (nel vero senso della parola, su carta o con tool per il mockup) il flusso dell'applicazione.

Le cose da non fare durante le operazioni di design dell'app sono sicuramente il creare schermate sovrabbondanti: l'utente non conosce l'app e tutti i suoi flussi quindi inserire troppe cose in una sola schermata confonderebbe l'utente.

Un'altra cosa da evitare sono lunghe o macchinose procedure: l'utente si aspetta risposte immediate ai suoi "stimoli" e non concepisce il dover fare molte azioni per accedere ad una funzionalità. Le risposte devono essere, quindi, immediate, con tempi di attesa brevi e features accessibili con pochi tocchi dello schermo.

Durante un'intervista ad un magazine online Kevin Systrom, CEO di Instagram⁹, disse che quando un utente arriva su una schermata, l'applicazione deve sapere già cosa l'utente vorrà fare. Ovviamente questa è una forzatura, perché le applicazioni sono pur sempre codice, però l'iperbole calza bene quando si parla di reattività e semplicità di utilizzo.

Ma non basta che i dati vengano visualizzati in tempi brevissimi se poi sono mostrati all'utente in modo confuso: Uday Gajendar, Visual Design di Citrix, in un'intervista a Dice disse di non spaventarsi che l'utente non riesca a vedere una mole di dati ampia se il modo in cui essi sono presentati è chiaro, pulito, se il font del testo è leggibile e "leggero", se i dati sono equamente spaziati e distribuiti all'interno dello schermo.

In sostanza la sensazione che l'utente deve avere è che i dati siano accessibili.

In questo si inserisce anche il design dell'app stessa (che può avvenire in contemporanea con la definizione del flusso dell'esperienza utente) che deve essere anch'esso minimale, con pochi colori, font "neutri" e, perché no, anche spazi vuoti così da far sì che "tutto respiri"¹⁰.

Citiamo esempi illustri quali Facebook e Twitter, la cui scelta di colori quali il blu e l'azzurro è volta a rendere rilassante l'esperienza nelle due piattaforme, e Apple con un restyling di font per il nuovo iOS7 (con l'approdo al Arial, un font tra i più minimali).

Bess Ho, sviluppatrice e UI designer, afferma che in un'app di successo almeno il 30-40% del tempo deve essere investito nel design.

L'ultima cosa da considerare in questa fase è il rendere l'app specifica nelle funzioni: applicazioni con troppe features risultano spesso confuse, dispersive e non danno una buona impressione all'utente.

Attenzione, però, a non cadere nell'errore opposto cioè nel non popolare a dovere l'applicazione di contenuti: l'esperienza utente che termina con un "fail" (cioè una schermata dove non ci sono contenuti o i contenuti non sono quelli che l'utente si aspetta) sono frustranti per l'utente.

Una volta trovata la "nicchia commerciale" all'interno della quale l'app si sviluppa, concepito il flusso dell'esperienza dell'utente e definita la grafica, si passa alla scrittura del codice vero e proprio: come già anticipato nell'introduzione, il codice sorgente si sviluppa all'interno di un IDE attraverso un linguaggio di programmazione.

A questo livello c'è ben poco da dire, il grosso del lavoro è già stato fatto; fondamentale in questa fase è l'organizzazione del team di sviluppo: suddividere il flusso in macro gruppi e

⁹ Instagram è un'applicazione sviluppata da Kevin Systrom e Mike Krieger che permette di scattare e condividere foto all'interno dell'applicazione stessa.

È stata recentemente comprata dal colosso dei Social Network Facebook che l'ha integrata nei suoi servizi di condivisione.

¹⁰ Citazione dall'intervista di Uday Gajendar a Dice.

assegnare la scrittura del codice a diverse persone aiuta a scrivere applicazioni in tempi ridotti.

Importante fissare delle “mile-stone”¹¹ in cui viene fatto il punto dei progressi dell’applicazione.

Infine la fusione di tutti i pezzi darà l’applicazione conclusa.

Spesso non considerata come un vero e proprio momento di sviluppo ma importante almeno quanto gli altri, è il testing/debugging: questo livello consiste nel provare l’applicazione in tutti i suoi aspetti e scovare “bug”¹² o errori di funzionamento del codice. Spesso coloro che provano l’applicazione (i così detti “beta tester”) sono persone esterne al team di sviluppo, ma potrebbero anche essere gli stessi programmatori che testano in prima battuta la loro app.

2.2 Osservazioni tecniche sullo sviluppo del codice in un Hackathon e confronto con lo sviluppo “classico”

Le regole per lo sviluppo di un’applicazione vengono sfidate durante competizioni come l’Hackathon. Come già accennato, l’Hackathon è una manifestazione in cui i partecipanti, divisi in gruppi, devono consegnare entro il termine di tempo stabilito (da 24 ore ad una settimana) la soluzione, in formato hardware e/o software, ad un problema postogli.

La parola chiave durante un Hackathon è “tempo”: 24 o 48 ore sulla carta possono sembrare abbastanza ma quando lo sviluppo di un’applicazione “in ufficio”, seguendo puntualmente i passi sopra elencati, può impegnare anche 3-4 mesi, 24 ore passano in un attimo.

La progettazione di un’app durante un Hackathon non si discosta molto dalla progettazione fuori di esso, ma a causa del ridotto tempo a disposizione, ci sono alcuni importanti paletti da rispettare.

Partendo dal presupposto che i punti sopra citati siano fondamentali anche per lo sviluppo di un’app durante un Hackathon, ripercorriamoli puntualizzando alcune cose:

1. Definizione dei requisiti

I requisiti rimangono, comunque, un punto cardine sia per rimanere aderenti al tema assegnato sia per avere ben chiaro cosa la nostra app dovrà fare.

Dato il tempo stringente, infatti, la nostra app non può contenere troppe funzionalità o funzionalità troppo complesse o lunghe da implementare. È fondamentale, in questa occasione più che mai, trovare quella famosa “nicchia commerciale” che abbiamo citato prima, vista ora come il concentrarsi su una specifica e ben delineata strada verso la soluzione del problema.

¹¹ Mile stone (in italiano, pietra miliare) è un termine utilizzato in discipline quali Ingegneria del Software e Project Management che indica importanti traguardi intermedi nello svolgimento del progetto.

¹² Bug (in italiano, insetto) è un errore nel codice che porta ad un malfunzionamento o ad un “crash” (interruzione) del programma stesso.

Il termine deriva, probabilmente, da un episodio risalente al 1947 quando un malfunzionamento di un pionieristico PC (un Mark II) fu causato da una falena incastrata tra le valvole del dispositivo.

Il tecnico rimosse l’insetto e annotò sul registro del computer: “1545. Relay #70 Panel F (moth) in relay. First actual case of bug being found”.

Da qui il termine bug, per identificare un errore, e debug, cioè la sua rimozione.

2. Progettazione del flusso del codice

Questo livello rimane importante ma può essere considerato come meno stringente rispetto allo sviluppo “in ufficio”. Infatti il prodotto che verrà creato in un Hackathon non è un prodotto finito e pronto per la commercializzazione; è, invece, una versione beta¹³ che vuole dare un’idea alla giuria di come il team ha affrontato e risolto il problema.

3. Design dell’applicazione

Come per il flusso del codice, anche il design non è di importanza fondamentale per la riuscita di una buona applicazione durante l’Hackathon anche se un applicativo curato esteticamente risulta più gradevole di uno più scarso di grafica.

La mia personale esperienza porta a dire che bisogna progettare una discreta grafica anche per un’app presentata in una manifestazione del genere ma questo solo perché, nel mio team, sono sempre stato affiancato da un ottimo grafico che, in tempi brevi, è riuscito a produrre design degni di nota.

4. Scrittura del codice sorgente

Mai come in una competizione a tempo l’organizzazione del lavoro e la suddivisione dei compiti è fondamentale.

Un buon team è composto da non più di 4 persone che devono essere esperte nel numero più ampio possibile di settori. Questo favorisce non solo l’adattabilità del team a qualsiasi situazione (non si conosce a priori il tema e le piattaforme a disposizione durante la manifestazione) ma anche la possibilità di dividere il lavoro in settori ben delineati.

Le milestone, tranne per maratone molto lunghe, sono inutili: il codice viene unito poco prima della scadenza per una breve sessione di testing.

In conclusione la programmazione all’interno di un Hackathon non si discosta molto da quella all’esterni di una competizione. L’accento va messo sulla questione tempo che accentua concetti, comunque già definiti, come “semplicità”.

¹³ La versione beta di un’applicazione o un modulo software è una versione pre-release (precedente al rilascio commerciale) consegnata normalmente ai beta tester per i test.

3. Presentazione dell'Hackathon "ICT Days 2013"

La maratona Hackathon presa in esame si è svolta durante gli ICT Days 2013 presso l'Università di Trento: la manifestazione ha avuto l'obiettivo di far incontrare studenti universitari e mondo del lavoro e della ricerca.

La maratona è stata organizzata sia dal Comitato delle Universiadi, il quale ha messo in palio la volontà di utilizzare l'app vincitrice durante le Olimpiadi Universitarie Invernali 2013, ma anche da SmartCampus, spin-off dell'azienda TrentoRise, che sviluppa applicazioni per dispositivi mobili in collaborazione, tra le altre, con FBK (Fondazione Bruno Kessler), Provincia di Trento e Università di Trento.

Per lo sviluppo dell'app "Universiadi 2013" era necessario conoscerne i requisiti. SmartCampus e il Comitato Organizzatore delle Universiadi forniscono tutta la documentazione necessaria per conoscere i requisiti che l'applicazione doveva soddisfare. La descrizione degli stessi è spiegata nei paragrafi § 3.2. Inoltre l'app doveva essere popolata di informazioni estratte da una base di dati la cui struttura è definita nel paragrafo § 3.1.

3.1 Piattaforme e servizi per l'Hackathon "ICT Days 2013"

Come struttura di supporto alla progettazione delle applicazioni durante l'Hackathon "ICT Days 2013" è stata fornita da SmartCampus una piattaforma di servizi suddivisi in due gruppi: Core Services e Storage Services.

- Core Services

Rappresenta la parte di servizi interni e si suddividono ulteriormente in:

1. Servizi per mobilità ed eventi a Trento
2. Servizi di autenticazione
3. Servizi per la gestione del flusso dati

La prima tipologia fornisce informazioni su eventi e mobilità a Trento. Gli eventi sono oggetti con descrizione, data e ora, geolocalizzazione¹⁴ e altre informazioni utili. Per quanto riguarda la mobilità, i servizi interpretano due punti GPS passati come parametro e restituiscono le indicazioni stradali all'interno di una mappa. Le indicazioni possono essere filtrate secondo tempi di percorrenza, mezzi utilizzati o altro.

La seconda tipologia gestisce l'autenticazione degli utenti attraverso username e password. All'utente viene data la scelta se autenticarsi tramite il proprio account Google, FBK o UniTN¹⁵. In tutti i casi viene restituito un token di sessione¹⁶ che permette l'accesso a tutti gli altri servizi della piattaforma SmartCampus.

¹⁴ La geolocalizzazione è l'identificazione della posizione geografica nel mondo reale di un dato oggetto, come ad esempio uno smartphone o un computer. Le tecniche per la geolocalizzazione si basano sia su reti di satelliti GPS, che sulla rete cellulare GSM che sulla rete Internet.

¹⁵ Account personale rilasciato dall'azienda Fondazione Bruno Kessler (FBK) oppure dall'Università di Trento.

¹⁶ Il token di sessione non è altro che una stringa alfanumerica che rappresenta univocamente la sessione dell'utente da cui si può risalire, in senso inverso, a tutti i dati dell'utente loggato.

La terza tipologia fornisce strumenti per la gestione del flusso dati per le notifiche push¹⁷: viene registrato il codice ID del telefono nei server e permette l'invio e la ricezione di notifiche dagli utenti registrati.

- **Storage Services**

Rappresenta i servizi per l'archiviazione di dati. I servizi si poggiano su database di più tipo, SQL¹⁸ e NoSQL¹⁹. Viene data, inoltre, la possibilità di interfacciarsi con il framework di Dropbox²⁰ per una condivisione dei dati con altri utenti.

Sono state progettate 4 librerie specifiche per ogni servizio del server. Esse ricoprono ("wrappano") le funzionalità dei servizi stessi.

Ne esiste, inoltre, una quinta per la gestione delle chiamate da e per il server. Essa (il "protocol carrier") maschera le chiamate di tipo REST in un oggetto e ritorna oggetti JSON che verranno poi analizzati.

Tutta questa infrastruttura era stata utilizzata anche prima della maratona all'interno di SmartCampus per varie applicazioni: citiamo ViviTrento che si prefigge di aiutare gli abitanti del capoluogo trentino a vivere la città in modo tecnologico. Informa gli utenti sugli eventi, diurni e notturni, fornisce mappe con viabilità e pianificazione degli spostamenti. Anche gli studenti sono coinvolti nel progetto SmartCampus: citiamo StudyMate, un applicativo che gestisce la vita studentesca, accademica e non, a 360 gradi.

3.2 Requisiti richiesti per applicazione sviluppata durante Hackathon "ICT Days 2013".

Prima che la maratona cominciasse, ci vennero fornite le specifiche che l'applicazione doveva avere; data la concomitanza all'interno dell'Hackathon di due entità quali il Comitato e SmartCampus, le specifiche forniteci furono di due tipi: le prime di tipo contenutistico, cioè cosa l'applicazione deve rendere visibile all'utente, le seconde di tipo strutturale, cioè requisiti tecnici sullo sviluppo.

1. *Requisiti di tipo contenutistico (committente: Comitato Organizzatore)*

Le Universiadi, come del resto tutte le grandi manifestazioni, portano all'interno del paese ospitante (in questo caso la regione ospitante, il Trentino) due grandi tipologie di persone:

¹⁷ La tecnologia push rappresenta un tipo di comunicazione dove la stessa è iniziata da un server centrale. Le notifiche push sono un metodo con cui le applicazioni avvertono l'utente di un evento.

¹⁸ SQL (Structured Query Language) è un linguaggio standardizzato per database basato sul modello relazionale.

¹⁹ NoSQL (database) è una base di dati implementata non seguendo l'algoritmo relazionale tipico dei database SQL. Questi archivi di dati tipicamente non richiedono uno schema fisso (schemaless).

²⁰ Dropbox è un software di cloud storage multipiattaforma. Offre servizi di file hosting e sincronizzazione automatica dei file tramite web.

gli ospiti (categoria che comprende i visitatori, gli atleti, le delegazioni) e lo staff organizzativo.

Ogni gruppo porta con sé, a sua volta, problemi logistici:

- Per il gruppo degli ospiti gli ostacoli da prendere in considerazione sono, fondamentalmente, gli alloggi e gli spostamenti.

È stato calcolato che le Olimpiadi Universitarie 2013 porteranno nella regione trentina quasi 85.000 spettatori e altri 2000 sportivi. Il problema della logistica sia per quanto riguarda l'alloggio che per gli spostamenti, è un problema da prendere in seria considerazione.

Il Comitato Organizzatore ha ritenuto fondamentale che un'applicazione indirizzata al gruppo degli ospiti deve presentare informazioni sugli alberghi, sui ristoranti e sulle fermate dei mezzi pubblici; essa deve dare la possibilità di creare un planning del soggiorno dell'utente su, almeno, due livelli:

1. Organizzare la giornata di manifestazione

A questo livello l'utente può scegliere a quali gare o a quali eventi vuole partecipare e l'app restituisce l'itinerario migliore per vedere tutto quello che l'utente desidera, sia utilizzando mezzi pubblici che mezzi privati.

2. Organizzazione dell'intera manifestazione

A questo livello l'utente può pianificare l'esperienza in Trentino ancora prima di arrivarci: gli vengono consigliati gli hotel più vicini agli stadi in cui si svolgono le competizioni che lui segue maggiormente o quelli meglio serviti dai mezzi pubblici o, ancora, secondo qualsivoglia altra caratteristica che lui ritiene fondamentale.

Un altro requisito sono sicuramente le informazioni della manifestazione stessa: un calendario gare coronato da un metodo per tracciare un percorso tra il punto in cui si trova l'utente e il punto gara. Non possono mancare informazioni di contorno quali gli atleti partecipanti a quella disciplina, una breve storia dello sport stesso e altri dettagli di "costume".

Non un requisito stringente ma sicuramente utile è il social: nell'era del Web 2.0²¹ un'app non può mancare della sua parte sociale, la possibilità di condividere la manifestazione a cui si sta assistendo o il condividere foto e contenuti multimediali.

Questo non è stato identificato come un vero e proprio requisito ma più come una caratteristica da inserire perché immancabile in una qualsiasi applicazione odierna; effettivamente tra i progetti presentati al termine della maratona, nessuno includeva i social network.

- Per il gruppo dello staff organizzativo, da sempre non considerato come target per applicazioni per dispositivi mobili all'interno di manifestazioni come Universiadi ma con un bacino di quasi 1500 utenti, un'app per la gestione del loro lavoro sembrava essere apprezzata dal Comitato.

Proprio per questa noncuranza verso questo tipo di target i requisiti che ci sono stati dati erano molto vaghi e lasciavano ampio spazio di manovra.

²¹ Con il termine Web 2.0 si identifica un'evoluzione del World Wide Web le cui caratteristiche sono piattaforme online che creano reti sociali e di interazione tra siti e utenti.

Servizi web che hanno "tragheggiato" Internet nel 2.0 sono sicuramente Facebook, Twitter e i social media in genere ma anche siti quali Youtube, Vimeo (social video) e Flickr (social photo).

Inoltre la possibilità che non tutti i 1500 membri dello staff possedessero uno smartphone su cui visualizzare l'app paventava la possibilità che le features per gli operatori non fossero mai utilizzate.

Due cose, però, sono state, altresì, chiare: un'app che gestisce il lavoro dello staff doveva risolvere i problemi di ticketing e di sicurezza.

1. Ticketing

Un'esigenza in cui si incorre durante una qualsiasi manifestazione è la risoluzione dei problemi: dal guasto elettrico al malore di uno spettatore, le calamità devono essere risolte nel minor tempo possibile.

2. Identificazione personale

I sistemi classici di riconoscimento tramite codice a barre o tramite riscontro visivo attraverso un tesserino di riconoscimento sono metodi ottimi ma spesso fallimentari a causa di rotture, deterioramento o smarrimento del badge stesso.

I requisiti, quindi, sono tanti ed eterogenei e il Comitato Organizzatore ci diede libertà di scelta su quale dei due gruppi incentrare l'app e su quale problema essa avrebbe dovuto risolvere.

2. Requisiti di tipo strutturale (committente: SmartCampus)

I requisiti di tipo strutturale, forniti dall'azienda SmartCampus, riguardano il lato tecnico ed implementativo dell'applicazione.

In particolare, l'unica limitazione impostaci è stata quella dello sviluppo dell'app per smartphone Android, causa problemi con il fatto che il codice sorgente dovesse essere open source. Da notare che i sistemi operativi come Android, iOS e Windows Phone sono disponibili in varie versioni; SmartCampus non ha ritenuto necessario limitare la compatibilità dell'applicativo a una o più versioni ma ha demandato gli sviluppatori alla scelta della versione.

4. Interpretazione e implementazione dell'applicazione “Universiadi 2013”

Una volta presa visione dei requisiti, il mio team ha impostato la competizione (della durata di 48h) in questo modo:

1. Interpretazione dei requisiti richiesti dal Comitato Organizzatore delle Universiadi (§ 4.1)
2. Progettazione del flusso dell'esperienza utente, della grafica e implementazione del codice (§ 4.2)

4.1 Interpretazione del progetto “Universiadi 2013”

Dopo lo studio dei requisiti, la necessità risultata lampante a me e al mio gruppo è stata quella di creare un sistema intuitivo, di facile consultazione e che prevedesse comunicazioni rapide, sfruttando le potenzialità offerte dai moderni smartphone.

Analizzando i requisiti di tipo contenutistico, la scelta è ricaduta nel risolvere i problemi di entrambi i gruppi di utenti.

Per lo staff organizzatore ci siamo concentrati sulle seguenti problematiche:

- La turnistica dei volontari.
- Un FAQ²² interattivo (che simula un'intelligenza artificiale molto rozza) per la risoluzione delle principali problematiche poste agli attachè.
- La gestione del parco auto interno alle Universiadi.
- Un risolutore di problematiche a vario livello di gravità (probabilmente il più interessante e importante tra le features proposte).

Questo gruppo di funzionalità saranno raggruppate sotto il nome di “strumenti per lo staff” all'interno del paragrafo § 4.2.

Per il gruppo degli ospiti è stato scelto di ricoprire le problematiche legate alla logistica:

- Eventi in calendario
- Eventi per tipo di sport
- Eventi per luogo di svolgimento

Tutti gli eventi comprendono descrizione, data e ora, luogo di svolgimento e geolocalizzazione.

A questo proposito abbiamo pensato di implementare un sistema di planning del viaggio giornaliero dall'alloggio fino ai campi gara scelti.

Questo gruppo di funzionalità saranno raggruppate sotto il nome di “sezione informativa” all'interno del paragrafo § 4.2.

La stessa caratteristica è visualizzabile anche dal gruppo staff organizzatore che deve essere sempre informato sugli eventi che si stanno svolgendo o che si svolgeranno a Trento e dintorni.

Analizzando i requisiti di tipo strutturale è stato scelto di implementare l'applicazione su dispositivi Android con dimensioni dello schermo da 4.0” in avanti e con sistema operativo con versione 2.1 in avanti.

La scelta dello schermo è stata dettata dal fatto che schermi più piccoli avrebbero reso

²² FAQ (Frequently Asked Question) è un elenco di domande maggiormente poste con relative risposte. Il tipo di domande dipende dal contesto in cui il FAQ è inserito.

difficile la fruizione di contenuti, mentre scegliere versioni del SO precedenti avrebbe impedito l'utilizzo di funzioni native dell'ADT²³ (in particolare funzioni grafiche).

Avendo chiara la struttura dell'applicazione il team è stato suddiviso per competenze:

- Mattia Scagno, sviluppatore iOS e Android, responsabile dell'implementazione del codice sorgente
- Jacopo Penzo, sviluppatore iOS e UI Designer, responsabile della grafica e dell'esperienza utente

4.2 Implementazione del progetto “Universiadi 2013”

Le funzionalità sopra descritte non sono state tutte implementate durante il corso della maratona. Nonostante questo, grazie alla vincita nella maratona stessa, il mio team ha avuto modo di completare le restanti funzionalità all'interno di uno stage durato 2 mesi presso SmartCampus.

4.2.1 Implementazione del progetto “Universiadi 2013” durante l’Hackathon

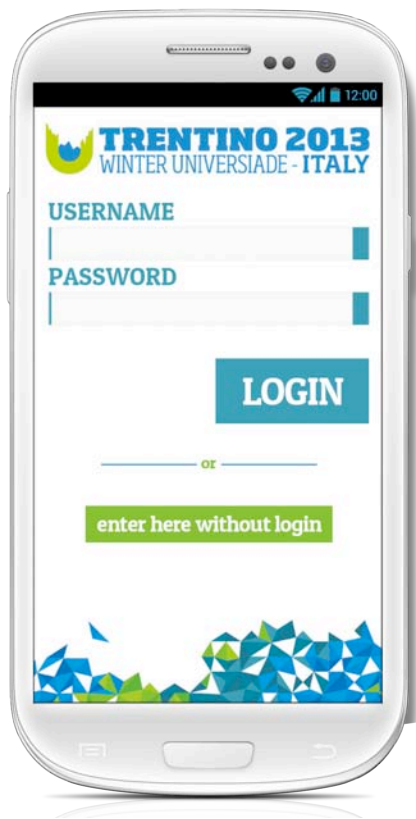
In questo paragrafo si descrivono le funzionalità dell'applicativo così come è stato implementato durante le 48 ore dell'Hackathon. I dati che vengono mostrati in questa sezione sono richiamati da metodi del backend di SmartCampus (vedi § 3.1) già esistenti prima che la maratona cominciasse.

L'applicazione prevede due sistemi di accesso in modo da poter rendere fruibile i contenuti e le funzionalità sia agli “addetti ai lavori” che agli utenti spettatori.

Il primo, riservato allo staff organizzativo si basa su accesso tramite credenziali (username e password) messo a disposizione dall'Organizzazione delle Universiadi; tutti questi dati sono archiviati in un database in possesso del Comitato che contiene, inoltre, le anagrafiche e le generalità dei membri dello staff.

Il secondo, pensato per l'utente spettatore, è un accesso senza credenziali: esso, ovviamente, permette la fruizione di una minore quantità di funzionalità.

²³ ADT (Android Development Tools) è un plugin compatibile con i maggiori IDE per la programmazione di applicazioni Android. Mette a disposizione librerie e framework per lo sviluppo, documentazione e emulatori di dispositivi che montano SO Android.



(Fig. 1)

Apertura dell'applicazione (Fig. 1):

La scelta è tra

- 1.L'accesso tramite username e password e quindi la possibilità di utilizzare tutte quelle features riservate allo staff.
- 2.L'accesso come utente ospite e visualizzare solo informazioni generiche sulla manifestazione.

Login:

Il Comitato Organizzatore, come già accennato in precedenza, possiede una serie di dati, anagrafiche, e generalità sui singoli volontari; a questo sono state associate anche delle credenziali di accesso all'area riservata dell'applicazione. Per motivi di sicurezza, i dati sensibili, quali sono le credenziali di autenticazione, non possono passare attraverso server terzi. In questo caso è stato creato un canale diretto tra i server dell'Organizzazione e la nostra applicazione per l'autenticazione dell'utente: il processo si basa sull'invio di login e password ai server dell'Organizzazione e la risposta degli stessi attraverso un token di autenticazione. Inoltre il token viene mantenuto attivo dal server che lo ha generato per un periodo di tempo limitato (sessione), scaduto il quale è necessario rigenerarlo.

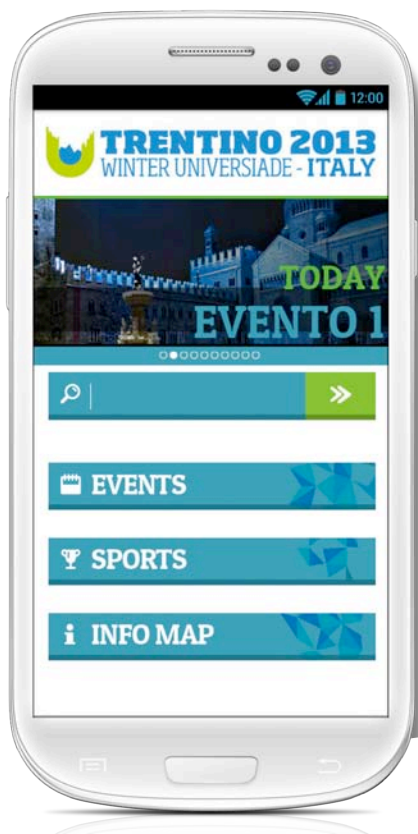
A seconda del tipo di accesso dell'utente (con o senza credenziali) l'esperienza utente cambia.

Sezione informativa:

La sezione informativa è visualizzabile sia dall'utente loggato (quindi membro del gruppo staff) sia dall'utente non loggato (quindi membro del gruppo ospite).

Essa contiene dati relativi agli eventi della manifestazioni, categorizzati a vario titolo, e altre informazioni utili per lo spettatore della Manifestazione.

Questi dati, non essendo sensibili in alcun modo, possono essere prelevati dai server del Comitato Organizzatore attraverso dei wrapper che fungono da buffer, e in parte minima da filtro, tra i server sopra citati e i server SmartCampus, dove i dati risiedono per una fruizione più rapida da parte dell'app.



(Fig. 2)

I dati richiesti in questa sezione (Fig. 2) sono per lo più gli eventi sportivi della manifestazioni, i meeting e le riunioni dello staff dell'Organizzazione e i POIs (Point Of Interest) della manifestazione come hotels, parcheggi, ristoranti, biglietterie.

Le informazioni sono state collegate tra loro a doppio filo: nei dettagli evento viene visualizzato il luogo dove l'evento si svolge mentre alla selezione di un POI sulla mappa si può accedere agli eventi che si svolgono nel luogo stesso.

Le riunioni e i meeting dello staff, visualizzate solo nel caso di utente loggato con credenziali, vengono elencate nella Home attraverso uno slider insieme agli eventi del giorno (al click vengono mostrati i dettagli, Fig. 3-4).



(Fig. 3)



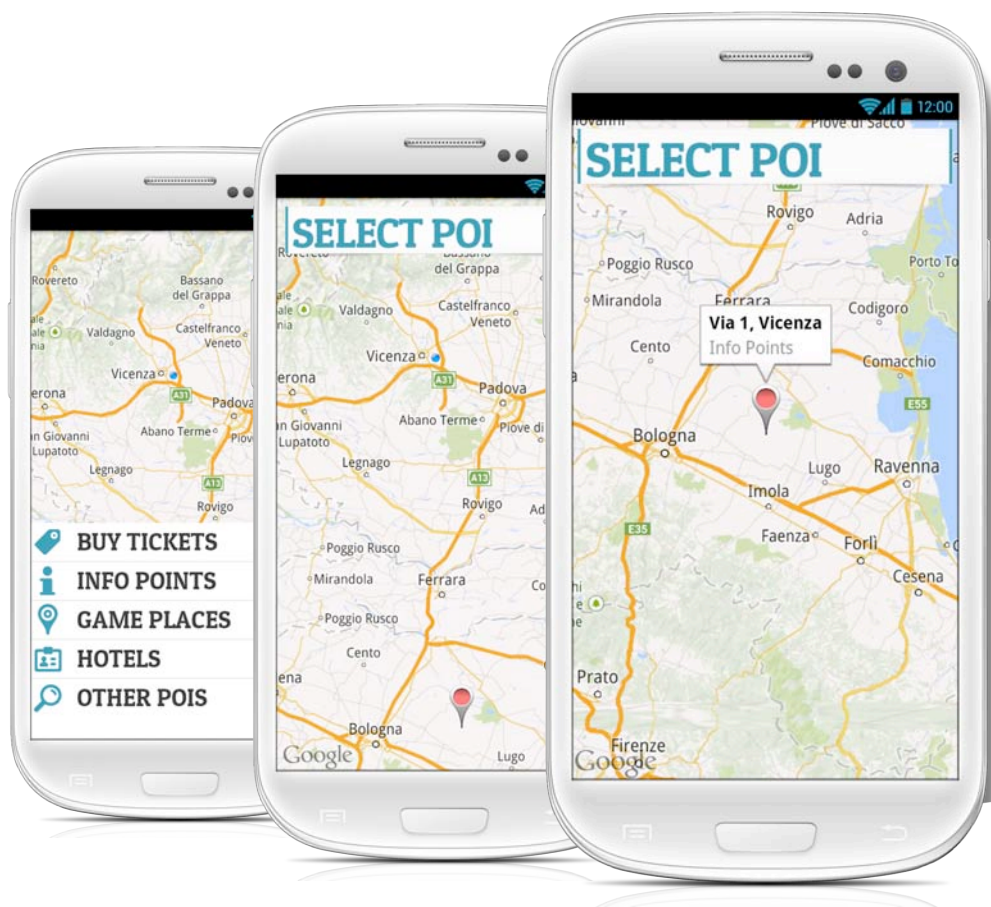
(Fig. 4)



(Fig. 5)

(Fig. 6)

- Gli eventi sportivi del giorno vengono mostrati nella schermata principale (la Home) attraverso uno slider. Accedendo alla sezione “Events” (Fig.5) o “Sports” (Fig. 6) si possono visualizzare gli eventi divisi per data o categoria di sport, rispettivamente; essi sono arricchiti da descrizione dell’evento, geolocalizzazione (e relativo percorso per raggiungerlo, grazie all’integrazione con l’applicazione “Mappe” di Android) e lista degli atleti partecipanti.

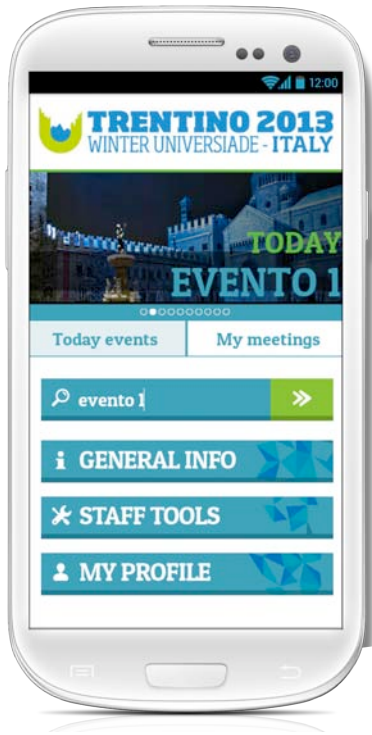


(Fig. 7)

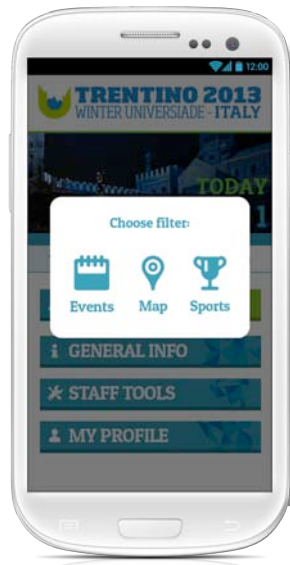
(Fig. 8)

(Fig. 9)

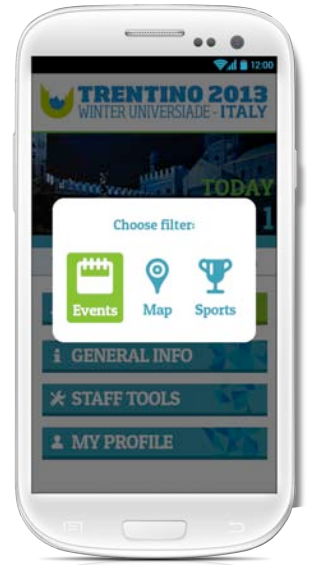
- I POIs sono mostrati nella sezione “Info Map” tramite una mappa a tutto schermo (Fig. 8-9).
Con un popup è possibile selezionare il tipo di POIs da visualizzare (Fig. 7).



(Fig. 10)



(Fig. 11)



(Fig. 12)

- Funzionalità presente in entrambe le versioni (utente loggato o ospite) è il “Search”: dopo l’inserimento di una o più parole chiave nell’apposito box di testo (Fig. 10), viene selezionata la categoria nella quale si vuole ricercare tra Eventi, POIs o Sports (Fig. 11, 12).

La ricerca delle parole chiave viene effettuata all’interno degli oggetti Evento, POI o Sport. Ad esempio se la categoria selezionata è “Evento”, la ricerca è effettuata nei campi “descrizione”, “data”, “nome evento”, “atleti partecipanti”.

Le entry che possiedono almeno un’occorrenza di una o più parole chiave vengono visualizzate tramite una lista.

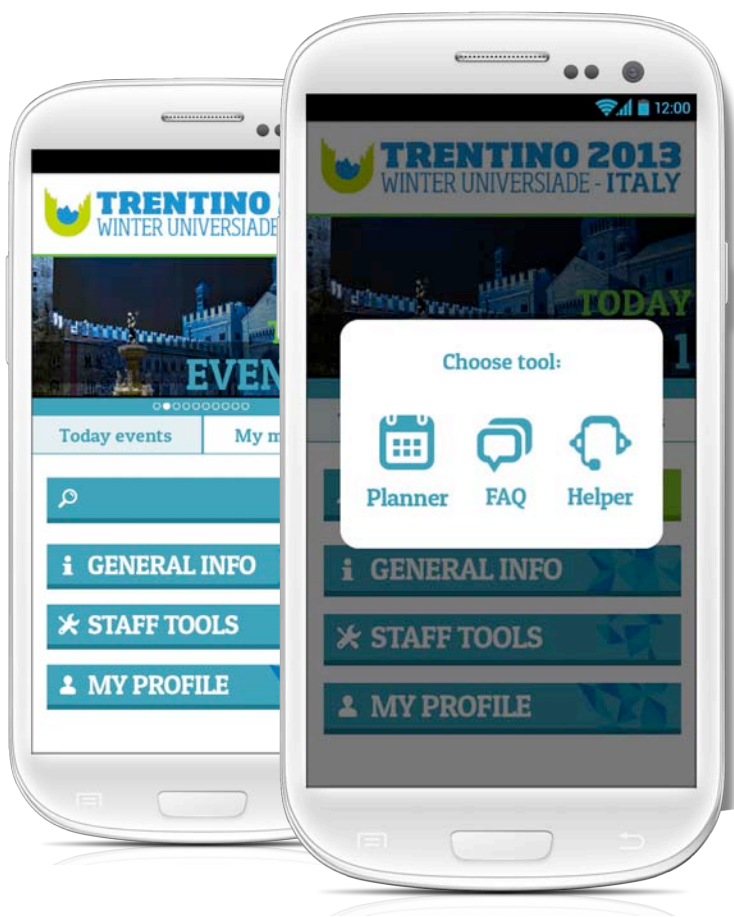
4.2.2 Implementazione del progetto “Universiadi 2013” durante stage in SmartCampus.

In questo paragrafo viene completata l’analisi dell’implementazione dell’applicazione “Universiadi 2013” dal momento in cui l’Hackathon si è concluso, e il progetto è stato presentato, a quando lo sviluppo è terminato.

I dati visualizzati in queste sezioni sono richiamati attraverso metodi del backend di SmartCampus (vedi § 3.1) creati ad-hoc per il nostro team dai programmatori di SmartCampus. Questi metodi sono stati implementati durante lo stage in accordo con le esigenze del mio team.

Strumenti per la gestione del personale:

Nel caso in cui l’utente possieda login e password per l’accesso (Fig. 13), l’applicazione si arricchisce di 3 ulteriori funzionalità riservate allo staff della Manifestazione (Fig. 14):



(Fig. 13)

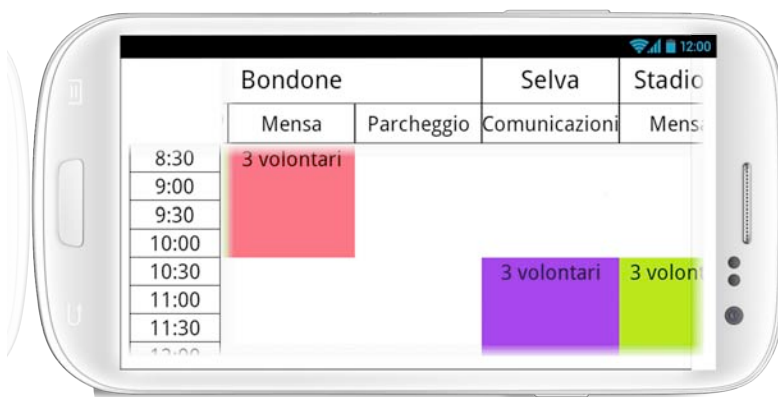
(Fig. 14)

- Calendario turni
- FAQ interattivo
- Gestore parco macchine (da implementare)
- Risolutore di problematiche

• *Calendario turni*

Organizzato ispirandosi al calendario Apple, esso richiede innanzi tutto l'immissione di uno o più filtri sui turni: filtro per data (obbligatorio), filtro per luogo dove si svolge il turno (facoltativo), filtro per categoria di appartenenza dei volontari (facoltativo).

Una volta selezionati i filtri, il tool mostra il calendario raggruppando in macro gruppi tutti i volontari che lavorano nella stessa fascia oraria (Fig. 15). Alla selezione di un macro blocco viene visualizzata la lista dei volontari che prestano servizio in quello slot orario, con relative informazioni di gerarchia, ruolo e categoria di appartenenza.



(Fig. 15)

- *FAQ interattivo*



(Fig. 16)

Studiato ispirandosi al servizio Siri (anch'esso di proprietà Apple), il FAQ interattivo prevede l'immissione, in via testuale e non vocale, di una domanda espressa in forma completa, e il conseguente ritorno di una lista di risposte in ordine decrescente di coerenza (concetto che verrà esplicitato in seguito) (Fig. 16).

Il sistema si basa su un database Mongo²⁴ con due tabelle: la principale è composta da record con tre campi, ID, domanda e risposta.

La secondaria associa a ciascun ID della tabella principale una serie di tag utilizzati per "riconoscere" la domanda dell'utente (Fig. 17).

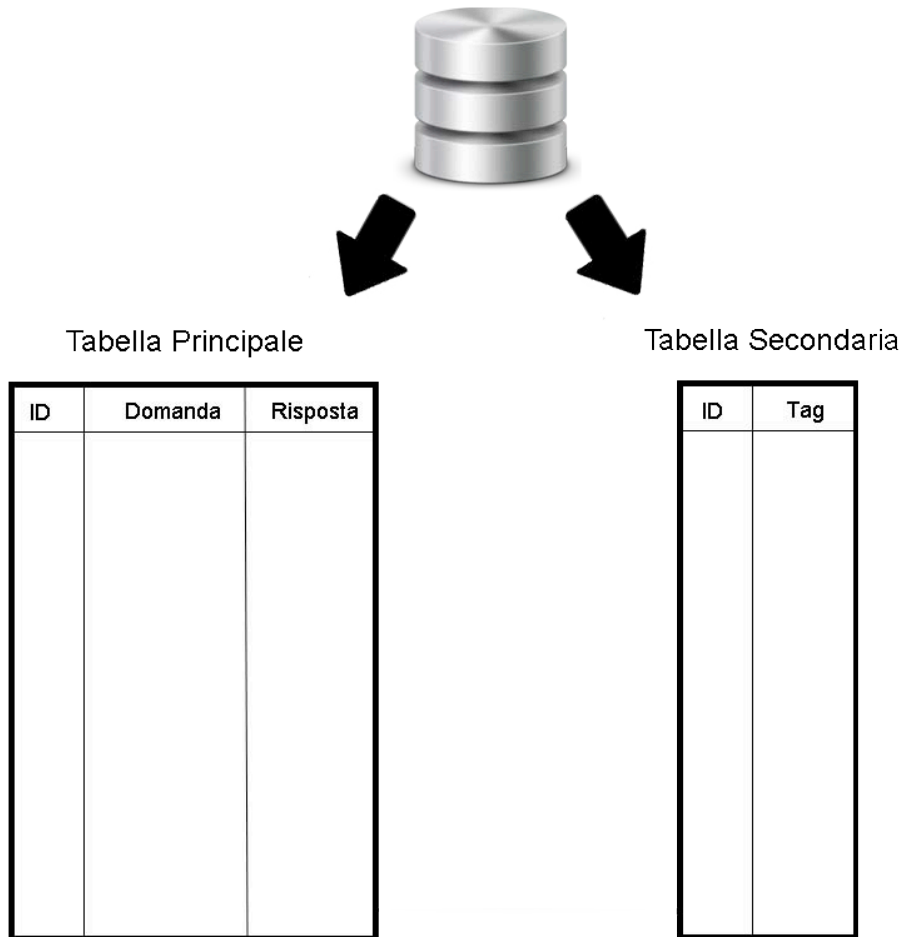
Una volta che la richiesta viene inviata, tramite una chiamata REST, al backend, la domanda viene scomposta in segmenti (token), ripulita dalle parole più corte, dalle preposizioni e dagli articoli; successivamente ogni singolo token viene confrontato con la colonna "tag" della tabella secondaria e si costruisce una mappa (struttura composta da coppie chiave-valore) che contiene tutti gli ID relativi al token preso in esame e il numero delle sue occorrenze. Terminata l'analisi della domanda si avrà a disposizione una mappa i cui elementi avranno come chiave l'ID della risposta e come valore il numero di token della domanda che hanno trovato una corrispondenza nei tag di quella determinata risposta (Fig. 18).

La percentuale di token della domanda che hanno una corrispondenza con i tag della risposta rispetto al numero totale di tag della risposta è stata definita *coerenza*.

Viene, quindi, ritornata la lista delle risposte con coerenza maggiore o uguale al 50%, in ordine decrescente di coerenza.

²⁴ MongoDB è un sistema gestionale di basi di dati non relazionale, orientato ai documenti, di tipo NoSQL. Il fatto che sia orientato ai documenti prevede che i record non siano salvati in tabelle con campi uniformi ma in documenti con determinate caratteristiche. La caratteristica NoSQL prevede che i database di tipo Mongo non seguano lo schema relazionale e quindi i dati, anche all'interno di una stessa tabella, possono essere strutturati in qualsiasi modo e non devono rispettare alcuna regola.

(Fig. 17)
(Schema della struttura del DB)



(Fig. 18)
 (Schema del funzionamento del FAQ interattivo)

Domanda: Dove sono i ristoranti a Trento?



Token: 0 : [Dove]
 1 : [sono]
 2 : [ristoranti]
 3 : [Trento]



Tabella Secondaria

ID	Tag
1,2,3,4	"dove"
1,5	"sono"
1,2	"ristoranti"
1,3	"trento"



Tabella Principale

ID	Domanda	Risposta
1	Domanda 1	Risposta 1
2	Domanda 2	Risposta 2
3	Domanda 3	Risposta 3
4	Domanda 4	Risposta 4
5	Domanda 5	Risposta 5



Return:
 0 : [Risposta 1, 100%]
 1 : [Risposta 2, 50%]
 2 : [Risposta 3, 50%]

Il sistema è da un lato semplice, come idea di fondo e come realizzazione, dall'altro complesso se pensiamo che per essere efficace deve contenere un numero di risposte (e relativi tag) molto grande. Da non trascurare il tempo di esecuzione che cresce al crescere del numero, non tanto delle risposte, quanto dei tag per ogni risposta.

• *Gestore del parco macchine:*

L'Organizzazione delle Universiadi mette a disposizione a figure illustri e delegazioni un parco macchine, con tanto di autisti, per gli spostamenti durante i giorni di manifestazione. La gestione di questo gruppo di macchine (che consta di circa 150 unità) è stato gestito in altre manifestazioni (vedi Campionato del Mondo di Sci Nordico svoltosi a Fiemme tra Febbraio e Marzo 2013) con un sistema web. Il sistema è tutt'ora attivo e verrà utilizzato anche nel corso delle Universiadi 2013.

Il nostro tool si prefigge di portare questo applicativo già presente nel web, in mobilità. Esso funziona come un carico/scarico di un magazzino: quando un driver prende in consegna una macchina, "blocca" il mezzo per una quantità di tempo desiderata e la "sblocca" una volta riportata all'interno del magazzino.

Il tool richiede la situazione delle macchine al server dell'Organizzazione attraverso un'API del backend e sincronizza dati (attraverso una chiamata POST²⁵, sempre del backend) quando un driver sceglie una macchina.

• *Risolutore di problematiche:*

Forse il tool più importante e che ha riscosso più successo tra i membri del Comitato è proprio il risolutore di problemi.

Esso si prefigge la creazione di un canale diretto tra colui che individua la problematica e colui che la risolve; il sistema si snoda su due livelli:

il primo è pensato per l'utente che incontra una gravità e non sa come fare per risolverla; il secondo è studiato per chi vede o è stato informato di una criticità e deve organizzare la risoluzione.



(Fig. 19)

Il flusso del tool parte, in qualunque caso, con la descrizione della criticità; viene chiesto all'utente di creare un "ticket" (un pacchetto del problema) composto da:

- Geolocalizzazione del luogo dove il problema è insorto
- Descrizione testuale del problema
- Categorizzazione del problema
- Foto del problema (facoltativa)

(Fig. 19)

²⁵ Chiamata REST che richiede la scrittura sul server (vedi nota 20)

Una volta completata la scheda del ticket, il flusso si snoda attraverso la scelta del canale più adatto per trasmettere il ticket:
 per il primo livello abbiamo la scelta se mandare una notifica al responsabile della categoria a cui il problema fa riferimento (un guasto elettrico verrà recapitato al Responsabile del reparto elettrico della venue) o una notifica direttamente all'HelpDesk creato dall'Organizzazione proprio per fronteggiare le problematiche.
 Per il secondo livello, grazie ad un gateway SMS messoci a disposizione dal Comitato, viene data la possibilità di inviare un SMS multiplo ad un numero qualsiasi di volontari, appartenenti alla categoria a cui il problema fa riferimento, per organizzare e coordinare la risoluzione del problema (Fig. 20).



(Fig. 20)

5. Conclusioni

5.1 La fine dell'Hackathon: presentazione e sviluppi successivi

Al termine delle 48 ore i progetti realizzati sono stati presentati: il pitch²⁶ consisteva in 10 minuti di esposizione delle funzionalità del progetto, coadiuvate da screenshot dell'applicativo e prova "in diretta" del funzionamento.

La nostra presentazione è stata impostata nel mostrare l'attinenza del nostro prodotto ai requisiti richiesti: lo scheletro grafico e concettuale di tutta l'applicazione era definitivo. Il codice implementato, invece, copriva tutta la sezione informativa e il login iniziale.

La giuria (composta da Marco Pistore, ricercatore senior all'interno della fondazione FBK, Andrea Molinari, responsabile ICT per il Comitato Organizzatore e altri ricercatori senior di TrentoRISE) reagì con interesse sia all'idea di rendere unica l'applicazione per spettatori e staff organizzatore, sia a come sono state gestite alcune criticità quali riconoscimento dell'utente, ticketing ecc.

Resisi conto delle potenzialità dell'applicativo e della nostra, se pur minima, esperienza nel campo dello sviluppo app²⁷, io e il mio team fummo decretati vincitori dell'Hackathon.

Il premio, oltre ad una vincita in denaro, consisteva nel continuare lo sviluppo dell'app all'interno dell'azienda SmartCampus: all'interno della struttura, coadiuvati da programmatori senior esperti in vari ambiti informatici tra cui programmatori Android, programmatori di backend e grafici, abbiamo concluso lo sviluppo con le rimanenti tre features.

Importante è stato il confronto, anche se tardivo, con una delegazione dello staff che andrà ad utilizzare l'applicazione (il famoso "target" citato § 2.1): i membri hanno accolto favorevolmente molte delle nostre proposte (in particolar modo il "risolitore di problematiche") e scartandone altre hanno creato un collegamento solido e costruttivo tra programmatori e utenti, tra domanda e offerta.

Ad oggi l'applicazione è conclusa; il lavoro finale è riservato al backend e al wrapping dei dati da server terzi.

L'applicazione sarà scaricabile da Google Play gratuitamente dai primi di Novembre, in tempo per l'inizio delle Universiadi che si svolgeranno dal 11 al 21 Dicembre 2013.

²⁶ Un "pitch" (termine mutuato dal baseball che indica l'azione di lancio fatta dal lanciatore, appunto) è una presentazione della propria startup. Esso è un documento fondamentale (insieme al business plan) per l'acquisizione di fondi da parte degli investitori.

Il termine è stato un po' estrapolato dal contesto e quando si parla della presentazione di un'applicazione spesso si parla di "pitch", anche se in senso lato.

²⁷ Più di un anno fa fondai assieme a un socio il CodeFish Studio la cui vocazione è lo sviluppo di applicazioni native per smartphone e siti web.

5.2 Conclusione del lavoro di tesi

Mai avrei pensato che l'Università mi potesse dare così tante opportunità in così breve tempo; ma così è stato: grazie alla mia passione per la programmazione mi sono avvicinato a un ambito lavorativo dinamico, giovane e in continua crescita.

Come studio di programmazione il bilancio delle competizioni a cui abbiamo partecipato (tre, ad oggi) non può essere che positivo: oltre a conoscere nuovi punti di vista sullo sviluppo delle applicazioni, gli Hackathon e le esperienze lavorative che da lì sono cominciate (vedi tirocinio in SmartCampus) sono state un momento di crescita professionale e personale fondamentale che tutt'ora fa' parte del bagaglio che portiamo con noi in ogni progetto.

6. Bibliografia

- [1] Wikipedia (Internet), (consultato il 17 Settembre 2013).
Disponibile al sito: <http://www.wikipedia.org>
- [2] Corriere delle Comunicazioni (Internet). Veronica Benigno, “Gartner: app, giro d'affari da 25 mld di dollari” (pubblicato il 4 Marzo 2013, consultato il 17 Settembre 2013).
Disponibile al sito: <http://www.corrierecomunicazioni.it/>
- [3] Business Magazine (Internet). Paolo Corsini, “Oltre un miliardo di smartphone nel 2013” (pubblicato il 9 Settembre 2013, consultato il 17 Settembre 2013)
Disponibile al sito: <http://http://www.businessmagazine.it/news>
- [4] Intervista per il magazine “Dice” durante Silicon Valley Code Camp (Internet).
(Pubblicato il 18 Ottobre 2011, consultato il 17 Settembre 2013).
Disponibile al sito: <http://http://www.youtube.com>