

**ALMA MATER STUDIORUM - UNIVERSITÀ DI  
BOLOGNA  
SEDE DI CESENA**

**FACOLTÀ DI SCIENZE MATEMATICHE, FISICHE E  
NATURALI**

**Corso di Laurea Magistrale in Scienze e Tecnologie Informatiche**

# **Sistema integrato di crowdsourcing e data integration in contesto accessible smart city**

Relazione finale in:  
**Sistemi Multimediali**

Relatore:  
**Prof.ssa Paola Salomoni**

Presentata da:  
**Paolo Giovannini**

Correlatore:  
**Dott.ssa Catia Prandi**

**Sessione I  
Anno Accademico 2012/2013**

---

---

*Ai miei genitori, per esser stati un costante sostegno*

*durante questo lungo e difficile percorso.*

---

---

<b>Introduzione .....</b>	<b>9</b>
<b>Smart city: accessibilità e barriere architettoniche .....</b>	<b>13</b>
<b>1.1 Barriere Architettoniche e Accessibilità .....</b>	<b>14</b>
<b>1.2 Leggi e Normative .....</b>	<b>17</b>
1.2.1 Legge 13/89 .....	18
1.2.2 Legge 104/92 .....	19
1.2.3 D.M. 236/89 .....	19
1.2.4 DPR 503/1996 (art 2,7,10,13) .....	20
1.2.5 DPR 380/2001 (Art. 77-79) .....	21
<b>1.3 Smart City .....</b>	<b>23</b>
1.3.1 Descrizione .....	23
1.3.2 Digital City.....	25
1.3.3 Internet of Things .....	26
1.3.3.1 Tecnologie abilitanti.....	27
1.3.3.2 Applicazioni .....	27
1.3.4 Crowdsourcing.....	28
1.3.5 Participatory Sensing e Citizen Sensing.....	28
<b>1.4 Sistemi informativi territoriali .....</b>	<b>30</b>
1.4.1 Storia e utilizzo .....	30
1.4.2 Definizione .....	30
1.4.3 Struttura .....	31
<b>1.5 Data Integration .....</b>	<b>33</b>
1.5.1 Descrizione .....	34
1.5.2 Analisi e normalizzazione .....	35
1.5.3 Concetti comuni.....	36
1.5.4 Le fasi dell'integrazione .....	37
1.5.5 Pulizia dei dati.....	37
<b>1.6 Analisi e valutazione dei servizi esistenti .....</b>	<b>37</b>
1.6.1 Descrizione dei servizi.....	38
1.6.1.1 EASY WAY - VODAFONE .....	38
1.6.1.2 LIBERI DI MUOVERSI.....	39
1.6.1.3 MAP ABILITY .....	40
1.6.1.4 INGRESSO LIBERO .....	41
1.6.1.5 WHEELMAP.....	42
1.6.1.6 COMUNI PER TUTTI .....	43
1.6.1.7 ACCESS TOGETHER .....	44
1.6.1.8 DECORO URABANO - WE DU!.....	45

---

1.6.2 Servizi a confronto .....	47
<b>Progettazione.....</b>	<b>49</b>
<b>2.1 Fusion Table.....</b>	<b>49</b>
2.1.1 Visualizzare online i dati .....	49
2.1.2 Costruire Mappe.....	50
2.1.3 Condividere e controllare i dati on line .....	50
2.1.4 Organizzare dati esterni .....	51
2.1.5 Collaborazione .....	51
<b>2.2 Foursquare .....</b>	<b>51</b>
2.2.1 Il perché di un successo .....	52
<b>2.3 Tecnologie utilizzate.....</b>	<b>53</b>
2.3.1 API.....	53
2.3.2 OAuth 2.0.....	54
2.3.2.1 Service Provider e Consumer .....	54
2.3.3 HTTPS e SSL .....	56
2.3.4 JSON .....	56
2.3.5 JavaScript .....	59
2.3.6 PHP .....	60
2.3.7 Google Maps.....	61
2.3.7.1 Caratteristiche .....	61
2.3.8 Database spaziali .....	62
2.3.9 Fusion Table API.....	63
2.3.9.1 Concetti di base .....	63
2.3.9.2 Operazioni base.....	65
2.3.9.3 Sending requests.....	65
2.3.9.4 Identificare l'applicazione .....	66
2.3.9.5 Utilizzare i dati Geografici.....	67
2.3.10 Foursquare API.....	68
2.3.10.1 Funzionamento .....	69
2.3.10.2 Strumenti disponibili .....	70
2.3.10.3 Real-Time API.....	70
2.3.11 Tecnologia Push .....	71
2.3.11.1 HTTP server push.....	71
2.3.11.2 Pushlet.....	72
2.3.11.3 Long polling.....	72
<b>2.4 Architettura .....</b>	<b>72</b>
2.4.1 Connessione a Foursquare .....	73
2.4.2 Richiesta di Check-in.....	74

---

2.4.3 Raccolta informazioni .....	75
2.4.4 Salvataggio informazioni .....	75
<b>Implementazione .....</b>	<b>77</b>
<b>3.1 Struttura Database .....</b>	<b>77</b>
<b>3.2 Integrazione.....</b>	<b>79</b>
<b>3.4 Creazione applicazione.....</b>	<b>80</b>
<b>3.5 Autenticazione e permessi in Foursquare.....</b>	<b>80</b>
<b>3.5 Applicazione connessa.....</b>	<b>83</b>
3.5.1 Utilizzo Server Push.....	83
3.5.2 Applicazione Real-Time .....	84
<b>3.6 Questionario.....</b>	<b>86</b>
3.6.1 Raccolta dei dati sul check-in .....	87
3.6.2 Domande.....	88
3.6.3 Salvataggio questionario .....	90
3.6.4 Inserimento Tip .....	90
<b>3.7 Scrittura su Fusion Table .....</b>	<b>91</b>
3.7.1 Autenticazione .....	92
3.7.2 Operazioni su Fusion Table.....	95
3.7.2.1 Selezione.....	95
3.7.2.2 Aggiornamento .....	95
3.7.2.3 Inserimento.....	96
3.7.2.4 Invio a Fusion Table .....	96
<b>3.8 Visualizzazione dei dati.....</b>	<b>96</b>
3.8.1 Google Maps .....	96
3.8.1.1 Stili e proprietà .....	96
3.8.1.2 Caricamento Layer da Fusion Table .....	98
<b>Conclusioni .....</b>	<b>101</b>
<b>Bibliografia.....</b>	<b>103</b>

---

---

# Introduzione

La qualità della vita delle persone dipende e sempre più dipenderà da quando gli agglomerati urbani sapranno conciliare e soddisfare le esigenze dei cittadini in numerosi campi: comunicazione, mobilità, ambiente ed efficienza energetica ne sono alcuni esempi. Diventa quindi una strada obbligata, la trasformazione delle attuali città in Smart Cities. Vale a dire centri urbani intelligenti, dove grazie alla tecnologia e alle informazioni ottenute dai cittadini e dalle infrastrutture tecnologiche stesse, si potrà agire attivamente per migliorare la vita dei cittadini.

Questa visione è ancora lontana dalla realtà. Un esempio ne è dato dalle barriere architettoniche che sono ancora presenti nei luoghi delle città impattando negativamente sullo svolgimento delle attività quotidiane da parte di persone con bisogni e necessità specifiche e quindi sulla loro vita. Diversi sono i casi, in città, di strisce pedonali con rampe per disabili solo da un lato della strada e/o di marciapiedi pieni di buche e avvallamenti proprio in prossimità di discese in strada. Se a questo si aggiunge l'insensibilità di molti cittadini che parcheggiano sulle strisce pedonali e davanti alle rampe, si capisce quanto sia lontana l'idea di Smart City. Proprio per questo i governi di molti paesi, e l'Unione Europea *in primis*, stanno finanziando programmi per lo sviluppo di sistemi intelligenti nelle città.

La nostra attenzione si concentra sull'accessibilità, ovvero la possibilità per persone con ridotta o impedita capacità motoria o sensoriale di raggiungere un edificio e le sue singole unità immobiliari e ambientali, di entrarvi agevolmente e di fruire di spazi ed attrezzature in condizioni di adeguata sicurezza e autonomia.

Il progetto sviluppato in questo volume vuole cercare di sensibilizzare il cittadino al problema delle barriere architettoniche, cercando di renderlo partecipe alla raccolta d'informazioni sugli impedimenti presenti nei luoghi visitati utilizzando semplicemente il proprio smartphone. Tutto ciò creando un portale web dove i dati raccolti possono essere cercati e visionati.

Il sistema integrato che abbiamo progettato e implementato vuole offrire l'opportunità di sapere in anticipo se, ad esempio, per entrare in un certo locale ci sono dei gradini oppure se è dotato di servizi accessibili: per alcune persone, queste informazioni sono di grande importanza nel momento in cui devono decidere se recarsi in un posto o in altro.

Data la presenza di diversi servizi web e mobile, che offrono soluzioni al problema delle barriere architettoniche si è pensato di effettuare un'operazione di data integration in modo tale da dare all'utente una visione globale di tutte le informazioni raccolte sulle barriere architettoniche. Vedremo come ogni servizio è differente rispetto agli altri per le tecnologie e metodi implementativi che sono stati utilizzati per la loro creazione. Con un'attenta analisi si è riusciti a integrare e i dati cogliendo informazioni importanti. Il progetto è nato come strumento per raccogliere, visualizzare e editare i dati sull'accessibilità. Utilizzando il crowdsourcing per collezionare le informazioni, consente a chiunque, non solo ai diretti interessati, di usare il proprio smartphone o tablet per descrivere e aggiornare la situazione di accessibilità di un preciso luogo. Infatti, non occorre essere esperti del tema e delle difficoltà connesse per partecipare alla raccolta di dati.

Per avere delle informazioni solide sui luoghi descritti dagli utenti, si è scelto di utilizzare un sistema già affermato chiamato Foursquare. Esso è un social network che si basa sulla condivisione della posizione geografica degli utenti attraverso un dispositivo mobile, come smartphone o tablet. Gli utenti possono eseguire il cosiddetto *check-in* nei locali, nei negozi, nei musei, in qualsiasi posto d'interesse pubblico o privato.

Il sistema che abbiamo sviluppato funziona nel seguente modo: per prima cosa, occorre dare all'applicazione il permesso di collegarsi al nostro account Foursquare; poi, dopo aver fatto check-in in un posto, compare un breve questionario da compilare tramite il dispositivo mobile, segnalando per prima cosa in che tipo di luogo ci troviamo. Le domande riguardano la disponibilità di servizi accessibili e altre caratteristiche, tutto questo avviene all'interno di Foursquare.

Le risposte sono inserite nel database e messe a disposizione degli utenti attraverso il sito web. Queste informazioni raccolte sono visualizzate tramite mappa geografica con dei marcatori, il colore del marcatore denota il livello di accessibilità del luogo: rosso non accessibile, giallo parzialmente accessibile e blu accessibile. Quindi un

---

utente può consultare la mappa per ritrovare le informazioni sui luoghi che gli interessano.

Questo documento di tesi è organizzato nel seguente modo. Nel primo capitolo sono introdotte tutte le leggi e le normative di legge presenti in Italia, in modo da mostrare un quadro generale del problema delle barriere architettoniche. Poi verrà descritto il concetto di Smart City e di come le tecnologie, la digitalizzazione e la sensibilizzazione dei cittadini permette ad una città digitale di diventare “smart”. Sono poi introdotti i Sistemi Informativi Territoriali che sono utilizzati dal nostro sistema per la rappresentazione dei dati, in modo da renderli disponibili per la consultazione. Verranno poi descritti i vari servizi web e mobile che raccolgono informazioni sulle barriere architettoniche per analizzare le tecnologie utilizzate e le loro basi di dati in modo da effettuare l’operazione di data integration.

Nel secondo capitolo si parla di progettazione del sistema, quindi tutte le tecnologie e le scelte architettoniche fatte. Inizia con la descrizione del servizio Google Fusion Table, dove sono salvate le informazioni raccolte. Si passa alla descrizione del servizio Foursquare, utilizzato per raccogliere i dati necessari. Poi si entra nel dettaglio con le tecnologie utilizzate per usufruire dei servizi web messi a disposizione da Google e Foursquare, per gestire l’autenticazione e lo scambio di dati in modo sicuro. È inoltre descritta l’architettura e come vengono utilizzate le tecnologie elencate per far sì che il progetto funzioni in modo adeguato.

Nel terzo capitolo sono illustrati i dettagli implementativi, quindi come sono state utilizzate le tecnologie, descritte nel capitolo 2, a livello tecnico per creare l’applicazione su Foursquare e il questionario che è mostrato all’utente per raccogliere le informazioni sulle barriere architettoniche. Per prima cosa c’è la descrizione di come è stata strutturata la base di dati che poi raccoglierà tutte le informazioni e come sono stati integrati i dati presenti nei servizi già esistenti. Sono poi mostrate tutte le procedure seguite per utilizzare il servizio di database di Google e le procedure utilizzate per interrogare e salvare le informazioni. Infine è descritto come è stato creato il sito web nel quale sono rappresentate le informazioni sotto forma di mappa geografica.



---

# Smart city: accessibilità e barriere architettoniche

In questo capitolo sarà introdotto il problema delle barriere architettoniche e saranno illustrate le normative, le leggi e le proposte di legge esistenti in materia presenti in Italia e in altri Paesi.

Vedremo come le tecnologie informatiche sono impiegate per creare il concetto di Smart City ovvero *città intelligente*. Un modello urbano in cui, grazie alle tecnologie digitali e a infrastrutture moderne, si cerca di migliorare la qualità della vita dei cittadini. Una città diventa *smart* potenziando i servizi di comunicazione, trasporto e rifornimento energetico, riducendo l'inquinamento atmosferico e cercando di rendere sostenibili tutte le attività cittadine. Infatti, l'obiettivo che guida un progetto in contesto smart city è proprio quello di cambiare in meglio la vita dei cittadini attraverso le tecnologie tipiche delle digital city.

Un ruolo importante in molte sperimentazioni è giocato dalla messa in rete di opinioni, cioè dalla condivisione d'informazioni fra più soggetti. Questa è l'idea su cui si basano molti progetti, tutti nati con l'esigenza di rendere una digital city più accessibile. Le informazioni geolocalizzate raccolte dagli utenti, sono raccolte principalmente tramite dispositivi mobili.

In questo capitolo sono descritte: cosa sono le barriere architettoniche e le leggi che le riguardano e come le norme italiane definiscono un luogo accessibile. È poi approfondito il concetto di smart city e come le tecnologie coinvolte possano essere sfruttate per rendere una città più accessibile. Infine viene fatta l'analisi di servizi web e mobile che raccolgono informazioni dagli utenti con lo scopo di rendere più vivibile la città con particolare riguardo alle persone con qualche tipo di disabilità, sia temporanee che permanenti.

## 1.1 Barriere Architettoniche e Accessibilità

Una barriera architettonica è un qualunque elemento costruttivo che impedisce o limita gli spostamenti o la fruizione di servizi, in particolar modo a persone disabili, con limitata capacità motoria o sensoriale. La barriera architettonica può essere una scala, un gradino, una rampa troppo ripida. Qualunque elemento architettonico può trasformarsi in barriera architettonica e l'accessibilità dipende sempre dalle caratteristiche personali della singola persona.

Ne è conseguita una non corretta applicazione della pur importante normativa vigente sull'argomento. Nei primi anni del 21° sec. la situazione è andata lentamente modificandosi, facendo emergere le vere esigenze delle persone. Ci si è cioè resi conto che le barriere architettoniche producono per molti individui riduzione del benessere, disagio, affaticamento e condizioni di pericolo. Si è iniziata a rifiutare l'idea che l'uomo debba adattarsi a una città piena di ostacoli, scomoda e defaticante. Per queste ragioni è cresciuta la richiesta di ambienti privi di barriera architettonica, facilmente accessibili, fruibili e in grado di risultare 'inclusivi'. Tutto questo in coerenza con quanto è stato affermato dall'OMS (*Organizzazione Mondiale della Sanità*), che definisce la salute "non come assenza di malattia, ma come stato di completo benessere psico-fisico e sociale".

È interessante elencare i vari passaggi storici che hanno portato alla situazione attuale. Il primo documento ufficiale italiano in cui viene citata la dizione *barriera architettonica* è la circolare del Ministero dei Lavori pubblici nr. 425 del 20 gen. 1967, relativa agli standard residenziali. Le barriere architettoniche vennero definite come “gli ostacoli che incontrano individui fisicamente menomati nel muoversi negli spazi urbani e negli edifici: ostacoli costituiti essenzialmente da elementi altimetrici che si incontrano lungo i percorsi (gradini, risalti, dislivelli, scale ecc.) ovvero da esiguità di passaggi e ristrettezza di ambienti (strettezze, cabine di ascensori, apertura di porte ecc.)”.

Il 19 giugno 1968 fu emanata la circolare del Ministero dei Lavori pubblici nr. 4809 "Norme per assicurare l'utilizzazione degli edifici sociali da parte dei minorati fisici e per migliorarne la godibilità generale". Era evidenziata la necessità di eliminare le barriere architettoniche, "essendo ormai accertato che tali barriere interessano circa 8 milioni di cittadini pari al 15% della popolazione italiana". Si precisava che le norme avevano come presupposto la generalizzazione dei vantaggi derivanti

dall'eliminazione di tali ostacoli, e che non si escludevano soluzioni più avanzate che sarebbero potute "essere esaminate ed approvate dai competenti uffici". Il primo cogente provvedimento legislativo, tuttora in vigore, è costituito dall'art. 27 della l. 3 marzo 1971 nr. 118. Esso prescriveva l'applicazione delle norme tecniche contenute nella sopracitata circolare 4809/68. Ampliava il campo di applicazione delle stesse a tutti gli "edifici pubblici o aperti al pubblico, alle istituzioni scolastiche prescolastiche o di interesse sociale di nuova edificazione [...] anche apportando le possibili e conformi varianti agli edifici appaltati o già costruiti all'entrata in vigore della presente legge".

Dopo circa sette anni fu emanato il d.p.r. 384/78 - regolamento dell'art. 27 della l. 118/71 - senza peraltro suscitare la dovuta attenzione, con conseguente scarsa applicazione. Nei successivi undici anni si è continuato a sottovalutare il problema. Si è fatto poco e male per far emergere il contenuto sociale dei provvedimenti e far rispettare la normativa tecnica prescritta dalla legge, con forti ricadute negative a livello macroeconomico.

Con l'emanazione della l. 9 genn. 1989 nr. 13, e del suo regolamento di attuazione, decreto del Ministero dei Lavori pubblici del decreto nr. 236 del 14 giugno 1989, il concetto di b. a. si è molto ampliato ed evoluto; va anzi registrata una vera e propria svolta, dovuta alla nuova definizione contenuta nell'art. 2 del decreto stesso. Infatti vi si legge: "Per barriere architettoniche si intendono: 1) gli ostacoli fisici che sono fonte di disagio *per la mobilità di chiunque* e in particolare di coloro che, per qualsiasi causa, hanno una capacità motoria ridotta o impedita in forma permanente o temporanea; 2) gli ostacoli che limitano o impediscono *a chiunque* la comoda o sicura utilizzazione di parti, attrezzature o componenti; 3) la mancanza di accorgimenti e segnalazioni che permettano l'orientamento e la riconoscibilità dei luoghi e delle fonti di pericolo *per chiunque* e in particolare per i non vedenti, per gli ipovedenti e per i sordi". Sono quindi considerati ostacoli da rimuovere o superare (con soluzioni alternative possibili): percorsi con pavimentazione sdruciolevole, irregolare o sconnessa, e/o con eccessiva estensione; scale prive di corrimano; rampe con forte pendenza o troppo lunghe; luoghi di attesa privi di protezione dagli agenti atmosferici o senza sistemi di seduta; attrezzature o terminali di impianti posizionati troppo in alto o troppo in basso eccetera.

Le barriere architettoniche vanno dunque considerate come una caratteristica negativa dell'habitat dell'uomo, riguardante la generalità dei cittadini. In particolare le 'utenze deboli' (anziani, bambini piccoli, incidentati, obesi, cardiopatici, artrosici,

ipovedenti ecc.), stimate in oltre il 20% della popolazione europea. Minore è la presenza di barriere architettoniche, maggiore risulta, con benefici generalizzati, l'*accessibilità urbana*. Quest'ultima va intesa come concreta possibilità da parte di una utenza ampliata di accedere agevolmente a spazi, edifici, attrezzature e sistemi di trasporto. Tale caratteristica qualitativa dell'ambiente edilizio e urbano riguarda il vivere quotidiano. A essa si collegano concetti e sensazioni importanti come il potenziamento delle scelte individuali e quello delle pari opportunità.

Per *accessibilità*, al di là dell'esigenza di poter disporre di spazi privi di barriere architettoniche, deve quindi intendersi *comfort ambientale*, in situazione di sicurezza e autonomia, anche per chi ha ridotte capacità motorie o sensoriali, per periodi brevi o in maniera permanente. Quest'aspetto deve assumere il giusto valore quando si affronta qualsiasi progetto.

Un progetto responsabile deve essere impostato, a partire dalle sue prime fasi, tenendo anche conto delle reali caratteristiche dei futuri utilizzatori. Questi hanno necessità e aspettative molto variegata. Non a caso si comincia ad avere come riferimento la cosiddetta utenza ampliata. Tutto ciò fa riferimento ai criteri innovatori del cosiddetto Universal Design, riconosciuto nei Paesi evoluti come base per un corretto approccio a una progettazione 'inclusiva'. Significa immaginare, sviluppare e commercializzare spazi, prodotti, servizi, sistemi e ambienti di uso generale in modo che possano risultare fruibili dal più vasto numero possibile di utenti e non siano specificamente dedicati alle persone disabili. Questo modo corretto di intendere la progettazione tiene conto delle esigenze multigenerazionali; cioè di quelle dei bambini, degli anziani e di coloro che, per qualsiasi motivo, abbiano difficoltà motorie o sensoriali.

Il risultato positivo è un notevole contenimento dei costi e un maggior gradimento anche da parte delle persone senza necessità particolari, che pertanto non si sentono oggetto di speciali attenzioni.

Tali temi dell'architettura, intesa anche come missione sociale, interessano prima o poi tutti noi. Il semplice trascorrere del tempo modifica comunque fisiologicamente le caratteristiche e le conseguenti esigenze di ciascuno.

Il diritto alla fruibilità nell'ambito degli ambienti urbani o naturali diventa sempre più pressante da parte di tutti. In modo particolare da parte di quella fascia di persone che vive, in modo diretto o indiretto, situazioni di disagio ed emarginazione causate da un'autonomia ridotta o da limitate energie disponibili.

Uno sviluppo corretto e responsabile del Paese non può quindi prescindere dal mettere in atto ogni tentativo di avvicinarsi a una mobilità sostenibile, intesa anche come fruibilità agevole dello spazio costruito, da parte di chiunque. Nell'ambito urbano e nelle strutture complesse, tra i diversi input per una buona progettazione occorre considerare quelli che rendono più facile l'orientamento. Quest'ultimo va inteso come capacità di sapere dove ci si trovi, e come possibilità di capire quali possano essere gli spostamenti da compiere, e con quali modalità, per raggiungere un determinato obiettivo prescelto (*wayfinding*). Aspetti, questi ultimi, di particolare importanza per tutti coloro che hanno una ridotta autonomia per compiere una certa azione o raggiungere una determinata meta.

Le incertezze e le perplessità dovute alla non conoscenza dei luoghi, delle attrezzature e dei sistemi di trasporto, specie se in ambiti articolati e complessi, provocano in chiunque un aumento dell'ansia e dell'affaticamento fisico e psichico. La mera operazione di messa a norma è assolutamente insufficiente, e spesso inutile, se non è raccordata e coordinata con altri fondamentali aspetti, quelli psicologici *in primis*.

L'accessibilità quindi va intesa, in modo ampio, come l'insieme delle caratteristiche spaziali, distributive e organizzativo-gestionali dell'ambiente costruito (e dei relativi sistemi per la mobilità); un insieme che consenta realmente una fruizione agevole dei luoghi e delle attrezzature della città da parte dell'intera cittadinanza. Accessibilità, quindi, come sistema complesso e articolato di elementi, collegati tra loro o interdipendenti, che consenta di avvicinarsi al concetto di autonomia o di autosufficienza. Tale sistema va perseguito come obiettivo irrinunciabile, sia di un importante settore interdisciplinare, sia di un movimento sociale [BA06].

## 1.2 Leggi e Normative

La legge quadro italiana che tratta il problema dell'accessibilità è la legge 13/89 che stabilisce i termini e le modalità in cui deve essere garantita l'accessibilità ai vari ambienti, con particolare attenzione ai luoghi pubblici.

- *accessibilità*: possibilità per persone con ridotta o impedita capacità motoria o sensoriale di raggiungere l'edificio e le sue singole unità immobiliari e ambientali, di entrarvi agevolmente e di fruire di spazi e attrezzature in condizioni di adeguata sicurezza e autonomia;
- *visitabilità*: possibilità per persone con ridotta o impedita capacità motoria o sensoriale di accedere agli spazi di relazione e ad almeno un servizio

igienico di ogni unità immobiliare. Vengono considerati spazi di relazione gli spazi di soggiorno dell'alloggio e quelli dei luoghi di lavoro, servizio ed incontro, nei quali il cittadino entra in rapporto con la funzione ivi svolta;

- *adattabilità*: possibilità di modificare nel tempo lo spazio costruito a costi limitati, allo scopo di renderlo completamente e agevolmente fruibile anche da parte di persone con ridotta o impedita capacità motoria o sensoriale.

Sempre il D.M. 236/89 stabilisce anche, per gli edifici e gli spazi privati, i parametri tecnici e dimensionali correlati al raggiungimento dei tre livelli di qualità sopra riportati: per esempio le dimensioni minime delle porte, le caratteristiche delle scale, la pendenza delle rampe pedonali, gli spazi necessari alla rotazione di una sedia a ruote, le dimensioni degli ascensori e le casistiche della loro necessità, le caratteristiche di un servizio igienico accessibile e altri ancora. I requisiti sono stabiliti in modo differente secondo la tipologia degli edifici e degli spazi. Ogni nuova costruzione deve infatti rispettare tali norme, ed i vecchi edifici devono essere opportunamente adeguati alla normativa in caso di ristrutturazione (D.M. 236/89, art 6). Per quanto riguarda gli edifici e gli spazi pubblici vi è stata l'emanazione di un ulteriore decreto attuativo[BAEP13].

Di seguito saranno elencate le normative di riferimento.

### **1.2.1 Legge 13/89**

Disposizioni per favorire il superamento e l'eliminazione delle barriere architettoniche negli edifici privati (9 gennaio 1989). Le prime disposizioni tecniche per il superamento delle barriere architettoniche sono contenute nella Circolare Ministeriale del 19 giugno 1968, in seguito riprese ed ampliate dal D.P.R. del 27 aprile 1978 che costituisce un punto fondamentale nella disciplina della materia, soprattutto per quanto riguarda le direttive di progettazione senza barriere architettoniche negli edifici pubblici a carattere collettivo e sociale.

La legge 13 del 1989 affronta le problematiche della progettazione senza barriere nell'ambito dell'edilizia residenziale, quindi negli edifici privati di nuova costruzione, negli interventi di ristrutturazione, negli spazi esterni di pertinenza e di accesso. Precedentemente le prescrizioni normative si riferivano alle opere ed agli edifici pubblici e privati aperti al pubblico, e poco significativamente agli interventi di edilizia residenziale pubblica. Con la legge 13 le disposizioni per favorire la fruizione

degli spazi vengono estese a tutti gli edifici privati, residenziali e non, in sede di nuova costruzione o di ristrutturazione degli stessi [L.13/89].

### **1.2.2 Legge 104/92**

La legge 5 febbraio 1992 n. 104/92, è il riferimento legislativo "per l'assistenza, l'integrazione sociale e i diritti delle persone handicappate". Principali destinatari sono dunque i disabili, ma non mancano riferimenti anche a chi vive con loro. Il presupposto è infatti che l'autonomia e l'integrazione sociale si raggiungono garantendo alla persona handicappata e alla famiglia adeguato sostegno. E questo supporto può essere sotto forma di servizi di aiuto personale o familiare, ma si può anche intendere come aiuto psicologico, psicopedagogico, tecnico. Dopo l'entrata in vigore, la legge è stata aggiornata in alcune sue parti, per effetto di norme introdotte in seguito [L.104/92].

### **1.2.3 D.M. 236/89**

Legge sul superamento delle barriere architettoniche, prescrizioni tecniche necessarie a garantire l'accessibilità, l'adottabilità e la visibilità degli edifici privati e di edilizia residenziale pubblica sovvenzionata e agevolata, ai fini del superamento e dell'eliminazione delle barriere architettoniche. Le norme contenute nel presente decreto si applicano:

- 1) agli edifici privati di nuova costruzione, residenziali e non, ivi compresi quelli di edilizia residenziale convenzionata;
- 2) agli edifici di edilizia residenziale pubblica sovvenzionata ed agevolata, di nuova costruzione;
- 3) alla ristrutturazione degli edifici privati di cui ai precedenti punti 1) e 2), anche se preesistenti alla entrata in vigore del presente decreto;
- 4) agli spazi esterni di pertinenza degli edifici di cui ai punti precedenti.

Per barriere architettoniche s'intendono:

- a) gli ostacoli fisici che sono fonte di disagio per la mobilità di chiunque ed in particolare di coloro che, per qualsiasi causa, hanno una capacità motoria ridotta o impedita in forma permanente o temporanea;
- b) gli ostacoli che limitano o impediscono a chiunque la comoda e sicura utilizzazione di parti, attrezzature o componenti;

c) la mancanza di accorgimenti e segnalazioni che permettono l'orientamento e la riconoscibilità dei luoghi e delle fonti di pericolo per chiunque e in particolare per i non vedenti, per gli ipovedenti e per i sordi. Il rispetto normativo della legge 13/1989 (accessibilità, visitabilità e adattabilità a seconda del tipo di intervento), deve portare alla realizzazione di elaborati grafici di progetto atti a dimostrare il superamento delle barriere architettoniche [DM236/89].

### **1.2.4 DPR 503/1996 (art 2,7,10,13)**

Di seguito elencati gli articoli che più ci interessano:

Art. 2.

#### *Contrassegni*

1. Gli edifici, i mezzi di trasporto e le strutture costruite, modificate o adeguate tenendo conto delle norme per l'eliminazione delle barriere, devono recare in posizione agevolmente visibile il simbolo di *accessibilità* secondo il modello di cui all'allegato A.
2. E' fatta salva la specifica simbologia dell'Organizzazione internazionale dell'aviazione civile ove prescritta.
3. Il sistema di chiamata di cui all'art. 1 deve essere posto in luogo accessibile e contrassegnato con il simbolo di "accessibilità condizionata" secondo il modello di cui all'allegato B.
4. Uffici, sale per riunioni, conferenze o spettacoli, posti telefonici pubblici ovvero apparecchiature quali ascensori e telefoni che assicurano servizi di comunicazione per sordi, devono recare in posizione agevolmente visibile il simbolo internazionale di accesso alla comunicazione per le persone sorde di cui all'allegato C.

Art. 7.

#### *Scale e rampe*

1. Per le scale e le rampe valgono le norme contenute ai punti 4.1.10, 4.1.11 e 8.1.10, 8.1.11. del decreto del Ministro dei lavori pubblici 14 giugno 1989, n. 236. I percorsi che superano i 6 metri di larghezza devono essere, di norma, attrezzati anche con corrimano centrale.

Art. 10.

#### *Parcheggi*

1. Per i parcheggi valgono le norme di cui ai punti 4.2.3 e 8.2.3 del decreto del Ministro dei lavori pubblici 14 giugno 1989, n. 236.

2. Per i posti riservati disposti parallelamente al senso di marcia, la lunghezza deve essere tale da consentire il passaggio di una persona su sedia a ruote tra un veicolo e l'altro. Il requisito s'intende soddisfatto se la lunghezza del posto auto non è inferiore a 6 m; in tal caso la larghezza del posto auto riservato non eccede quella di un posto auto ordinario.

3. I posti riservati possono essere delimitati da appositi dissuasori.

Art. 13.

#### *Le norme generali per gli edifici*

1. Le norme del presente regolamento sono riferite alla generalità dei tipi edilizi.

2. Negli edifici pubblici deve essere garantito un livello di accessibilità degli spazi interni tale da consentire la fruizione dell'edificio sia al pubblico che al personale in servizio, secondo le disposizioni di cui all'art. 3 del decreto del Ministro dei lavori pubblici 14 giugno 1989, n. 236.

3. Per gli spazi esterni di pertinenza degli stessi edifici il necessario requisito di accessibilità si considera soddisfatto se esiste almeno un percorso per l'accesso all'edificio fruibile anche da parte di persone con ridotta o impedita capacità motoria o sensoriale.

4. Le normative specifiche riguardanti singoli tipi edilizi possono articolare o limitare il criterio generale di accessibilità in relazione alla particolarità del tipo.

5. In sede di definizione e di applicazione di norme concernenti specifici settori, quali sicurezza, contenimento consumi energetici, tutela ambientale, ecc., devono essere studiate o adottate, nel rispetto di tali normative, soluzioni conformi alle disposizioni del presente regolamento.

6. Per gli alloggi di servizio valgono le disposizioni di cui all'art. 3.3 del decreto del Ministro dei lavori pubblici 14 giugno 1989, n. 236, relative agli alloggi di edilizia residenziale sovvenzionata.

7. Negli interventi di recupero, gli eventuali volumi aggiuntivi relativi agli impianti tecnici di sollevamento non sono computabili ai fini della volumetria utile [DPR503].

### **1.2.5 DPR 380/2001 (Art. 77-79)**

Di seguito elencati gli articoli:

Art. 77 (L)

Progettazione di nuovi edifici e ristrutturazione di interi edifici (legge 9 gennaio 1989, n. 13, art. 1)

1. I progetti riguardanti la costruzione di nuovi edifici privati, ovvero alla ristrutturazione di interi edifici, ivi compresi quelli di edilizia residenziale pubblica, sovvenzionata ed agevolata, sono redatti in osservanza delle prescrizioni tecniche previste dal comma 2.
2. Il Ministro delle infrastrutture e dei trasporti fissa con decreto, adottato ai sensi dell'articolo 52, le prescrizioni tecniche necessarie a garantire l'accessibilità, l'adattabilità e la visibilità degli edifici privati e di edilizia residenziale pubblica, sovvenzionata ed agevolata.
3. La progettazione deve comunque prevedere:
  - a) accorgimenti tecnici idonei all'installazione di meccanismi per l'accesso ai piani superiori, ivi compresi i servoscala;
  - b) idonei accessi alle parti comuni degli edifici e alle singole unità immobiliari;
  - c) almeno un accesso in piano, rampe prive di gradini o idonei mezzi di sollevamento;
  - d) l'installazione, nel caso d'immobili con più di tre livelli fuori terra, di un ascensore per ogni scala principale raggiungibile mediante rampe prive di gradini.
4. E' fatto obbligo di allegare al progetto la dichiarazione del professionista abilitato di conformità degli elaborati alle disposizioni adottate ai sensi del presente capo.
5. I progetti di cui al comma 1 che riguardano immobili vincolati ai sensi del decreto legislativo 29 ottobre 1999, n. 490, devono essere approvati dalla competente autorità di tutela, a norma degli articoli 23 e 151 del medesimo decreto legislativo.

#### Art. 78 (L)

Deliberazioni sull'eliminazione delle barriere architettoniche (legge 9 gennaio 1989, n. 13, art. 2)

1. Le deliberazioni che hanno per oggetto le innovazioni da attuare negli edifici privati dirette ad eliminare le barriere architettoniche di cui all'articolo 27, primo comma, della legge 30 marzo 1971, n. 118, ed all'articolo 1 del decreto del Presidente della Repubblica 24 luglio 1996, n. 503, nonché la realizzazione di percorsi attrezzati e la installazione di dispositivi di segnalazione atti a favorire la mobilità dei ciechi all'interno degli edifici privati, sono approvate dall'assemblea del condominio, in prima o in seconda convocazione, con le maggioranze previste dall'articolo 1136, secondo e terzo comma, del codice civile.
2. Nel caso in cui il condominio rifiuti di assumere, o non assuma entro tre mesi dalla richiesta fatta per iscritto, le deliberazioni di cui al comma 1, i portatori di handicap, ovvero chi ne esercita la tutela o la potestà di cui al titolo IX del libro primo del codice

civile, possono installare, a proprie spese, servoscala nonché strutture mobili e facilmente rimovibili e possono anche modificare l'ampiezza delle porte d'accesso, al fine di rendere più agevole l'accesso agli edifici, agli ascensori e alle rampe delle autorimesse.

3. Resta fermo quanto disposto dagli articoli 1120, secondo comma, e 1121, terzo comma, del codice civile.

Art. 79 (L)

Opere finalizzate all'eliminazione delle barriere architettoniche realizzate in deroga ai regolamenti edilizi (legge 9 gennaio 1989, n. 13, art. 3)

1. Le opere di cui all'articolo 78 possono essere realizzate in deroga alle norme sulle distanze previste dai regolamenti edilizi, anche per i cortili e le chiostrine interni ai fabbricati o comuni o di uso comune a più fabbricati.

2. E' fatto salvo l'obbligo di rispetto delle distanze di cui agli articoli 873 e 907 del codice civile nell'ipotesi in cui tra le opere da realizzare e i fabbricati alieni non sia interposto alcuno spazio o alcuna area di proprietà o di uso comune.

## 1.3 Smart City

Introduciamo il concetto di *Smart City* e tecnologie innovative applicate nelle città. L'espressione città intelligente (dall'inglese *smart city*) indica, in senso lato, un ambiente urbano in grado di agire attivamente per migliorare la qualità della vita dei propri cittadini. La città intelligente riesce a conciliare e soddisfare le esigenze dei cittadini, delle imprese e delle istituzioni, grazie anche all'impiego diffuso e innovativo delle TIC (tecnologie dell'informazione e della comunicazione), in particolare nei campi della comunicazione, della mobilità, dell'ambiente e dell'efficienza energetica.

Benché il significato di tale espressione non sia ancora stato univocamente definito nei dettagli, si riscontra un certo accordo sulle caratteristiche di attenzione ai bisogni delle persone, di gestione oculata delle risorse, di sviluppo sostenibile e di sostenibilità economica [CI13].

### 1.3.1 Descrizione

Le città intelligenti si realizzano grazie allo sforzo congiunto dell'amministrazione pubblica, dei partner privati, dei protagonisti del terzo settore,

delle università, dei centri di ricerca e naturalmente di quella cittadinanza che, se messa nelle condizioni di esprimersi, si è sempre dimostrata attiva.

Apertura dei dati, interoperabilità delle reti, perdita di dominio esclusivo su singoli servizi sono passaggi necessari che permettono alle città di mettere a sistema i propri *asset* e raggiungere una potente integrazione delle proprie attività. Ciò vale sia dal punto di vista della macchina amministrativa che da quello della cittadinanza, perché permettere al cittadino di monitorare – anche attraverso nuovi indicatori – le dimensioni che riconosce come prioritarie per il proprio benessere, significa gettare le basi per una ricostruzione del rapporto con la politica.

Le tecnologie coinvolte in una smart city offrono grandi potenzialità da sfruttare, si possono infatti:

- acquisire dati, tramite i sensori oggi disponibili a costi sempre inferiori e i dispositivi mobili come *tablet* e *smartphone*;
- comunicare, attraverso la rete fatta di cavi e onde che consentono la circolazione dei dati;
- elaborare, sia grazie a processori sempre più performanti installati su computer e dispositivi, sia - soprattutto - grazie alle straordinarie risorse oggi disponibili in *Cloud Computing* [CP13], che consentono di concentrare le risorse di calcolo al di fuori degli apparati;
- memorizzare e gestire l'enorme massa di dati disponibili e trasformarli in risorse utilizzabili. Il potenziale di questa nuova dimensione è straordinario, ma richiede la disponibilità e una gestione intelligente di tecnologie abilitanti senza le quali la strada verso la *Smart City* non è percorribile.

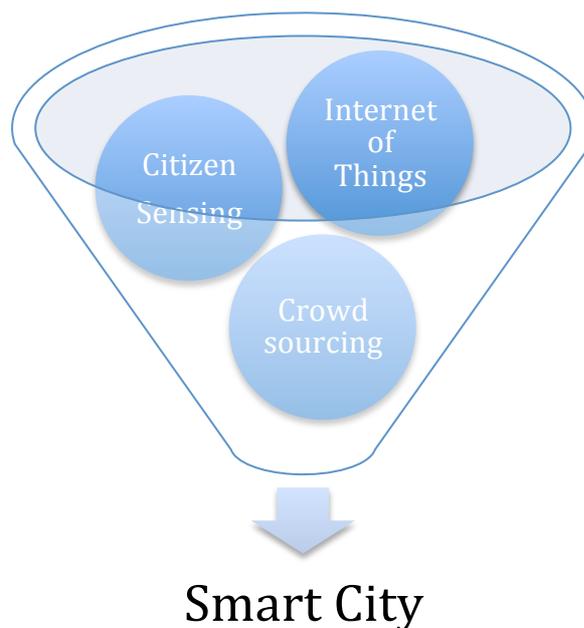
Oggi fasce sempre più ampie della popolazione urbana sono dotate di *smartphone* o *tablet*, disponibili a prezzi sempre più accessibili. Questi apparati sono in grado di comunicare non solo con voce o testo, ma dispongono di accesso a Internet ad alta velocità possono mostrare video ad alta risoluzione e sono dotati di strumenti di geoposizionamento. I costi continueranno a scendere contemporaneamente a una ulteriore crescita delle funzionalità. Questi sofisticati device sono una importante sorgente di dati legati al loro uso e al comportamento di chi li possiede.

E' di fondamentale importanza operare per una maggiore consapevolezza dei consumatori italiani che devono essere messi nelle condizioni sia di poter contribuire attivamente con i loro comportamenti allo sviluppo delle *Smart City* – per esempio

adottando le applicazioni delle città – sia di comprendere i rischi dell’adozione di strumenti che non consentono il controllo dei livelli di privacy dei propri dati, che possono essere usati per scopi illegali o indesiderati [PSC13].

Per realizzare le attività *intelligenti*, la smart city ha bisogno di avere dei data base aperti (*open data*) e sempre accessibili. Questo, anche per agevolare il ruolo attivo dei cittadini nella produzione, modifica, aggiornamento e scambio d’informazioni. Quella degli open data è una filosofia che abbraccia anche le attività della pubblica amministrazione e i software (i programmi *open source*, per esempio, sono quelli disponibili gratuitamente su web e migliorabili dagli utenti), e ha a che vedere con la condivisione del sapere e il miglioramento della qualità della vita.

La figura 1.1 descrive come l’insieme delle tecnologie che verranno descritte in dettaglio nei prossimi paragrafi contribuiscano alla formazione del concetto di smart city.



*Figura 1.1: Diagramma delle tecnologie che formano una smart city.*

### 1.3.2 Digital City

Il termine Digital City o Città digitale [DGC13] si riferisce a una comunità connessa che combina infrastrutture di comunicazione a banda larga, flessibile, una infrastruttura informatica orientata ai servizi basata su standard aperti e di servizi innovativi per soddisfare le esigenze di governi, cittadini e imprese. La dimensione

geografica (spazio) delle comunità digitali variano: possono essere estesi da un quartiere della città fino a molte metropoli.

Un elemento chiave dell'infrastruttura Città Digitale è di progettare l'infrastruttura wireless. La Città Digitale può richiedere infrastrutture a banda larga cablata, ed è molto di più che la rete normale. Una Città Digitale fornisce interoperabili, i servizi pubblici basati su Internet che consentono la connettività ubiqua per trasformare i processi chiave del governo, sia internamente tra i reparti e dipendenti ed esternamente con cittadini e imprese.

Servizi di Digital City sono accessibili attraverso i dispositivi mobili e wireless. Una smart city si basa sul fatto di avere dei servizi di Digital City efficienti.

### **1.3.3 Internet of Things**

In telecomunicazioni, Internet delle cose (o, più propriamente, Internet degli oggetti o IoT, acronimo dell'inglese Internet of Things) [IOT13] è un neologismo riferito all'estensione di Internet al mondo degli oggetti e dei luoghi concreti.

L'Internet delle cose è vista come una possibile evoluzione dell'uso della rete. Gli oggetti si rendono riconoscibili e acquisiscono intelligenza grazie al fatto di poter comunicare dati su se stessi e accedere a informazioni aggregate da parte di altri. Le sveglie suonano prima in caso di traffico, le piante comunicano all'innaffiatoio quando è il momento di essere innaffiate, le scarpe da ginnastica trasmettono tempi, velocità e distanza per gareggiare in tempo reale con persone dall'altra parte del globo, i vasetti delle medicine avvisano i familiari se si dimentica di prendere il farmaco. Tutti gli oggetti possono acquisire un ruolo attivo grazie al collegamento alla rete.

L'obiettivo dell'Internet delle cose è di far sì che il mondo elettronico tracci una mappa di quello reale, dando un'identità elettronica alle cose e ai luoghi dell'ambiente fisico. Gli oggetti e i luoghi muniti di etichette Identificazione a radio frequenza (RFID) [IRF13] o Codici QR, comunicano informazioni in rete o a dispositivi mobili come i telefoni cellulari.

I campi di applicabilità sono molteplici: dalle applicazioni industriali (processi produttivi), alla logistica e all'info-mobilità, fino all'efficienza energetica, all'assistenza remota e alla tutela ambientale.

### ***1.3.3.1 Tecnologie abilitanti***

Nella visione dell'Internet delle cose, gli oggetti creano un sistema pervasivo ed interconnesso avvalendosi di molteplici tecnologie di comunicazione (tipicamente a corto raggio). I tag RFID hanno rappresentato uno dei primi esempi in tale ambito. Tuttavia, nel tempo sono emerse nuove tecnologie in grado di rendere più efficiente la comunicazione tra gli oggetti. Tra esse spicca lo standard IEEE 802.15.4 e, soprattutto, il suo recente emendamento IEEE 802.15.4e [IEE13], in grado di incrementare notevolmente l'affidabilità dei collegamenti a radio frequenza e l'efficienza energetica, grazie all'adozione del meccanismo di accesso al mezzo Time Slotted Channel Hopping.

Queste tecnologie di più basso livello, quando integrate in architetture protocollari basate sul protocollo IP [IPP13], possono dar concretamente vita alla visione dell'Internet delle cose, essendo in grado di dialogare con i nodi della rete Internet. Sono in grado di creare operativamente una rete IP di oggetti che può dialogare con la rete Internet per creare nuovi servizi in molteplici domini applicativi.

### ***1.3.3.2 Applicazioni***

I principali domini applicativi e ambiti operativi interessati dallo sviluppo della Internet of Things sono riportati sinteticamente nel seguente elenco:

- domotica;
- robotica;
- avionica;
- industria automobilistica;
- biomedicale;
- monitoraggio in ambito industriale;
- telemetria;
- reti wireless di sensori;
- sorveglianza;
- rilevazione eventi avversi;
- smart grid e smart city;
- sistemi embedded.

### **1.3.4 Crowdsourcing**

E' un neologismo coniato da Jeff Howe (giornalista di Wired) nell'articolo *The Rise of Crowdsourcing* ed è l'unione di:

- crowd, folla/persone;
- source, fonte/sorgente;
- outsourcing, esternalizzare un'attività al di fuori della propria impresa/organizzazione/gruppo.

È una metodologia di collaborazione con la quale viene richiesto un contributo attivo alla rete (attraverso delle open call), delegando ad un insieme distribuito di persone, che si aggregano attorno ad una piattaforma web, lo sviluppo di un progetto o di una parte di un'attività di un'azienda. Fondamentali in questo caso gli UGC (User Generated Content), i contenuti generati dagli utenti: il contributo attivo è proprio quello delle persone che decidono di partecipare su base volontaria.

Questo processo viene favorito dagli strumenti che mette a disposizione il web. Ad esempio, al pubblico può essere richiesto di sviluppare nuove tecnologie, portare avanti un'attività di progettazione, definire o sviluppare un algoritmo, gestire e organizzare grandi quantità di dati (l'enciclopedia *Wikipedia* è considerata un esempio di crowdsourcing volontario).

### **1.3.5 Participatory Sensing e Citizen Sensing**

Si parla di un nuovo paradigma che consente la raccolta distribuita d'informazioni e dati da parte d'individui, agendo da soli o in gruppo. La raccolta di tali informazioni avviene utilizzando tecnologie mobili e servizi web, quindi abbiamo la possibilità di raccogliere dati di tutti i tipi dalla salute alla cultura.

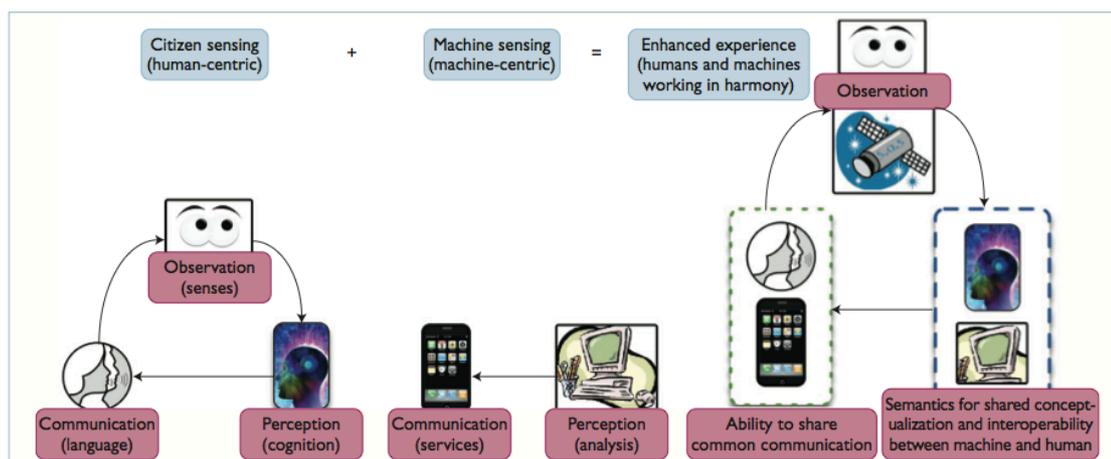


Figura 1.2: Schema dell'estensione del concetto di Citizen Sensing a cui è applicata la tecnologia.

Nella figura 1.2 vediamo come si passa da un primo stato in cui si descrive la concezione umana dell'argomento. Si osservano come le cose sono elaborate dalla nostra mente e poi sono comunicate ad altre persone.

Nella seconda parte è descritto come invece interagisce una macchina e come percepisce i dati tramite delle analisi e questi dati vengono comunicati tramite l'utilizzo di altre macchine connesse.

La parola citizen sensing, fonde questi due concetti in modo da avere i seguenti rapporti:

- comunicazione: da la possibilità all'utente di comunicare tramite smartphone o altro;
- osservazione: vengono utilizzate immagini satellitari e tecnologie di visione artificiale;
- capire: le informazioni ottenute utilizzando tecnologie semantiche per far capire uomo e macchina.

Una tipica infrastruttura di *Participatory Sensing* (PS) [PS12] coinvolge le seguenti parti:

- i nodi mobili sono l'unione degli utenti con un sensore installato sul telefono, essi forniscono le informazioni necessarie ad un progetto Participatory Sensing;
- dare la possibilità di interrogare le informazioni raccolte;
- raccogliere informazioni utilizzando anche i dati presi dal gestore di rete;
- i Service Provider devono fungere da intermediari tra le interrogazioni e i dati mobili.

Mentre nascono nuovi servizi e progetti di ricerca, l'impatto di questi progetti nel mondo reale è limitato dalla partecipazione globale dell'utente. Se gli utenti non hanno alcun incentivo, o sentono che la loro privacy è messa in pericolo, è probabile che non parteciperanno al progetto causandone il fallimento totale o parziale.

## **1.4 Sistemi informativi territoriali**

Introduciamo il concetto di Sistemi informativi territoriali (GIS) [GIS00].

Il termine nasce per indicare quei programmi informatici in grado di geo-referenziare le informazioni (cioè associare stabilmente i fenomeni alle coordinate geografiche) e quindi fornire risposte a problemi territoriali.

### **1.4.1 Storia e utilizzo**

Il GIS commissionato nel 1968 dal governo canadese per la descrizione dell'uso del suolo e quello del Bureau of the Census degli Stati Uniti per la riorganizzazione della base territoriale del censimento 1970 vanno ricordati tra le prime applicazioni di grande rilievo. Da quel momento, e soprattutto negli anni Ottanta, l'offerta di GIS sul mercato dell'informatica è aumentata rapidamente tanto che oggi si possono contare alcune centinaia di prodotti software di questo tipo con caratteristiche e funzioni diverse; e il crescente interesse per i GIS è evidenziato dal moltiplicarsi di riviste specializzate e di convegni che si occupano dell'evoluzione dei sistemi informativi geografici e segnalano l'installazione di applicativi in tutto il mondo.

Molte discipline concorrono allo sviluppo dei sistemi informativi geografici e questi, a loro volta, sono utilizzati in molti settori della ricerca applicata, nelle amministrazioni civili e militari e nelle imprese.

### **1.4.2 Definizione**

La vastità e l'eterogeneità degli interessi scientifici e operativi che convergono su di essi rendono particolarmente difficile una loro definizione. Si può dire, tuttavia, che i GIS sono sistemi hardware-software finalizzati all'acquisizione, gestione, elaborazione, analisi, modellizzazione e rappresentazione di *data-base* (banche dati) i cui elementi possiedono una posizione geografica. Oltre al posizionamento geografico

degli oggetti, il *data-base* contiene anche attributi e informazioni che, in relazione al tipo di utilizzazione, servono a distinguere tra loro gli oggetti e a metterne in evidenza le relazioni al fine di risolvere problemi di gestione e pianificazione territoriale.

Il GIS è dunque, in sostanza, un software informatico in grado di trattare e gestire dati associati a una base cartografica. Diversamente dai programmi di disegno assistito da calcolatore o dai modelli di cartografia numerica, i GIS dispongono di specifici strumenti di analisi quali la ricerca di elementi distribuiti nel territorio e la loro selezione per attributi (per es.: la selezione di tutte le strade di un ambito territoriale con flussi di traffico determinato o di tutte le sezioni di censimento di un'area urbana con densità di popolazione superiore a un numero prefissato di abitanti per unità di superficie ecc.), la rappresentazione di aree d'influenza, la scomposizione delle immagini in livelli (*layers o themes*, come le curve di livello, il reticolo idrografico ecc.) e la loro sovrapposizione (*map overlay*), la generazione di modelli digitali del terreno, la scelta della modalità di rappresentazione mediante il cambio di scala o di proiezione.

Per queste loro caratteristiche e per le estensioni e personalizzazioni che si possono ottenere, i GIS sono utilizzati in numerosi campi applicativi tra i quali possono essere citati quelli che si riferiscono alla progettazione urbanistica e regionale, allo studio, catalogazione e gestione dei beni culturali, all'organizzazione delle risorse agricole, alla gestione dei beni demaniali e del catasto, alla protezione civile, al monitoraggio ambientale, alle analisi socioeconomiche, alla gestione e progettazione di reti di distribuzione e trasporto (acquedotti, oleodotti, gasdotti, elettrodotti) e di telecomunicazioni, all'organizzazione sul territorio di operazioni militari e di polizia.

### 1.4.3 Struttura

Punto di partenza per la creazione di un GIS è la realizzazione di banche dati geografiche (database) che opportunamente strutturate consentono di organizzare tutti i dati di interesse. Le principali caratteristiche di un database sono:

- forma;
- contenuto informativo;
- sistema di riferimento geodetico;
- sistema di coordinate;
- strutturazione dei particolari geografici;

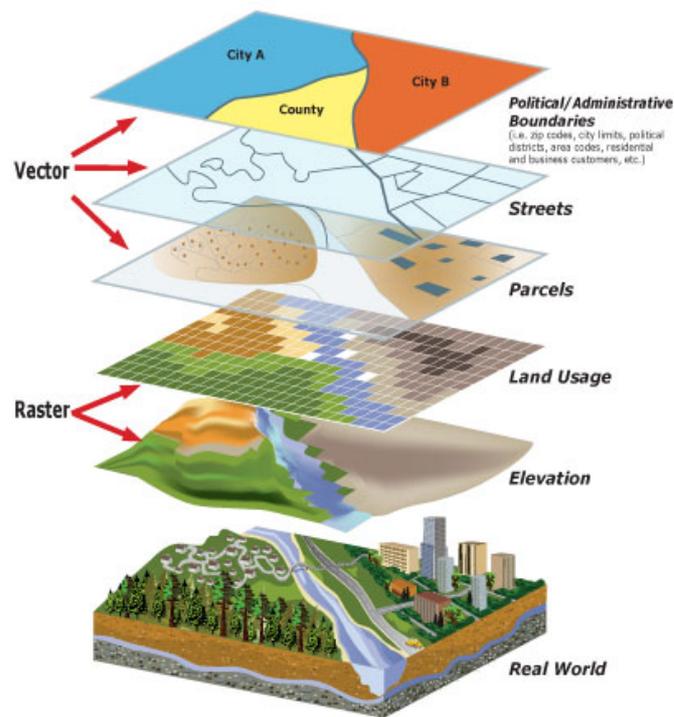
- qualità;
- sistema di codifica;
- formato di scambio;
- taglio cartografico e reticolato chilometrico;
- simbologia convenzionale.

Le forme di rappresentazione utilizzate sono: vettoriale, rasterizzate e matrici. Nella forma vettoriale il dato geografico viene rappresentato attraverso tre componenti:

- geometrica: che descrive la forma degli oggetti utilizzando le primitive punto, linea ed area;
- attribuzione: che descrive le caratteristiche degli oggetti utilizzando un database relazionale associato;
- relazione: che descrive i mutui legami fra gli oggetti utilizzando le relazioni topologiche (adiacenza, appartenenza e connessione).

Il modello topologico è la struttura che si assegna ai dati per definire esplicitamente le relazioni, i rapporti di connessione e di continuità tra gli elementi spaziali e per collegare tali elementi alle relative descrizioni (attributi); essa consente di memorizzare più efficacemente le informazioni, eseguire analisi sui dati cartografici ed elaborare più velocemente le complesse banche dati.

La topologia è utilizzata con notevole beneficio nei database in quanto permette di effettuare analisi spaziali senza prendere in considerazione le coordinate degli oggetti (geometria).



*Figura 1.3: Struttura sistema GIS*

Nella figura 1.3 possiamo notare come un'architettura GIS sia fatta a strati che corrispondono, partendo dall'alto:

- trasporti e pubblica amministrazione;
- utilizzo del suolo e strade;
- censimento e lottizzazione;
- dove sono collocate le strutture;
- rilievo;
- immagini satellitari.

## 1.5 Data Integration

La grande disponibilità di dati dovuta all'avvento di Internet ha imposto alla comunità di ricerca nuove sfide. Nell'ottica di consentire a ciascuno di "memorizzare, organizzare, accedere ed analizzare l'informazione disponibile in linea", uno dei problemi centrali è quello dell'integrazione dei dati [TIDE13], ovvero la possibilità di consentire l'accesso unificato a sorgenti di dati eterogenee e indipendenti, come se fossero una sorgente unica. Senza l'applicazione di opportuni metodi, gli utenti hanno a

disposizione grandi quantità di dati, ma trovano inevitabili difficoltà nel sintetizzare l'informazione utile ai propri scopi.

Il problema dell'integrazione ha ricevuto moltissima attenzione negli ultimi anni, ma non esistono, allo stato attuale, soluzioni complete. La causa principale di questo fatto è probabilmente la complessità intrinseca del problema. E' difficile fornire una visione unitaria di un insieme di sorgenti di dati eterogenee e sviluppate in modo autonomo. Questa complessità ha fatto sì che la ricerca si concentrasse di volta in volta su aspetti specifici del problema. Viceversa, il processo d'integrazione è estremamente articolato.

Le tecniche d'integrazione sono state utilizzate per poter raccogliere in un'unica base di dati tutte le informazioni presenti nei diversi database dei servizi che hanno già raccolto dei dati. Parte fondamentale del progetto è raggruppare tutte le informazioni che sono state finora raccolte da altri servizi già attivi e utilizzarle come base di partenza.

### **1.5.1 Descrizione**

Possiamo schematizzare il processo di integrazione come il risultato di varie attività principali. Il punto di partenza è tipicamente la ricerca di corrispondenze tra schemi, in cui vengono scoperte analogie tra le descrizioni e il contenuto delle varie sorgenti di dati, definendo modalità per passare da uno schema all'altro.

In seguito, si pone il problema della traduzione delle istanze, in cui, sulla base delle corrispondenze tra schemi scoperte al passo precedente, vengono affrontati tutti i problemi che riguardano la traduzione dei dati. In questa fase si pongono, tipicamente, problemi di *data cleaning*, incluso il rilevamento e la soluzione di possibili inconsistenze tra formati e vincoli delle sorgenti.

Infine, è necessario risolvere il problema dell'esecuzione delle query [QIS13] (*query execution*), che consente effettivamente di stabilire la cooperazione applicativa tra le sorgenti di dati. Risulta quindi importante ricercare nuove metodologie per l'integrazione di sorgenti eterogenee di dati, per il progetto di basi di dati destinate all'analisi in linea di dati di sintesi e per la scoperta di nuovi collegamenti e proprietà non facilmente intuibili all'interno di una sorgente o di sorgenti diversi.

Numerose architetture applicative sono state adottate per la soluzione di problemi d'integrazione. Ad un estremo ci sono i sistemi basati su mediatori che si occupano di rendere visibili i dati di sorgenti diverse come se fossero costruiti secondo

modelli comuni. All'altro estremo ci sono le più moderne applicazioni d'integrazione basate su sistemi peer-to-peer, in cui tipicamente non ci sono ruoli di coordinamento, e l'integrazione è basata sulla definizione di trasformazioni tra gli schemi e le istanze dei peer.

La problematica dell'integrazione di dati riveste grande importanza applicativa e industriale, specialmente in relazione alla opportunità di estrarre informazione di rilevanza direzionale e strategica dai dati del livello operativo delle imprese.

### **1.5.2 Analisi e normalizzazione**

Analizzando più in dettaglio la fase d'integrazione si evidenziano le seguenti operazioni.

- **analisi e normalizzazione:** consiste nell'analizzare i diversi schemi locali producendo come risultato un insieme di schemi concettuali localmente completi e consistenti;
- **definizione dello schema riconciliato:** è la fase più importante in cui i diversi schemi locali vengono fusi in un unico schema globalmente consistente;
- **definizione del mapping:** utilizzando lo schema logico riconciliato si definisce la relazione (mapping) tra concetti degli schemi sorgenti e dello schema riconciliato.

I procedimenti sono riassunti nella figura 1.3.

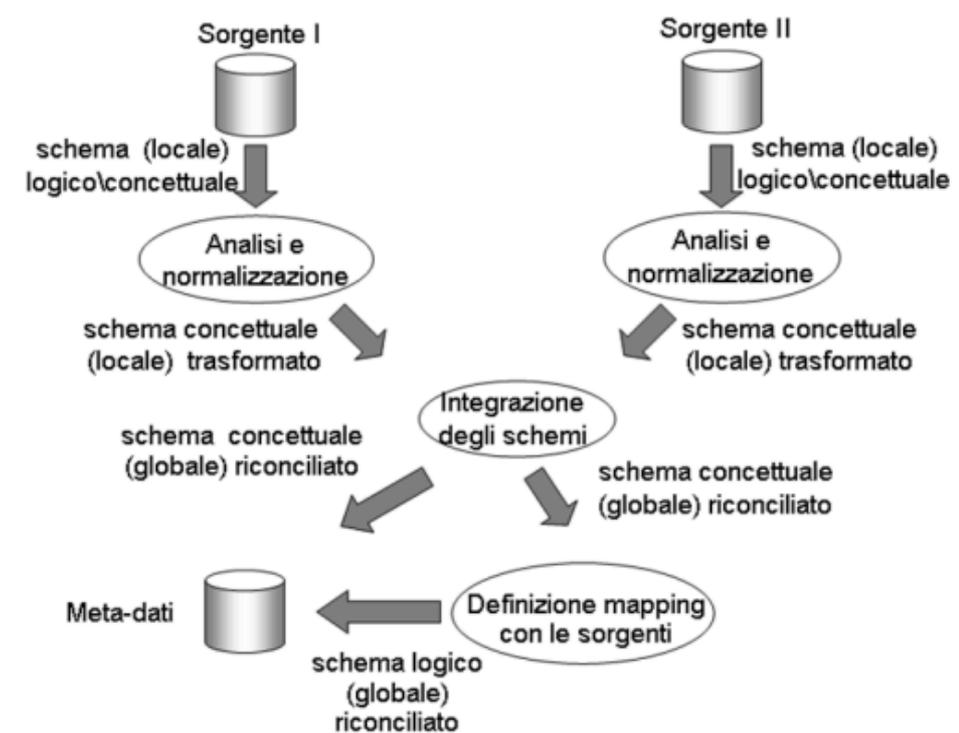


Figura 1.3 Diagramma funzionamento integrazione.

### 1.5.3 Concetti comuni

È necessario definire il tipo di relazione semantica esistente tra concetti comuni modellati diversamente in schemi distinti. Quattro sono le possibili relazioni esistenti tra due distinte rappresentazioni R1 e R2 di uno stesso concetto:

- identità: vengono utilizzati gli stessi costrutti, il concetto è modellato dallo stesso punto di vista e non vengono commessi errori di specifica; in altre parole quando R1 e R2 coincidono;
- equivalenza: R1 e R2 non sono esattamente le stesse poiché sono stati utilizzati costrutti diversi (ma equivalenti) e non sussistono errori di specifica o diversità di percezione;
- comparabilità: R1 e R2 non sono né identici né equivalenti, ma i costrutti utilizzati e i punti di vista dei progettisti non sono in contrasto tra loro;
- incompatibilità: R1 e R2 sono in conflitto a causa dell'incoerenza nelle specifiche. In altre parole la realtà modellata da R1 nega la realtà modellata da R2.

A esclusione della situazione di identità, i casi precedenti determinano dei conflitti la cui soluzione rappresenta la componente principale nella fase di integrazione.

#### **1.5.4 Le fasi dell'integrazione**

Risolvere i problemi fin qui elencati ed evidenziare le proprietà che emergono a seguito dell'integrazione degli schemi locali richiede un complesso insieme di operazioni, per la cui corretta gestione è necessario adottare una metodologia. Le molte metodologie proposte in letteratura concordano sulla sequenza di passi che devono essere svolti e che possono essere così sintetizzati:

- pre-integrazione: comprende la fase di analisi e di scelta della strategia di integrazione;
- comparazione degli schemi: durante questa fase gli schemi locali vengono confrontati per evidenziare i conflitti e i concetti correlati;
- allineamento degli schemi: durante questa fase il progettista deve risolvere i conflitti precedentemente individuati;
- fusione e ristrutturazione degli schemi: gli schemi resi coerenti possono essere fusi a creare un unico schema globale.

#### **1.5.5 Pulizia dei dati**

Con il termine pulizia dei dati (*Data Cleaning*) s'intende l'insieme delle operazioni atte a garantire la consistenza e la correttezza dei dati.

Le principali cause d'inconsistenza dei dati sono:

- errori di battitura;
- differenze del formato dei dati;
- inconsistenza tra valori e descrizione dei campi;
- informazioni mancanti o duplicate.

### **1.6 Analisi e valutazione dei servizi esistenti**

In seguito sono analizzati i diversi servizi web e non, che offrono soluzioni al problema delle barriere architettoniche. Vedremo come ogni servizio è differente rispetto agli altri per le tecnologie e metodi implementativi diversi che sono stati

utilizzati per la creazione del servizio. Tutti i servizi presi in considerazione saranno confrontati per vedere il possibile utilizzo e recupero dei dati.

### **1.6.1 Descrizione dei servizi**

Vediamo cosa i servizi mettono a disposizione elencandone le principali caratteristiche e confrontandoli.

#### ***1.6.1.1 EASY WAY – VODAFONE***

Il servizio Easy Way [EAW12] messo a disposizione da Vodafone permette l'inserimento delle valutazioni di accessibilità alla comunità degli utenti. Per ogni punto d'interesse segnalato l'utente può esprimere sia una valutazione di accessibilità generale, sia una valutazione specifica sugli spazi della struttura (parcheggio, entrata, Spazi interni, ecc.). Riguardo a una o più tipologie di disabilità motoria scelte in base alle tue competenze (carrozzina elettronica, carrozzina manuale senza accompagnatore, carrozzina manuale con accompagnatore, difficoltà di deambulazione).

Non sono ancora attive la rilevazione e la segnalazione delle particolari barriere esistenti per le persone con disabilità di tipo sensoriale. Oggi chi vuole consultare le informazioni presenti nella banca dati, può cercare un punto d'interesse nella città e leggere la valutazione di accessibilità che gli è stata attribuita da un altro utilizzatore.

Per quanto riguarda il lato mobile andiamo a elencare le principali caratteristiche:

- accesso a punti d'interesse suddivisi per categorie: ristoranti, musei, etc;
- visualizzazione su mappa dei punti d'interesse;
- informazioni anagrafiche delle segnalazioni;
- inserimento di un giudizio generale (accessibile, parzialmente accessibile e non accessibile) su un punto d'interesse;
- inserimento di un giudizio di dettaglio per tipologia di difficoltà motoria (es. carrozzina elettronica, carrozzina con accompagnatore, etc.) e spazio fisico del punto d'interesse (es. parcheggio, spazio interno, esterno) e altro.

#### **CONSIDERAZIONI:**

Il servizio utilizza Bing Mappe [BMM13] e non permette nessuna possibilità di recuperare i dati.

### 1.6.1.2 LIBERI DI MUOVERSI

Liberi di Muoversi [LDM11] Un progetto nato online per permettere alle persone con disabilità di muoversi più agevolmente in un Paese dove le barriere architettoniche sono ancora un problema grave. Possibilità di visualizzare le segnalazioni tramite mappa, oppure eseguire la ricerca di un luogo in particolare tramite un filtro per nazione e per provincia. I risultati non sono molto chiari e non hanno un buon criterio di ordinamento.

Per quanto riguarda la procedura di registrazione di un luogo, troviamo:

- localizzazione manuale su mappe oppure tramite *smartphone*, dove sono indicate le coordinate GPS;
- scelta del livello di accessibilità (accessibile, parzialmente, non accessibile);
- inserimento del nome del luogo;
- inserimento di un commento testuale;
- possibilità di inserire una foto della barriera architettonica.

La segnalazione dei luoghi e delle barriere può avvenire sia attraverso il sito web che da dispositivo mobile. Tutte le segnalazioni, indipendentemente dallo strumento utilizzato per eseguirle, sono accessibili attraverso tutti i canali.

Per quanto riguarda l'applicazione mobile notiamo delle nuove funzionalità: accesso a frasi comuni tipo *si/no*; memorizzazione proprie frasi frequenti con interfaccia d'accesso *veloce*; memorizzazione di audio utilizzabili in modo diretto (senza *text to speech*); segnalazioni di emergenza con testo SMS spedito a destinatari predefiniti; chiamate di emergenza con chiamate e esecuzione audio a destinatari predefiniti.

L'Applicazione permette di caricare in automatico sul sito la barriera architettonica identificata in questo modo:

- si scatta una foto con lo *smartphone*;
- si scrive il contenuto per una nota da allegare;
- il sistema rileva automaticamente le coordinate geografiche e pubblica immediatamente il post sul portale del servizio.

CONSIDERAZIONI:

Riassumendo quanto analizzato, troviamo che il servizio dispone di un buon Sito web funzionale. Offre la possibilità di utilizzare le *API di Google Earth* [GEA13] per accedere e condividere le informazioni raccolte. Purtroppo i dati raccolti non sono ben strutturati. Infatti, non si riesce a estrapolare molte informazioni dalla sorgente dati del servizio.

Il metodo di geo localizzazione non è molto preciso, poiché le coordinate sono arrotondate fino a due cifre soltanto dopo la virgola. L'applicazione mobile non è molto utilizzabile e si limita a raccogliere le coordinate GPS.

### ***1.6.1.3 MAP ABILITY***

Mapability [MAPB13] utilizza un sistema mediante il quale sono gli utenti a segnalare le barriere architettoniche e le agevolazioni all'interno delle città. Queste segnalazioni includono la presenza di ostacoli, la larghezza dei marciapiedi, l'accessibilità degli edifici pubblici e molte altre informazioni utili a diversi tipi di accessibilità. Troviamo un elenco delle mappe disponibili secondo le città prese in soggetto. Sono stati creati dei gruppi di collaborazione per l'editing delle mappe relative a ogni città presa in considerazione.

Un utente può partecipare ad un gruppo di lavoro per editare la mappa, basta la registrazione al sito per poter entrare a far parte di un gruppo. Non le città mappate sono: Torino; Cremona; Pavia; Mantova; Verona; Campobasso; Treviso.

(che però non sono mappate al 100%).

Dopo l'apertura della mappa sono visualizzati i percorsi a piedi che sono stati segnalati dagli utenti con le seguenti descrizioni:

- lunghezza del percorso;
- presenza di veicoli che intralciano il percorso;
- tipo di pavimentazione;
- pendenza;
- larghezza;
- illuminazione;
- presenza di ostacoli;
- presenza di agevolazioni.

**CONSIDERAZIONI:**

Sono inoltre visualizzati luoghi d'interesse, di cui sono descritti tipologia, presenza di ostacoli e larghezza ingresso. Il sito web non è molto accessibile da un utente non esperto. La scelta di utilizzare i percorsi rende difficile la comprensione della mappa (creata con *Gmaps* [GMAP13]). Inoltre non c'è la possibilità di raccogliere dati e informazioni fruibili.

Non è stata sviluppata un'applicazione per Smartphone, la raccolta delle informazioni avviene unicamente on-line.

**1.6.1.4 INGRESSO LIBERO**

Il gruppo di lavoro che realizza il progetto [INL13] è formato da persone con e senza disabilità, che visitano e recensiscono vari luoghi d'interesse e inseriscono tutti i dati raccolti del database. Ogni informazione pubblicata sul sito è frutto di un sopralluogo diretto e tiene conto sia delle norme legate alle barriere architettoniche (ad esempio, la larghezza della porta o la pendenza dello scivolo) che di elementi di tipo qualitativo, quali, ad esempio, la disponibilità degli esercenti e del personale. Il database è in continuo ampliamento e le visite replicate periodicamente sui locali già recensiti per tenere costantemente aggiornata la base dati.

Per il momento il servizio è solo per la città di Bologna. I locali sono divisi per categorie e per tipologie. Le schede dei locali e delle strutture rilevati contengono le seguenti informazioni:

- i dati di servizio (indirizzo, orari, recapiti, mezzi di collegamento);
- un riassunto degli aspetti salienti legati all'accessibilità;
- una descrizione di ciò per cui il locale si caratterizza e delle proposte e risorse che offre;
- una descrizione dettagliata delle strutture (dimensione degli spazi, dei passaggi, delle porte, caratteristiche della luminosità, tipo di servizio);
- per ogni locale è possibile lasciare e aggiungere commenti personali. Necessaria registrazione al servizio.

**CONSIDERAZIONI:**

I luoghi che sono stati segnalati devono essere ricercati manualmente tramite ricerca testuale. I dati non sono georeferenziati. Non è presente un'applicazione per

smartphone e non si è riusciti a prendere contatto con i gestori del servizio per recuperare le segnalazioni.

#### **1.6.1.5 WHEELMAP**

Una mappa online dove chiunque può mettere in rete o segnalare, anche tramite *smartphone*, i luoghi che presentano ostacoli per una persona in carrozzina, ma anche per anziani che usano un deambulatore e genitori con passeggino. *WheelMap* [WHM13] serve a catalogare tutti i punti di difficile accesso per una persona in carrozzina.

La stazione ferroviaria, il cinema, il bar, il negozio può essere classificato con bandierine di diverso colore.

- il verde segnala che non ci sono problemi: tutte le località sono raggiungibili in sedia a rotelle, il gradino all'ingresso è più basso di sette centimetri, è disponibile il bagno per persone con disabilità motoria;
- la bandierina gialla indica i luoghi parzialmente accessibili: sono raggiungibili solo quelli più importanti e l'ingresso ha un gradino inferiore a sette centimetri;
- il rosso segnala le località off-limits: quelle principali non sono raggiungibili in carrozzina e l'ingresso ha un gradino più alto di sette centimetri.

Tramite registrazione al sito possiamo modificare la mappa. Caratteristiche che possono essere modificate dopo l'inserimento di un luogo:

- nome;
- accessibilità per sedie a rotelle;
- commento sull'accessibilità;
- tipo di luogo;
- possibilità di aggiungere descrizione anagrafica dettagliata.

#### **CONSIDERAZIONI:**

*Wheelmap* è basata su *OpenStreetMap* [OSM13], chiunque può contribuire a migliorare i dati accumulati. L'applicazione trae le sue informazioni direttamente dalla rete. Tutti i dati sono sempre aggiornati. Per cercare luoghi o per modificare lo stato di accessibilità di un percorso esistente, non è richiesta la registrazione. Se si desidera

inserire lo stato di accessibilità di una nuova posizione è necessario un account sia sul sito del servizio e sia su *OpenStreetMap*. L'Applicazione mobile è presente sia per Android che per *IOS* [IOS13].

Tantissime segnalazioni in tutta Europa, i dati però non sono molto esaurienti. Visualizzazioni *Google Street View* [GSV13]. Offre la possibilità di recuperare i dati (tramite API messe a disposizione da *OpenStreetMap*). La mappa è ben utilizzabile anche applicando dei filtri a seconda della categoria del punto di interesse.

La visualizzazione delle segnalazioni non è molto chiara, ma in questo caso è precisa. Il servizio è stato costruito in Germania ed è ben tradotto nelle altre lingue.

### **1.6.1.6 COMUNI PER TUTTI**

Comuni per tutti [CPT13] un progetto per lo sviluppo di comuni accessibili in Alto Adige interessati ad assumere una posizione di rilievo a livello provinciale nell'ambito dell'abbattimento delle barriere architettoniche. La Cooperativa Sociale indipendente svolge, su incarico dei comuni altoatesini, un'analisi professionale dell'accessibilità dei paesi e delle città della Provincia di Bolzano, ai fini di migliorarne la qualità di vita e di garantire alle persone con esigenze particolari una vita più autonoma possibile.

Obiettivo del progetto è l'individuazione sistematica di tutte le barriere architettoniche sia per disabili fisici che sensoriali e la conseguente elaborazione di specifiche proposte di miglioramento che consentano ai Comuni di effettuare graduali adattamenti e ammodernamenti per abbattere le barriere architettoniche. A tal fine è rappresentata graficamente la posizione esatta di ciascun ostacolo, documentato anche fotograficamente. L'analisi non riguarda solo gli spazi pubblici, bensì anche i servizi locali, tra cui ad esempio il trasporto pubblico.

Sulla base delle proposte di miglioramento elaborate, potranno essere migliorate l'offerta e le infrastrutture stesse. Ogni barriera rilevato sul territorio comunale è corredata dai seguenti dati:

- numero ID dell'ostacolo rilevato;
- indirizzo del punto rilevato;
- coordinate geo referenziate del punto;
- descrizione dell'ostacolo;
- materiale fotografico;
- categorie ostacoli;
- ricerca per categoria di impedimento;

- ricerca in base al tipo di disabilità.

## CONSIDERAZIONI:

Il servizio è limitato alla città di Bolzano e pochi comuni limitrofi. La navigazione tramite *Google Maps* è molto veloce. Fornisce una buona visualizzazione d'immagini degli ostacoli e/o impedimenti ma l'individuazione delle segnalazioni è confusionaria. Non è presente l'applicazione mobile. Sono presenti troppe tipologie d'impedimento, che rendono la compilazione molto confusionaria. Non c'è la possibilità diretta di accedere ai dati. Non tutte le segnalazioni offrono dati esaurienti.

### **1.6.1.7 ACCESS TOGETHER**

Il servizio è nato nel 2011 come strumento per raccogliere, visualizzare ed editare i dati sulla accessibilità. Access Together [ACT13] utilizza il Crowdsourcing per raccogliere le informazioni, e ciò consente a chiunque di usare il proprio *smartphone* per indicare e aggiornare la situazione di un preciso posto. Di recente l'applicazione si è connessa a *Foursquare*, e questo le ha consentito da una parte di avvantaggiarsi di un bel numero di rilevatori potenziali, dall'altra di diffondere il progetto tra le persone e sensibilizzare maggiormente sul tema.

Di seguito è descritto il funzionamento del servizio: per prima cosa, occorre dare all'applicazione il permesso di collegarsi al nostro account *Foursquare* [FSQ13]; poi, dopo aver fatto *check-in* in un luogo d'interesse, compare un questionario da compilare tramite il dispositivo mobile, segnalando per prima cosa in che tipo di luogo ci troviamo. Le domande riguardano la disponibilità di servizi accessibili e altre caratteristiche; il questionario è organizzato per tipo di disabilità:

- persone su sedia rotelle;
- cechi;
- sordi o con problemi di udito;
- suscettibili ai sensi;
- anziani.

## CONSIDERAZIONI:

Tramite un questionario molto dettagliato l'utente può inserire i dati. Come per tutte le *Connected App*, anche in questo caso il tutto avviene dentro *Foursquare*. Le risposte sono inserite nel database di Access Together e messe a disposizione degli utenti disabili attraverso il sito web, la versione mobile del sito e la stessa applicazione. Inoltre, le API di Access Together possono essere integrate in altre applicazioni.

I luoghi sono identificati in modo univoco per la loro ID luogo *Foursquare*, è possibile utilizzare questo campo univoco per associare l'accesso alla base di dati, dove sono descritte tutte le caratteristiche inserite dagli utenti. Lo schema del *database* contiene le informazioni sul luogo come: nome, indirizzo, città, provincia, paese, latitudine, longitudine, coordinate geografiche, posizione e categoria.

La tabella comprende anche tutti i servizi di accessibilità che sono stati raccolti dalla comunità. Ogni riga della tabella rappresenta un unico luogo. Ogni riga comprende colonne per tutti i servizi ma in una posizione avrà solo informazioni pertinenti a se stesso, ad esempio, ristoranti avranno diversi servizi a disposizione rispetto a un museo. Le colonne della tabella contengono le domande che sono poste nel questionario.

Sul sito web, ogni gruppo di persone che aderisce al progetto avrà una sua pagina con una mappa che mostra i luoghi censiti e classificati con icone colorate che indicano il livello di accessibilità (rosso, giallo, verde). Attualmente ci sono tre località che hanno aderito a Access Together: Chelsea (NYC), Boston (Massachusetts) e Winnipeg (MB, Canada).

Servizio ottimo con utilizzo di tecnologie all'avanguardia come ad esempio *Foursquare* per acquisire informazioni. Il questionario composto di molte domande divise per tipo di disabilità, che però risulta molto complesso e laborioso da compilare. Esiste la possibilità di recupero dei dati tramite il servizio *Google Fusion Table* [FT13].

### **1.6.1.8 DECORO URABANO – WE DU!**

*Social network* per il dialogo con le pubbliche amministrazioni. Creato da *Maiora labs*, con l'idea di creare un punto di ritrovo per chiunque vuole contribuire personalmente alla cura della propria città.

Decoro Urbano [DUR13] ha dato vita una comunità in tutta Italia. Attraverso l'applicazione mobile, chiunque può creare il proprio profilo dal sito. In alternativa è possibile compiere l'accesso con il proprio account *Facebook* [FBK13]. Le segnalazioni

possono essere inviate dal sito attraverso la procedura guidata o via *smartphone*, dove è sufficiente lanciare l'applicazione e scattare una foto per far sì che il dispositivo vi associ automaticamente le coordinate GPS e visualizzi la segnalazione sulla mappa. Da quel momento gli utenti possono commentarla, condividerla online o sottoscriverla accrescendone la visibilità.

Strumento per le Pubbliche amministrazioni: un comune con certe caratteristiche ha accesso ad un pannello di controllo per monitorare costantemente il territorio, ottimizzando la gestione degli interventi. Decoro Urbano è una grande banca dati che mostra in modo completamente trasparente segnalazioni relative all'intera superficie nazionale. Anche un comune non attivo può dunque visualizzare la mappa in tempo reale, così come i cittadini possono portare all'attenzione dell'amministrazione le proprie istanze servendosi degli strumenti offerti gratuitamente da Decoro Urbano. Le pagine di ciascun comune possono essere raggiunte dalla ricerca rapida o digitando l'indirizzo così composto: *nomecomune.decorourbano*. La procedura di caricamento segnalazione:

1. Scelta Tipologia:

- rifiuti;
- vandalismo;
- dissesto Stradale;
- zone Verdi;
- segnaletica;
- affissioni Abusive.

2. Descrizione:

- testuale immessa dall'utente;
- indirizzo;
- immagine.

CONSIDERAZIONI:

Il progetto è interamente *OpenSource*. Autenticazione tramite *Facebook* o iscrizione su sito web tramite *form*.

È presente un'applicazione mobile ben fatta sia per *Android* che *IOS*. Integrazione nella pagina principale delle news degli utenti in tempo reale dai *social network*. Esiste la possibilità di creare widget personalizzati da integrare in blog e

pagine web. Inoltre è stato sviluppato un sistema di punteggio a classifica in base alle segnalazioni effettuate, molto simile a Foursquare. Non è stato possibile raccogliere direttamente le segnalazioni.

### 1.6.2 Servizi a confronto

In seguito sono messi a confronto i servizi analizzati secondo le loro caratteristiche.

	<b>Mappa</b>	<b>App. mobile</b>	<b>Recupero dati</b>	<b>Esaustività dati</b>	<b>Area geografica</b>	<b>Ricerca segnalazioni</b>
<i>Easy Way</i>	Microsoft	Android, Iphone	No	Si	Italia	No
<i>Liberi di Muoversi</i>	Google	Android, Iphone	Parziale (xml)	Scarsa	Italia	Si
<i>MapAbility</i>	Google	No	No	Scarsa	Alcune città Italiane	No
<i>Ingresso Libero</i>	No	No	Si (csv)	Buona	Bologna	Si
<i>WheelMap</i>	Open Street Map	Android, Iphone	No	Buona	Mondo	Si
<i>Comuni per tutti</i>	Google	No	No	Buona	Bolzano e limitrofi	
<i>Access Togheter</i>	Si	Fourquare	Fusion Table	Ottima	USA	Si
<i>Decoro Urbano</i>	Si	Si	No	Buona	Italia	Si

*Figura 1.5 Tabella comparazione servizi.*

Analizzando la tabella possiamo notare come i molti dei servizi presi in considerazione siano incompleti. Soprattutto la possibilità di recuperare i dati è molto scarsa, poiché molti servizi non mettono a disposizione i loro dati a tutti e se li mettono a disposizione non sono ben strutturati.



---

# Progettazione

In questo capitolo introduciamo le scelte progettuali fatte, motivando tali decisioni, e le tecnologie utilizzate per implementare il servizio. Inoltre è descritta l'architettura con cui è stato progettato il sistema. Partendo dalla descrizione delle tecnologie base, man mano vediamo come tutte le altre sono state integrate e fatte interagire per implementare l'idea.

## 2.1 Fusion Table

Google Fusion Tables [FT13] è un innovativo servizio di Google che consente la creazione e la gestione di grandi quantità di dati on-line, non si configura però come un database tradizionale in quanto si concentra sulla condivisione e sulla collaborazione. L'obiettivo concreto di Fusion Tables consiste nel permettere a più utenti di lavorare via web sullo stesso database, rimanendo aggiornati in tempo reale sulle modifiche apportate da altri utenti. I dati di più database, inoltre, possono essere correlati.

Le Google Fusion Tables, infatti, possono essere impostate per essere pubbliche o visibili solo agli utenti che si desidera, i quali a loro volta possono essere eletti a collaboratori, in grado di modificare i dati, o proprietari in grado di invitare altri utenti. Questo servizio gratuito, reso pubblico nella primavera del 2009, permette di gestire database on-line rendendone i dati accessibili a varie applicazioni, tra cui Google Maps.

Sono ora riportate, tratte dal sito ufficiale di Google Fusion Tables, le principali caratteristiche e funzionalità.

### 2.1.1 Visualizzare online i dati

Importare i propri dati: si possono caricare le tabelle di dati da file CSV [CSV13] o fogli di calcolo. Gli sviluppatori possono utilizzare la Fusion Tables API [FTA13] per inserire, aggiornare, cancellare e interrogare i dati di programmazione.

È possibile esportare i dati in formato CSV o KML [KML13]. Visualizzare immediatamente le modifiche: passando alla visualizzazione mappa oppure grafico; è possibile vedere i propri dati immediatamente sotto forma di mappa o grafico.

Usando la funzione filtra si possono selezionare i dati da mostrare. È inoltre possibile condividere le proprie rappresentazioni di dati sul web: una volta realizzata una mappa o un grafico dei dati, è possibile incorporarli in una pagina web o in un blog oppure inviarli come *link* via email. Sarà sempre possibile visualizzare le ultime modifiche apportate ai dati dal proprio pc (gli utenti con cui si scelto di condividere i dati vedranno sempre l'ultima modifica in tempo reale).

### **2.1.2 Costruire Mappe**

Trasformare le tabelle in mappe: punti, linee, poligoni, indirizzi dei clienti, i toponimi e molto altro ancora può essere mappato in pochi minuti con Fusion Tables.

Personalizzare la rappresentazione dei dati: è possibile applicare colori o modificare le icone sulla base dei dati, fare una mappa d'intensità basata su Nazioni, regioni o province, utilizzare poligoni KML per fare mappe d'intensità personalizzate. Mostrare una mappa d'intensità in base alla posizione dei punti e visualizzare migliaia di percorsi in una sola volta.

### **2.1.3 Condividere e controllare i dati on line**

Fusion Table è come un formato di file online che accede tramite un *link* al proprio *dataset* ospitato nei server. Gli utenti del sito web sono in grado di vedere i dati senza eseguire alcun download, essi possono esplorare i dati tramite mappa, grafico, fare calcoli e ricerche. Essi possono anche applicare filtri e scaricare solo il sottoinsieme di dati che si desidera. Chiaramente si può bloccare l'esportazione, dipende dalle impostazioni.

Il servizio permette anche di distribuire sempre la versione più corretta dei dati: si può abilitare l'utilizzo delle tabelle senza creare centinaia di copie dei propri dati sparsi in giro su *hard disk*. Quando i dati sono ospitati nelle tabelle Fusion, gli utenti potranno vedere sempre la versione più recente.

Tramite le API messe a disposizione da Google, è possibile in qualsiasi momento interrogare i dati, semplicemente impostando i permessi agli utenti.

### 2.1.4 Organizzare dati esterni

Le tabelle di utenti differenti possono essere collegate tra loro in modo da formare una nuova tabella Fusion che contenga tutti i dati. Questi dati saranno sempre aggiornati: quando una qualsiasi tabella originale è modificata, la tabella risultante dall'unione mostra sempre i dati aggiornati.

Ogni sottoinsieme di dati può essere reso pubblico o rimanere privato. Se lo si desidera si può quindi creare un set di dati pubblici. Si possono cercare e analizzare questi insiemi di dati e richiamarli tramite l'API Fusion Table. È possibile unire con i confini KML pubblici di un'altra persona a fare una mappa d'intensità. Fusion Table aiuta l'utente a tenere traccia della provenienza dei dati. Durante l'importazione o in qualsiasi momento, è possibile specificare l'attribuzione per i dati che apparirà anche quando i dati sono uniti in altre tabelle.

### 2.1.5 Collaborazione

Si possono invitare collaboratori per visualizzare e/o modificare le colonne: basta inserire gli indirizzi email delle persone con cui si desidera condividere una tabella e inviare loro un invito. Affinchè gli ospiti abbiano sempre una visualizzazione corretta dopo aver applicato filtri, dato un ordine alla tabella e un criterio per rappresentare al meglio i dati, è possibile inviare un collegamento agli ospiti in modo che essi possano vedere esattamente ciò che desideriamo.

Discutere sui dati delle tabelle: gli ospiti possono ottenere un feedback sui dati in base alla colonna, riga o cella. Nelle discussioni si tiene traccia di tutte le cose dette e delle eventuali modifiche effettuate nel tempo. È possibile ottenere le notifiche di eventuali commenti direttamente nella propria casella di posta elettronica.

## 2.2 Foursquare

Senza dubbio Foursquare è il *social network* del momento, il servizio tra i più discussi nei siti che si occupano di *social media* e di web 2.0. Nel seguito verranno illustrate le sue caratteristiche principali che lo hanno reso così popolare.

Foursquare è una rete sociale fondata nel 2009 da Dennis Crowley e Naveen Selvadurai, che si basa sulla geolocalizzazione degli utenti, vale a dire sulla condivisione della propria posizione geografica via GPS attraverso un dispositivo mobile, uno *smartphone*, per i quali sono disponibili apposite applicazioni gratuite.

Gli utenti possono effettuare il cosiddetto *check-in* nei locali, nei negozi, nei musei, in qualsiasi posto di interesse pubblico o privato. Essi sono suggeriti direttamente dall'applicazione e nel caso non lo fossero, possiamo aggiungere un punto d'interesse direttamente noi stessi.

Tutto funziona come una specie di gioco. Maggiore sarà il numero dei *check-in* e dei *tip* (suggerimenti lasciati su un posto), e più punti otterremo per conquistare una lunga serie di *badge*, una sorta di *medaglia al merito* che indica la popolarità e l'influenza di un utente. I commenti possono essere sincronizzati con un account Twitter [TWT13] e Facebook. Ciò significa che è possibile condividerli contemporaneamente anche nei due *social network*.

Le notifiche *push* consentono di trovare gli amici che sono nelle vicinanze o addirittura che si trovano contemporaneamente nello stesso locale. In pratica più si usa Foursquare, maggiori saranno le informazioni a nostra disposizione e maggiore sarà l'incremento dei nostri meriti.

Uno degli obiettivi più ambiti è diventare *major*, vale a dire *il sindaco* di un posto. Tutto questo si consegue portando a termine il maggior numero d'ingressi nello stesso luogo entro due mesi, sempre se tale carica risulti ancora libera ovviamente.

### **2.2.1 Il perché di un successo**

Foursquare è utile per scoprire nuovi posti da esplorare e può essere usato a fini turistici. Da questo punto di vista è destinato a crescere esponenzialmente man mano che la *community* italiana a sua volta sarà più numerosa e partecipativa.

La forza del suo esito straordinario è in continuo aumento, ciò è anche dovuto all'aspetto ludico. Foursquare, infatti, si presenta come un vero e proprio gioco sociale partecipativo. Il meccanismo dei *badge*, favorisce la voglia di diventare *superuser*.

La condivisione della geolocalizzazione permette di incontrare la cerchia degli amici nelle vicinanze o addirittura di ingannare riguardo alla nostra posizione attuale. Non è un caso che Facebook, con la nuova funzione *Places* si metta in sua diretta concorrenza. Per ora Foursquare resta il migliore nel suo settore applicativo, sbaragliando la concorrenza.

Un altro aspetto da tenere in considerazione è quello del cosiddetto social marketing. Foursquare, infatti, può essere utilizzato dalle aziende a scopo promozionale, per pubblicizzare il proprio brand. C'è chi offre sconti, prezzi e tariffe speciali, inviti e agevolazioni ai clienti migliori in base al numero di *check-in* o addirittura la possibilità

di ottenere nuovi ambiti *badge*. Foursquare in questo senso sta pensando a incrementare il proprio piano di partnership con alcune importanti catene di negozi e di media americani.

Insomma, le sue possibilità di utilizzo sono ancora inesplorate e tra le persone nascono nuove forme di aggregazione e di gioco destinate ad avere più o meno fortuna.

Importante è tenere in considerazione gli aspetti di privacy. Chi usa Foursquare deve essere a conoscenza che sta condividendo dati riguardanti la sua *location privacy*. Il sistema permette di restringere la sfera delle amicizie e l'autorizzazione a visualizzare i *check-in* a poche persone affidabili oppure anche di mantenere semplicemente i *check-in* privati, evitando di dividerli con gli amici.

L'affermazione definitiva di Foursquare in Italia dipende da diversi fattori. Nello specifico, sicuramente dalla diffusione delle reti *Wi-Fi* gratuite di cui il nostro paese è davvero carente e dalla recezione del potenziale commerciale da parte di aziende sponsor, spesso restie a confondere l'aspetto ludico con quello legato al business.

## 2.3 Tecnologie utilizzate

Vediamo nel dettaglio le tecnologie di sviluppo utilizzate. Partendo dalle API che sono lo strumento utilizzato per usufruire dei servizi messi a disposizione da Fusion Table e da Foursquare. Saranno inoltre descritte la tecnologia per OAuth 2.0 per gestire la sicurezza dell'autenticazione e le tecnologie web impiegate per lo scambio e l'elaborazione delle informazioni.

### 2.3.1 API

*Application programming interface* si indica ogni insieme di procedure disponibili al programmatore, di solito raggruppate a formare un set di strumenti specifici per il compimento di determinati compiti all'interno di un certo programma. Spesso con tale termine s'intendono le librerie software disponibili in un certo linguaggio di programmazione. Nascono con l'intento di semplificare il lavoro di programmazione software. Le API quindi rappresentano un livello di astrazione intermedio.

### **2.3.2 OAuth 2.0**

*Open Authorization*, è un protocollo di comunicazione mediante il quale un'applicazione (o un servizio web) può gestire in modo sicuro l'accesso autorizzato ai dati sensibili. Il protocollo è compatibile con qualsiasi tipologia di applicazione: desktop, web e mobile.

Il protocollo OAuth è stato ideato da Blaine Cook nel 2006, mentre lavorava all'implementazione Twitter di OpenID, proponendosi come alternativa aperta ai molti protocolli proprietari già esistenti, come Google AuthSub, AOL OpenAuth, Yahoo BBAuth, Flickr API e tanti altri. Dalla versione 1.0 (pubblicata nel 2007) OAuth, ha subito diverse revisioni, recependo via via le RFC (*Request For Comments*) [RFC13] proposte dai vari esperti.

In generale le RFC sono un insieme di documenti di riferimento per la Comunità Internet che riportano informazioni o specifiche riguardanti nuove ricerche, innovazioni e metodologie dell'ambito informatico. Gli esperti possono pubblicare dei *memorandum* per esporre nuove idee da includere negli standard: seguendo questo iter è stata recentemente pubblicata la prima bozza del protocollo OAuth 2.0, la bozza è divenuta negli anni uno standard solido e robusto. I vari principali erogatori di servizi, come ad esempio i social network (Facebook, Twitter, LinkedIn [LKI13] e tanti altri), hanno già annunciato il supporto anche prima che la prima bozza è stata pubblicata ufficialmente.

#### **2.3.2.1 Service Provider e Consumer**

Gli attori coinvolti sono tre:

1. l'applicazione che richiede le informazioni, o Consumer (la nostra applicazione);
2. il servizio web che fornisce le informazioni, o Service Provider (ad esempio Facebook o Twitter);
3. l'utente che amministra l'applicazione, deve essere iscritto sulla piattaforma del Service Provider.

Per usufruire del servizio predisposto dal Service Provider, è necessario che il *Consumer* registri la sua applicazione. Con quest'operazione, il Service Provider fornisce al consumer le chiavi segrete (*Consumer Key e Consumer Secret*) che saranno utilizzate per lo scambio dei dati.

I principali passaggi della comunicazione *Consumer – Service Provider*, sono i seguenti.

- 1) Il Consumer ridirige l'utente ad un URL predisposto dal Service Provider per la fase di autenticazione. Nella richiesta HTTP sono presenti le due chiavi ricevute in fase di registrazione e un'URL al quale l'utente deve essere ridiretto al termine dell'operazione.
- 2) L'utente accede alla pagina di autenticazione predisposta dal Service Provider dove inserisce le proprie credenziali di accesso, di solito username e password. Normalmente la pagina di autenticazione lavora con il protocollo HTTPS, il quale garantisce che i dati scambiati tra browser e web server sono codificati e non siano trasferiti in chiaro come invece avviene nel protocollo HTTP. In questo caso la comunicazione risulta molto più sicura in caso di intercettazione.
- 3) Il Service Provider verifica l'identità dell'utente. Se la verifica ha esito positivo, gli presenta il tipo di richiesta avanzata dal Consumer che l'utente deve approvare. Tale richiesta riguarda, di solito, la durata dell'accesso e la tipologia d'informazioni che saranno rese disponibili al Consumer.
- 4) Il Service Provider ridirige l'utente all'URL inviata in precedenza dal Consumer. Nella richiesta HTTP è presente una stringa (detta *token*) che deve essere utilizzata successivamente dal Consumer per richiedere eventuali informazioni al Service Provider.
- 5) L'utente accede nuovamente alla pagina del Consumer. Per l'utente meno esperto, il passaggio da un server a un altro, è praticamente trasparente.
- 6) Il Consumer invia una HTTP al Service Provider nella quale richiede, ad esempio, i dati anagrafici dell'utente corrente. Nella richiesta deve anche fornire il *token* precedentemente ricevuto. Le informazioni disponibili dipendono dal Service Provider. Facebook, ad esempio, espone i dati anagrafici dell'utente oppure i suoi ultimi messaggi in bacheca; LinkedIn espone i collegamenti di un utente oppure le sue iscrizioni ai gruppi; Google fornisce moltissime informazioni come ad esempio l'elenco dei contatti in rubrica.

A differenza del protocollo Client/Server, il protocollo OAuth, come abbiamo potuto notare, non obbliga l'utente a fornire le credenziali di accesso al provider (il sistema esterno). L'utente è ridiretto sul server del Service Provider per l'autenticazione al sistema.

### 2.3.3 HTTPS e SSL

HTTPS è il risultato dell'applicazione del protocollo di crittografia SSL in sovrapposizione al normale protocollo di trasferimento ipertestuale (HTTP). Viene utilizzato per gestire informazioni sensibili per le quali è importante garantire un elevato livello di riservatezza ed una garanzia di integrità.

SSL (*Secure Sockets Layers*) [CSSL13] è una procedura per rendere sicure le transazioni effettuate via Internet. Lo standard SSL è stato elaborato da Netscape, in collaborazione con Mastercard, Bank of America, MCI e Silicon Graphics. Si basa su un processo di crittografia a chiave pubblica per garantire la sicurezza della trasmissione dei dati su internet.

Il suo principio consiste nello stabilire un canale di comunicazione sicuro (cifrato) tra due terminali (un client e un server) dopo una tappa di Autenticazione.

Il sistema SSL è indipendente dal protocollo usato, il che significa che può rendere sicure sia delle transazioni fatte sul Web tramite il protocollo HTTP sia delle connessioni via protocollo FTP, POP o IMAP. In effetti, SSL agisce come un livello supplementare, permettendo di assicurare la sicurezza dei dati, situati tra il livello applicazione e il livello trasporto (protocollo TCP ad esempio).

In questo modo, SSL è trasparente per l'utente (cioè l'utente può ignorare che utilizza SSL). Ad esempio un utente che usa un browser internet per connettersi ad un sito di commercio elettronico sicuro via SSL invierà dei dati cifrati senza alcuna manipolazione necessaria da parte sua.

La quasi totalità dei *browser web* supportano ormai il protocollo SSL, visualizzano ad esempio un lucchetto chiuso per indicare la connessione ad un sito sicuro via SSL.

### 2.3.4 JSON

*JSON (Javascript Object notation)* [IJS11], è un tipo di formato molto utilizzato per lo scambio dati in applicazioni client server come API o *Mashup*. È basato su *JavaScript* ma il suo sviluppo è specifico per lo scambio di dati ed è indipendente dallo sviluppo del linguaggio di *scripting* dal quale nasce e con il quale è perfettamente integrato e semplice da utilizzare.

È una valida alternativa al formato XML-XSLT e sempre più servizi di *Web Services* mettono a disposizione entrambe le possibilità di integrazione. Leggere e interpretare uno *stream* JSON è semplice in tutti i linguaggi, non solo in JavaScript [JSP13], con cui è completamente integrato ma anche con PHP [PHP13], Java [JAVA13] ed altri linguaggi server-side, per mostrare i dati da fonti esterne ed impaginarli secondo le proprie soluzioni personalizzate.

JSON prende origine dalla sintassi degli oggetti letterali in JavaScript. Un oggetto letterale può essere definito così:

```
var JSON = {  
    proprieta1: 'Valore',  
    proprieta2: 'Valore',  
    proprietaN: 'Valore'  
}
```

Si tratta di coppie di proprietà/valori separate dalla virgola ad eccezione dell'ultima. L'intero oggetto è racchiuso tra parentesi graffe. A differenza di questa notazione JavaScript, che può contenere anche funzioni e valori complessi, JSON ammette solo valori semplici e atomici, tra cui:

1. stringhe;
2. numeri;
3. array;
4. oggetti letterali;
5. true;
6. false;
7. null.

Un esempio:

```
{  
    "home": "mioSito.it",  
    "link": "http://www.mioSito.it",  
    "argomento": "Standard del web",  
    "aree": [  
        {
```

```
    "area": "CSS",
    "url": "http://css.mioSito.it"
  },
  {
    "area": "Basic",
    "url": "http://basic.mioSito.it"
  }
]
```

JSON inizia con una coppia di parentesi graffe che racchiudono il corpo dell'intera struttura. Seguono poi le coppie di proprietà e valori che, come in questo esempio, possono contenere i valori atomici elencati sopra. A differenza degli oggetti letterali, JSON richiede che i nomi delle proprietà e i valori stringa siano racchiusi tra doppie virgolette. Per accedere a ciascuno dei membri di un oggetto JSON utilizziamo la tradizionale notazione JavaScript:

```
oggetto.home;
oggetto.link;
var aree = oggetto.ree; // Array;
for(var i = 0; i < aree.length; i += 1) {
    var oggettoArea = aree[i];
    var area = oggettoArea.area;
    var url = oggettoArea.url;
}
```

JSON viene usato da JavaScript tramite AJAX [AJAX13]. Quando la richiesta ha luogo, è restituito l'oggetto mostrato sopra, cui possiamo accedere con la sintassi in precedenza indicata. Se si utilizza jQuery [JQR13], possiamo scrivere:

```
$.ajax({
    url: 'json.php',
    type: 'GET',
```

```
dataType: 'json',
success: function(oggetto) {

    var sito = $(oggetto).home;
    var url = $(oggetto).link;
    var aree = $(oggetto).aree;
    //...
}
});
```

### 2.3.5 JavaScript

JavaScript è un linguaggio di *scripting* lato-client, che viene interpretato dal browser. Il web funziona a due livelli: le pagine web sono inviate all'utente da un *web server*, cioè da un programma che si trova su un computer remoto, e che per lo più non fa nient'altro che inviare le pagine a chi ne fa richiesta [JST13].

L'utente da casa visualizza sul proprio *browser* le pagine che gli sono state inviate. Un *browser* è un programma che permette di leggere le pagine scritte in linguaggio HTML: si tratta di Internet Explorer, Netscape Navigator, Opera e altri.

Quando visualizziamo le nostre pagine web da un dispositivo ci sono dunque due computer che si parlano: il server e il client.

Alcuni linguaggi di *scripting* sono eseguiti dal web server (si chiamano appunto linguaggi server side o lato server). JavaScript, invece, è eseguito sul nostro computer dal browser (è un linguaggio client side).

Dire che JavaScript è un linguaggio lato client, significa anche che i vostri script avranno validità all'interno delle singole pagine web, e non da una pagina all'altra: con JavaScript è possibile passare una piccola quantità di dati da una pagina all'altra, ma è un'operazione che può essere effettuata con una certa difficoltà (con linguaggi server side, si esegue invece in maniera intuitiva); non è possibile invece trasmettere quantità di dati elevate.

Inoltre JavaScript è un linguaggio di *scripting*: questo significa che la sintassi JavaScript può essere integrata direttamente dentro una pagina HTML, senza bisogno di produrre alcun file compilato.

Con i linguaggi di programmazione invece (come il C, il C++) [ANSI13] si scrive la sintassi, e poi si passa a un compilatore, che produce un file compilato, in cui la sintassi è scomparsa.

JavaScript invece non è compilato: è possibile visualizzare in qualsiasi momento il codice di una pagina HTML e leggere le righe di sintassi JavaScript.

Dire che è un linguaggio di scripting sottintende dunque il fatto che sia un linguaggio interpretato: come abbiamo visto, non esiste nessun compilatore, ma è direttamente il browser che tramite un apposito motore di scripting (cioè di visualizzazione), che legge le parti di codice JavaScript.

### 2.3.6 PHP

PHP (acronimo ricorsivo per PHP: *Hypertext Preprocessor*) [PHP13] è un linguaggio di *scripting general-purpose open source* molto utilizzato, è specialmente indicato per lo sviluppo web e può essere integrato nell'HTML.

Vediamo un esempio:

```
Example #1 Un esempio introduttivo
<!DOCTYPE HTML PUBLIC "-//
//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
  <head>
    <title>Esempio</title>
  </head>
  <body>
    <?php
      echo "Ciao, sono uno script PHP!";
    ?>
  </body>
</html>
```

Invece di un sacco di comandi per produrre HTML (come per altri linguaggi tipo C o Perl), le pagine PHP contengono HTML con codice incorporato che fa qualcosa. Il codice PHP è delimitato da speciali istruzioni di elaborazione d'inizio e fine `<?php ?>` che permettono di entrare e uscire dalla *modalità* PHP.

Ciò che distingue PHP da altri linguaggi di scripting del tipo client-side come JavaScript è che il codice è eseguito nel server, generando HTML che sarà dopo inviato

al client. Il client riceve i risultati dell'esecuzione dello script, ma non potrà conoscere qual è il codice eseguito. Potete persino configurare il vostro web server per processare tutte i vostri file HTML con PHP e allora non ci sarebbe realmente alcun modo per gli utenti di sapere cosa avete sul vostro server.

La cosa più interessante nell'uso di PHP è che si tratta di un linguaggio molto semplice per il neofita, ma che, tuttavia, offre molte soluzioni avanzate al programmatore professionista.

### **2.3.7 Google Maps**

Google Maps (nome precedente Google Local) è un servizio accessibile dal relativo sito web che consente la ricerca e la visualizzazione di mappe geografiche di buona parte della Terra.

Oltre a questo è possibile ricercare servizi in particolari luoghi, tra cui ristoranti, monumenti, negozi, trovare un possibile percorso stradale tra due punti e visualizzare foto satellitari di molte zone con diversi gradi di dettaglio (per le zone che sono state coperte dal servizio si riescono a distinguere in molti casi; case, giardini, strade e così via). Le foto sono statiche (non in tempo reale), buona parte delle quali sono riferite alla fine degli anni novanta. Oltre a queste funzioni, Google Maps offre anche una ricerca di attività commerciali sulle stesse mappe.

Per mostrare risultati pertinenti, Google Maps combina le informazioni da molte fonti. Le schede includono informazioni provenienti da:

- nostri risultati di ricerca web;
- i dati inviati direttamente dai proprietari di attività commerciali locali;
- fotografie inviate dagli utenti;
- immagini *Street View*;
- fonti di terze parti (ad esempio le Pagine Gialle).

#### **2.3.7.1 Caratteristiche**

È possibile visualizzare il nostro pianeta come mappa o come semplice cartina stradale. Ultimamente solo per alcuni paesi, è stata implementata la funzione traffico che visualizza i problemi di traffico stradale.

Google Maps è basato su una variante della Proiezione di Mercatore. Se la Terra fosse perfettamente sferica, la proiezione sarebbe la stessa di quella di Mercatore.

Google Maps usa le formule della proiezione sferica di mercatore, ma le coordinate di sono del GPS basate sui dati del sistema geodetico mondiale WGS 84. La differenza tra una sfera e un ellissoide WGS 84 è la causa di una non precisa conforme proiezione, percepibile solo a piccola scala.

Assumendo che  $dE$  and  $dN$  siano le componenti delle coordinate locali ENU infinitesimali la loro larghezza e lunghezza proiettate sulla carta sono descritte così:

$$dx = a_{\text{map}} \left( \frac{a}{\sqrt{1 - e^2 \sin^2 \varphi}} \right)^{-1} \sec \varphi dE,$$

$$dy = a_{\text{map}} \left( \frac{a(1 - e^2)}{(1 - e^2 \sin^2 \varphi)^{3/2}} \right)^{-1} \sec \varphi dN,$$

dove  $\varphi$  è la latitudine geodetica,  $e$  è la prima eccentricità dell'ellissoide della Terra,  $a$  è il semiasse maggiore della Terra, e  $a_{\text{map}}$  è la scala della carta. Google Maps usa

$$a_{\text{map}} = \frac{256 \times 2^{\text{zoomLevel}}}{2\pi} \text{ pixels.}$$

Poiché la proiezione di Mercatore e le sue varianti ai poli hanno valori infiniti Google Maps non mostra i poli [WGM13].

### 2.3.8 Database spaziali

Per ogni strato informativo occorre definire una colonna che contenga le coordinate, e un sistema che gestisca in maniera differenziata le tabelle che contengono dati geometrici di tipo differente. Occorre inoltre introdurre un sistema che verifichi che i dati inseriti siano congruenti e che sia sempre verificata la bidimensionalità. Interrogazioni veloci alle tabelle sono la ragione d'essere dei database (assieme al supporto per le transazioni).

I database si differenziano proprio per la maggior robustezza e prestanza degli indici. Tutto avviene usando la sintassi SQL [SQL13]. Esistono comunque funzioni che costruiscono immediatamente tutta la struttura necessaria alla gestione dei dati territoriali.

Analogamente le interrogazioni al database avvengono utilizzando *SQL query* sfruttando combinazioni di dati, funzioni e di test vero/falso. Per comprendere come funzioni una query spaziale occorre tenere presenti due cose:

- esistono gli indici spaziali;
- le interrogazioni spaziali sono molto dispendiose in termini di calcolo se effettuate su un gran numero di entità geometriche.

Gli indici spaziali esistono per migliorare l'efficienza delle query. I dati geografici vengono “aggregati” e “amministrati” in gruppi spaziali distinti. In questo senso gli indici sono *lossy*: sono definiti per semplificare e nella semplificazione perdono informazioni. Ad esempio: spesso si vuole utilizzare l'operatore intersezione && che testa se il rettangolo che circonda le entità geometriche ne interseca altre. Questa funzione è ottimizzata per l'utilizzo degli indici spaziali: serve a scremare i dati per eseguire una ricerca più raffinata usando le funzioni messe a disposizione. Le funzioni che utilizzano l'indice spaziale servono per identificare le geometrie che possono soddisfare a una condizione. Le funzioni spaziali testano la condizione esattamente.

### 2.3.9 Fusion Table API

Il Google Fusion Tables API 1.0 [GFTA13] è una suite di API che consente agli sviluppatori di creare e gestire le risorse Fusion Tables, quali tabelle, colonne, righe, modelli e stili.

Gli sviluppatori utilizzano le istruzioni SQL per eseguire query per manipolare tabelle, colonne, stili e modelli.

#### 2.3.9.1 Concetti di base

Google Fusion Tables è costruito sui seguenti concetti.

- Table: le tabelle sono costituite da dati disposti in righe e colonne, simile a un foglio di calcolo. Ci sono tre tipi di tabelle: di base, vista, e unite:
  - Base: Una tabella creata da caricamento dei dati provenienti da un archivio o attraverso l'inserimento di righe con le istruzioni SQL SELECT. Questa è la tabella da cui si creano le viste;
  - Vista: Tabella creata selezionando un sottoinsieme di colonne da una tabella di base con o senza filtri sui dati di riga. Una vista è una tabella virtuale, con



- *Row*: i dati delle Fusion Table sono salvati nelle righe, ciascuna delle quali ha una ROWID generata internamente che identifica univocamente all'interno di una tabella. Per individuare una riga specifica di dati, si deve eseguire un'istruzione di query SQL con criteri per abbinare quella riga. La risposta include ROWID della riga, che è ora possibile utilizzare per aggiornare o eliminare quella riga.

### 2.3.9.2 Operazioni base

Per le tabelle, colonne, template e stili è utilizzata HTTP Request come descritto nella tabella.

Operation	Description	HTTP mappings
<b>list</b>	Elenca tutte le risorse del tipo	<b>GET</b> effettuata nella lista delle risorse URI.
<b>get</b>	Seleziona una risorsa specifica	<b>GET</b> effettuata nella risorsa URI.
<b>insert</b>	Crea una nuova risorsa	<b>POST</b> sulla lista delle risorse URI, dove vengono passati i dati che descrivono la nuova risorsa.
<b>update</b>	Aggiorna una specifica risorsa	<b>PUT</b> su di una risorsa URI, dove vengono passati i dati che descrivono la nuova risorsa.
<b>delete</b>	Elimina una specifica risorsa	<b>DELETE</b> di una risorsa URI.

### 2.3.9.3 Sending requests

Google Fusion Table API 1.0 supporta i seguenti approcci per inviare una richiesta al server Fusion Table:

- utilizza REST per tabelle, colonne, stili e *template*;
- utilizza SQL per le righe.

Indipendentemente da quale si utilizza, l'URI è:

```
https://www.googleapis.com/fusiontables/v1
```

Se una tabella è esportabile che sia pubblica o privata, è possibile inviare una richiesta GET direttamente dalla barra degli indirizzi del vostro browser, ma è necessario includere una chiave API. Vediamo in seguito degli esempi.

Questo esempio elenca tutte le colonne di una tabella.

```
https://www.googleapis.com/fusiontables/v1/tables/1KxVV0wQXhXhMScSDuqr-0Ebf0YEt4m4xzVplKd4/columns?key=your API key
```

Questo esempio esegue una query per selezionare tutte le righe di una determinata tabella.

```
https://www.googleapis.com/fusiontables/v1/query?sql=SELECT * FROM 1KxVV0wQXhXhMScSDuqr-0Ebf0YEt4m4xzVplKd4&key=your API key
```

#### ***2.3.9.4 Identificare l'applicazione***

Ogni richiesta inviata dall'applicazione alle Fusion Table ha bisogno di essere identificata. Ci sono due modi per identificare la vostra applicazione: si utilizza un token di OAuth 2.0 (che autorizza anche la richiesta) e/o con il tasto applicazione API. Ecco come fare per determinare quale di queste opzioni utilizzare:

- se la richiesta richiede l'autorizzazione (ad esempio una richiesta di dati privati di un individuo), quindi l'applicazione deve fornire un token OAuth 2.0 con la richiesta. L'applicazione può anche fornire la chiave API, ma non è necessario;
- se la domanda non richiede l'autorizzazione (ad esempio una richiesta di dati pubblici), allora l'applicazione deve fornire sia la chiave API o un gettone OAuth 2.0, o entrambi.

Sui protocolli di autorizzazione si consiglia di utilizzare OAuth 2.0 per autorizzare le richieste.

La Fusion Tables API supporta anche vecchie scelte di autorizzazione, come OAuth 1.0, Client Login, tuttavia, nella maggior parte dei casi non è consigliabile l'utilizzo di tali opzioni. Se l'applicazione utilizza già queste possibilità, si consiglia la migrazione a OAuth 2.0, se possibile.

Se l'applicazione ha determinati requisiti di autorizzazione insoliti, come la registrazione allo stesso tempo, la richiesta di accesso ai dati (ibrido) o il dominio a livello di delega di autorità, allora non si può utilizzare OAuth 2.0. In questi casi, è necessario invece utilizzare OAuth 1.0 il token e una chiave API.

È possibile trovare la chiave API dell'applicazione in Google API Console, nella sezione Accesso API semplice del riquadro di accesso della stessa.

Richieste alle tabelle API Fusion per i dati utente non pubblici, devono essere autorizzati da un utente autenticato. I dettagli del processo di autorizzazione, per OAuth 2.0 variano leggermente secondo il tipo di applicazione che si sta scrivendo. Il seguente processo generale si applica a tutti i tipi di applicazioni:

1. quando si crea l'applicazione, viene registrata con Google. Google quindi fornisce le informazioni di cui ha bisogno in seguito, ad esempio un ID cliente e un client secret;
2. attivare la Fusion Tables API nel riquadro servizi della console API di Google;
3. quando l'applicazione richiede l'accesso ai dati degli utenti, si chiede a Google per un particolare ambito di accesso;
4. Google visualizza una finestra di OAuth per l'utente, chiedendo loro di autorizzare l'applicazione per richiedere alcuni dei loro dati;
5. se l'utente approva, allora Google rilascia un token di accesso di breve durata;
6. ogni volta che un'applicazione richiede dei dati ai server Google, deve abbinare alla richiesta il token;
7. se viene stabilito che la richiesta e il token sono validi restituisce i dati richiesti.

Alcuni flussi includono passaggi aggiuntivi, come l'utilizzo di token di aggiornamento per acquisire nuovi token di accesso.

#### ***2.3.9.5 Utilizzare i dati Geografici***

Le Fusion Tables rendono semplice la mappatura in un database di luoghi ed eseguire l'esecuzione di query spaziali su di esso. Per visualizzare i risultati della query su una mappa di Google, è possibile utilizzare l'interfaccia utente oppure è possibile utilizzare l'API con il Fusion Tables Layer Maps[FTLM13] API v3.0.

- solo le prime 100.000 righe di dati in una tabella sono mappate o incluse nei risultati delle query spaziali. Pertanto mappe realizzate con Fusion Tables nella API di Google Maps o con l'applicazione web Fusion Tables verranno visualizzati solo i dati all'interno di 100.000 righe;

- durante l'importazione o l'inserimento di dati, la dimensione totale dei dati inviati in una chiamata API non può superare 1 MB. Chiamate API per scrivere i dati non dovrebbero superare 0,5 QPS per tabella;
- una cella di dati in Fusion Tables supporta un massimo di 1 milione di caratteri; talvolta può essere necessario ridurre la precisione delle coordinate o semplificare il poligono o la linea di descrizione;
- il numero massimo di vertici supportato per ogni tabella è di 5 milioni. Se si guarda la mappa, è possibile notare come v che vengono visualizzati.

Una posizione geografica è una stringa che contiene uno dei seguenti valori: un indirizzo, il nome di una città, il nome di un paese, o di qualunque descrizione luogo compreso da Google Maps.

Latitudine e longitudine possono essere salvati in formato numerico decimale oppure in formato KML [KML13]. Il formato KML definisce una coordinata geografica, viene utilizzato per aggiungere una posizione geografica in una tabella.

È possibile memorizzare le coordinate geografiche di punti, linee e poligoni in una tabella utilizzando KML. Per le coordinate geografiche Fusion Table mette a disposizione una apposita colonna che contiene le informazioni spaziali che interessa rappresentare nella mappa.

### **2.3.10 Foursquare API**

Uno degli aspetti più interessanti che Foursquare mette a disposizione è sicuramente quello del reperimento della gran mole di dati tramite API. Il portale dedicato in primis agli sviluppatori, senza però dimenticare gli ingegneri che progettano le applicazioni e i partner business. Quindi l'obiettivo è cercare di soddisfare trasversalmente categorie eterogenee di persone, facendo loro recuperare le informazioni di cui avevano bisogno nel più breve tempo possibile [FSP13].

Dietro a questo importante lavoro c'è Alice Lee che, con il supporto del platform evangelist di Foursquare Akshay Patil, ha cercato di mutuare le esperienze di altri siti tecnici come Facebook e .NET, e ha pensato di approcciare il problema della fruizione dei contenuti dividendo i possibili visitatori in tre categorie: i *newbies*, ovvero coloro che si avvicinano per la prima volta al sito, i visitatori abituali e i visitatori business.

Per i *newbies* si è cercato di favorire l'avvicinamento all'esperienza dello sviluppo tramite una sequenza logica di passi da seguire: si parte da un invitante bottone *Getting Started* inserito in bella vista al centro della Home Page; si prosegue con la presentazione dei casi più comuni di utilizzo supportati da esempi per poi finire con l'approfondimento del protocollo OAuth 2.0, uno dei più usati per l'autenticazione e punto cardine per la sicurezza e l'accesso ai dati.

Per i visitatori abituali l'obiettivo principe era far loro raggiungere le pagine della documentazione, i cosiddetti *endpoints*, nella maniera più veloce possibile.

Per i clienti business è stata creata una vetrina in cui si possono vedere gli esempi di applicazioni sviluppate con il supporto delle API di Foursquare; per raggiungerla basta semplicemente cliccare il link presente nelle Home in basso a destra *Apps Showcase*.

La Home Page si completa con un'interessante sezione *Explore*, all'interno della quale è possibile testare le chiamate direttamente dal proprio browser visualizzando la sintassi delle risposte, e con il link al *Forum* in cui si possono postare i quesiti sulla piattaforma Foursquare.

### 2.3.10.1 Funzionamento

L'API di Foursquare permette agli sviluppatori di applicazioni di interagire con la piattaforma Foursquare. L'API è di per sé un insieme d'indirizzi da cui è possibile inviare richieste, quindi non c'è davvero nulla da scaricare sul vostro server. È possibile utilizzare un insieme di librerie per aiutare a semplificare richieste e risposte.

È possibile richiedere l'output in formato XML [XML13] o JSON, fare richieste utilizzando URL che assomigliano a questo:

```
http://api.foursquare.com/v1/user.json.
```

Le API non si fermano a leggere solamente i dati di un utente. È possibile utilizzare sia il metodo GET che POST, ma si può anche fare cose utili come il check-in, inserimento di consigli nei luoghi visitati e la creazione di pagine che descrivono un luogo. Per quanto riguarda i limiti di velocità, l'applicazione è limitata a 200 richieste all'ora per metodo.

Per la maggior parte delle richieste alle API Foursquare, si deve utilizzare l'autenticazione di base o OAuth per sfruttare appieno e loro potenzialità.

### ***2.3.10.2 Strumenti disponibili***

Le API di Foursquare mettono a disposizione un metodo veloce e intuitivo per raccogliere le informazioni che ci interessano. Infatti tutte le informazioni sono ben strutturate e facili da comprendere. Vediamo in generale la struttura.

- **User:** contiene le informazioni del profilo dell'utente che si vuole selezionare. Se l'utente è un amico vengono aggiunte informazioni aggiuntive. Possiamo accedere a varie informazioni dai dati anagrafici, i punteggi, le foto, i luoghi visitati oppure gestire le operazioni che si possono effettuare sugli amici, come ad esempio approvazione e richiesta di amicizia.
- **Venues:** contiene i dettagli di un luogo, inclusa la locazione geografica, i sindaci, i tag, i consigli e le categorie.
- **Checkins:** contiene tutti i dettagli di un check-in.
- **Tip:** sono i commenti/consigli lasciati da un utente su un luogo.
- **Photos:** contiene tutti i dettagli delle foto che sono stata caricate su Foursquare.
- **Event:** crea o gestisce un evento che è associato al luogo amministrato dall'utente. Gli eventi possono essere visualizzati sotto forma di calendario anche nel web.
- **Pages:** permette all'utente di creare e gestire pagine.

### ***2.3.10.3 Real-Time API***

Real-Time API permettono alla nostra applicazione di ottenere in tempo reale le informazioni sui check-in in Foursquare. Il nostro server riceverà una richiesta HTTP ogni volta che è eseguito un check-in.

Questo tipo di API è stata progettato per le applicazioni i cui gli utenti sono responsabili delle sedi, consentendo agli amministratori di ottenere una visione in tempo reale del traffico nella loro sede, o utilizzare applicazioni che consentono l'automazione foursquare-powered dei luoghi visitati da un utente. Per consentire a un'applicazione di ricevere informazioni da un luogo, almeno uno dei responsabili di questa sede deve aver autorizzato l'uso della vostra applicazione, altrimenti si rischia che Foursquare notifichi un errore, dal momento che non si è autorizzati a ricevere la risorsa.

Quando l'applicazione riceve una notifica di tipo luogo, il server riceve una richiesta POST che contiene un parametro in cui sono raccolte tutte le informazioni su

un check-in effettuato. All'interno del parametro check-in ci sono le informazioni su un normale check-in. La differenza principale tra le Push API e le API standard è che le informazioni sono anonime, il che significa che alcune informazioni private non saranno incluse.

### 2.3.11 Tecnologia Push

La tecnologia *push* o *server push* [TSP13] descrive uno stile di comunicazione basata su Internet in cui la richiesta per una determinata operazione è disposta dall'editore o server centrale. La tecnologia push si basa su informazioni espresse in anticipo chiamato modello *pubblica/sottoscrivi*, un client apre un canale di comunicazione con il server. Ogni volta che sono presenti dei dati nel canale il server invia le informazioni al client.

#### 2.3.11.1 HTTP server push

HTTP server push (noto anche come streaming HTTP) è un meccanismo per l'invio di dati da un server web a un browser web. Il Push server HTTP può essere raggiunto attraverso diversi meccanismi.

In genere il server web non termina una connessione dopo che i dati di risposta sono stati inviati a un client. Il server web lascia aperta la connessione in modo che se un evento è ricevuto dal server, può essere inviato immediatamente a uno o più client, altrimenti i dati dovrebbero essere in coda finché non sarebbe arrivata la richiesta dal client. La maggior parte dei server web offrono questa funzionalità tramite CGI (Common Gateway Interface) [CGI13].

Un altro meccanismo è legato a un particolare tipo MIME chiamato *multipart/x-mixed-replace*, che è stato introdotto da Netscape nel 1995. I browser web interpretano questo come un cambiamento dei documenti ogni volta che il server ne invia una nuova versione al client. Questa tecnologia si può applicare ai documenti HTML, ma anche per lo streaming d'immagini in applicazioni webcam.

Tutto questo è incluso in HTML5 con l'API WebSocket, che consente a un server Web e un client di comunicare tramite una connessione TCP full-duplex.

### **2.3.11.2 Pushlet**

Utilizzando questa tecnica, il server sfrutta connessioni HTTP persistenti, lasciando la risposta perennemente *aperta* (ossia, il server non termina la risposta), effettivamente è ingannato dal browser a rimanere in stato di caricamento. Dopo il caricamento iniziale della pagina, la procedura potrebbe essere considerata completa.

Il server invia periodicamente frammenti di JavaScript per aggiornare il contenuto della pagina, ottenendo in tal modo la capacità di gestire le informazioni in coda. Utilizzando questa tecnica, il client non ha bisogno di applet Java o altri plug-in al fine di mantenere una connessione aperta al server; ad esso arrivano notifiche automaticamente sui nuovi eventi.

Un grave inconveniente di questo metodo, è la mancanza di controllo che il server ha sui tempi navigatore fuori; un aggiornamento della pagina è sempre necessario se si verifica un time-out nel browser.

### **2.3.11.3 Long polling**

Long polling non è di per sé una vera tecnologia push; è una variazione della tecnica tradizionale polling, ma permette emulando un meccanismo di inoltro, lo scambio di informazioni in circostanze in cui una richiesta push reale non è possibile.

Con long polling, il client richiede informazioni dal server in modo simile a un polling normale. Se il server non dispone di informazioni immediatamente utilizzabili per il client, allora invece di inviare una risposta vuota, il server trattiene la richiesta in memoria e attende che il client inoltri la richiesta. Dopo di che una risposta completa è inviata al client.

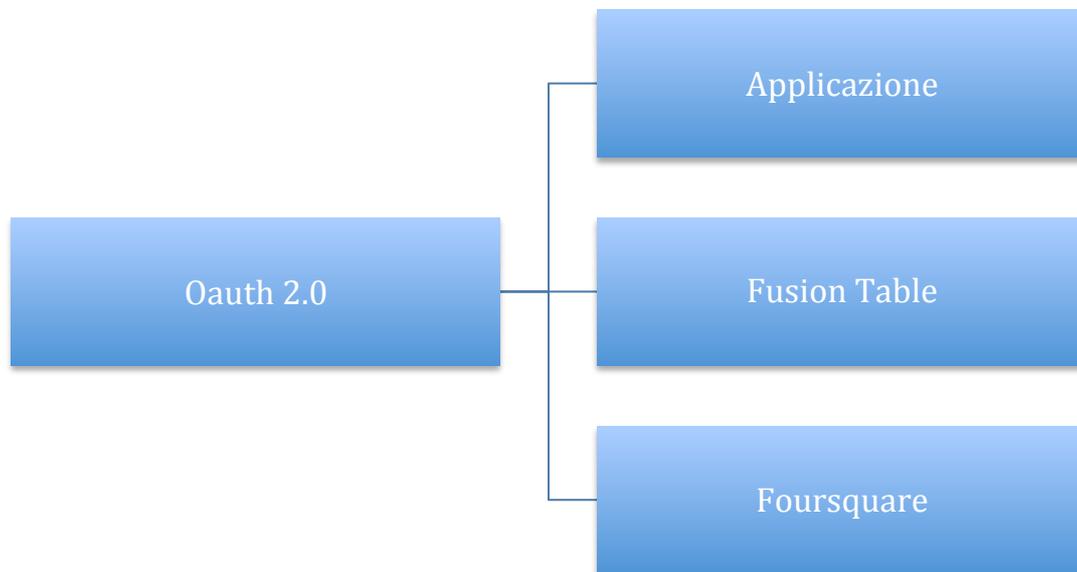
## **2.4 Architettura**

Andiamo ora ad analizzare come queste tecnologie sono state utilizzate nel progetto, come interagiscono tra di loro formando l'architettura base.

Riprendiamo le principali tecnologie che sono state utilizzate.

- Fusion Table: utilizzate come base di dati per contenere tutte le informazioni raccolte. Utilizzato anche come Database spaziale dal quale è costruita la mappa contenente tutte le segnalazioni con la loro descrizione.
- Foursquare: utilizzato come servizio consolidato per la raccolta di dati di utenti in mobilità. Quindi servizio molto utile nel contesto applicativo.

- OAuth: richiesto dalle API sia di Foursquare sia di Fusion Table come mezzo per rendere sicuro lo scambio di informazioni.



*Figura 2.1 Diagramma generale descrizione Architettura.*

Come possiamo notare nella figura 2.1 l'applicazione interagisce con i servizi di Foursquare e Fusion Table utilizzando OAuth 2.0 come protocollo per la sicurezza dei dati.

### **2.4.1 Connessione a Foursquare**

Entrando in dettaglio nella descrizione dell'architettura vediamo il primo passo che viene effettuato, la connessione al servizio Foursquare.

La connessione avviene tramite la richiesta dell'utente di connettersi al servizio utilizzando l'apposito bottone di richiesta nella home page del progetto. Fatto ciò si è indirizzati alla pagina di Foursquare dove viene richiesto di effettuare il login al servizio o se non registrati, di creare un account. Come descritto in figura 2.2.



Figura 2.2 Diagramma connessione al sito.

A questo punto abbiamo il userID e il *token* OAuth, che vengono salvati nel nostro server. Questo servirà all'applicazione Real-time per identificare i client che faranno richieste tramite Foursquare.

### 2.4.2 Richiesta di Check-in

Il server aspetta che gli arrivi una richiesta da Foursquare, identifica il client che ha eseguito la richiesta e interagisce con Foursquare. Rispondendo con dei dati che contengono quello che sarà visualizzato dall'utente.

Appena eseguito il *check-in*, l'utente visualizza la notifica dell'applicazione nella pagina del luogo, in questo modo l'utente può utilizzare l'applicazione. La figura 2.3 descrive tutti i passaggi che sono svolti durante il processo.

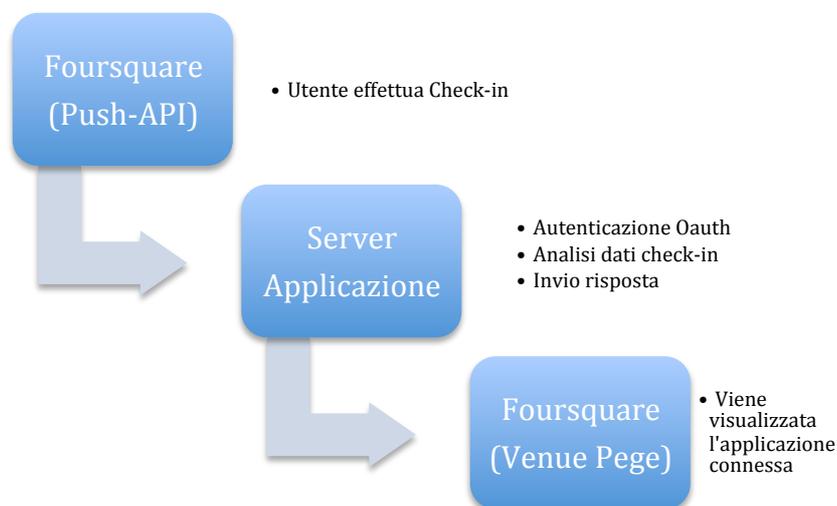
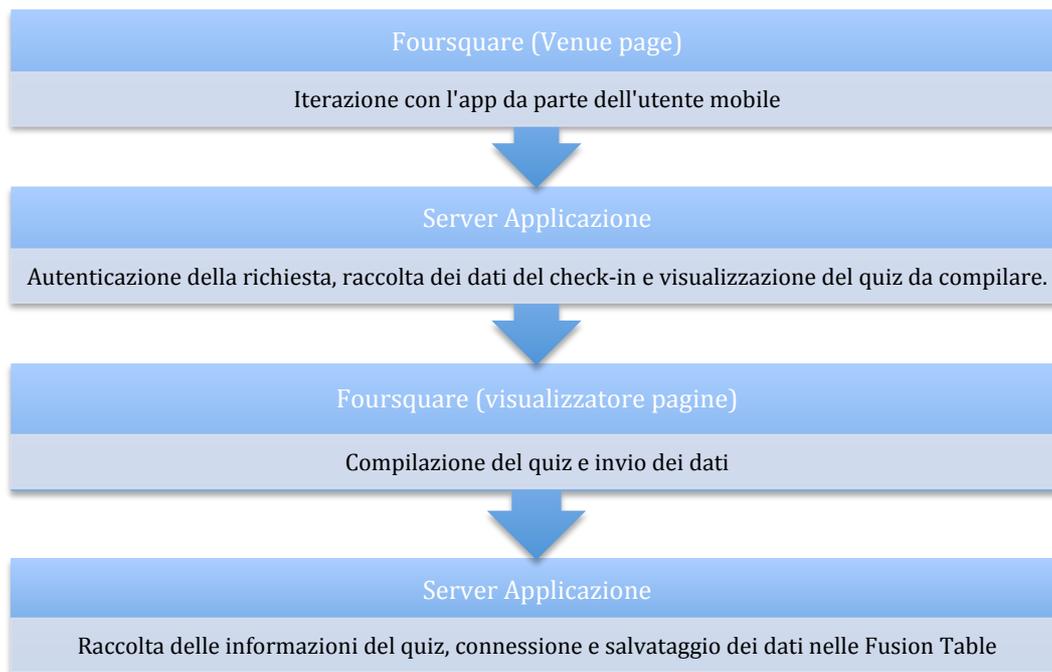


Figura 2.3 Diagramma della richiesta Real-Time

### 2.4.3 Raccolta informazioni

Nel momento in cui un utente interagisce con la richiesta dell'applicazione connessa su Foursquare, è visualizzato il questionario da compilare. Fatto ciò è tutto inviato al server della nostra applicazione in modo tale da poter salvare tutti i dati nelle Fusion Table. Nella figura 2.4 è descritto come ciò avviene a livello concettuale.

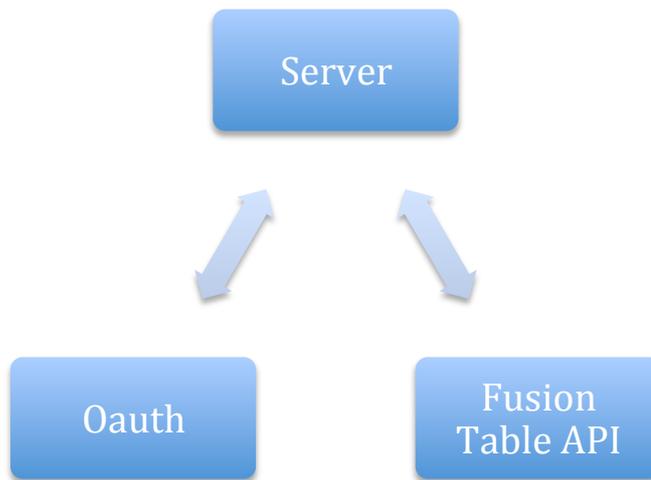


*Figura 2.4 Diagramma della procedura di raccolta informazioni.*

La pagina web contenente il quiz è caricata dal visualizzatore di pagine web integrato in Foursquare. Alla fine di questi processi abbiamo raccolto tutte le informazioni necessarie quindi possiamo procedere con il salvataggio dei dati tramite il servizio Fusion Table.

### 2.4.4 Salvataggio informazioni

In questo momento si hanno tutte le informazioni d'interesse per inserire una segnalazione nel database. Quindi come avvenuto per Foursquare adesso occorre autenticarsi alle Google Fusion Table API in modo da poter compiere operazioni sulla base di dati.



*Figura 2.5 Diagramma funzionamento API Fusion Table.*

Nella parte sinistra della figura 2.5 vediamo come il server interagisce con OAuth per ottenere un *token* di accesso. Appena acquisito il *token* si costruisce la richiesta che viene inoltrata alle Fusion Table API in modo da poter inserire i dati nel database. Le informazioni sono visualizzabili nel web tramite il servizio Google Maps.

Dopo aver elencato tutte le tecnologie utilizzate, vedremo, nel prossimo capitolo, i dettagli implementativi e le scelte strategiche intraprese per sviluppare il progetto. Andremo quindi ad analizzare il perché sono state fatte delle scelte invece che altre. Per quale motivo si è scelto un tipo d'implementazione rispetto ad un altro.

---

# Implementazione

In questo capitolo entriamo nei dettagli implementativi relativi alle scelte tecnologiche che hanno portato allo sviluppo del progetto. Volendo introdurre un quadro generale diciamo che, come prima cosa il progetto si basa sulla scelta e la definizione delle informazioni da inserire nella struttura dati fondamentale per la fase iniziale di data integration.

In secondo luogo si passa alla fusione di diverse basi di dati presenti nel web. Dati che sono salvati in modi molto differenti (come descritto nel capitolo 2), non sempre precisi e spesso ricchi di informazioni non utili per quello che è il contesto applicativo del nostro servizio.

Come terza operazione è stata analizzata l'architettura scelta e in base ad essa sono state scelte le tecnologie da utilizzare per l'implementazione. Nasce da qui la necessità di creare un sito web dove avere la possibilità di accedere, consultare e visualizzare i dati raccolti, fornendo anche la possibilità di autenticarsi a Foursquare permettendo così ad un utente di abbinare l'applicazione al proprio account.

Vedremo poi come l'applicazione creata interagisce con Foursquare per permettere all'utente di compilare un quiz che descriva le eventuali barriere architettoniche presenti nel luogo in cui si trova.

Raccolte tutte le informazioni necessarie in questo capitolo sono descritte in dettaglio le scelte implementative, per rendere il sistema il più efficiente possibile.

## 3.1 Struttura Database

Vediamo com'è stato strutturato il database in modo da renderlo il più possibile fruibile.

In seguito sono elencati i campi utilizzati nel database e fornita la spiegazione della loro scelta. Tutti i campi elencati sono utilizzati nella tabella Fusion Table.

Nome Campo	Descrizione
<b>fsqId</b>	Identificatore del check-in effettuato assegnato da Foursquare
<b>timeZoneOffset</b>	Zona geografica da dove è stato fatto check-in
<b>fsqVenueId</b>	Identificatore univoco del luogo
<b>lastUpdate</b>	Data dell'ultimo aggiornamento del campo
<b>name</b>	Nome del luogo
<b>category</b>	Categoria del luogo (es. bar, ristorante, municipio, teatro ...)
<b>geo</b>	Coordinate geografiche del luogo
<b>postalCode, address, city, state</b>	Anagrafica del luogo (nel database i campi sono separati)
<b>accessLevel</b>	Livello di accessibilità del luogo valutato dagli utenti
<b>comment</b>	Commento inserito dagli utenti
<b>doorways</b>	Indica la possibilità di accesso dalla porta principale
<b>elevator</b>	Indica la presenza o meno di ascensore
<b>escalator</b>	Indica la presenza o meno di scale mobili
<b>parking</b>	Indica la presenza di un parcheggio per disabili
<b>pin</b>	parametro per settare il colore del pin nella mappa a seconda del livello di accessibilità

Fusion Table abbina già un identificatore univoco ad ogni riga della tabella definito come ROWID quindi non è stato necessario inserire un campo per l'identificatore. È stato necessario inserire un campo per identificare i luoghi mappati su Foursquare, in modo da non aver problemi di duplicazione dei dati.

Sono stati scelti sei campi per la descrizione della barriera architettonica in modo da non rendere pesante la compilazione del quiz e il caricamento dei dati.

L'idea è nata dal fatto che un quiz breve e ben fatto è meglio di uno lungo e complesso, che potrebbe annoiare l'utente e quindi portarlo a non compilare il questionario.

Passiamo a come sono stati scelti i dati che sono stati integrati, dove è stato possibile farlo. Infatti, per alcuni dei servizi citati sono stati richiesti i listati dei dati del

database ma senza ricevere risposta alcuna dagli amministratori. Si sono riscontrati anche problemi di validità e verificabilità delle informazioni raccolte.

## 3.2 Integrazione

Utilizzando l'interfaccia on-line del servizio Google Fusion Table e Microsoft Excel per alcune modifiche sui dati, sono stati integrati con successo i dati del servizio di Access Togheter e Liberi di Muoversi. In questo modo il servizio è partito su una base solida di segnalazioni. Vediamo nel dettaglio come i dati sono stati integrati.

Per quanto riguarda Access Togheter non si sono riscontrati particolari problemi visto che viene utilizzato il medesimo servizio Fusion Table. Alcuni problemi si sono verificati nella pulizia dei dati, dato che il servizio gestisce molte caratteristiche dettagliate delle barriere architettoniche, quindi ha un alto numero di campi diversi.

Invece per il servizio Liberi di Muoversi i dati sono in formato .csv (formato di Excel) che ha comportato alcune difficoltà aggiuntive. Prima di tutte sono stati eliminati tutti quei dati che non servivano, cioè che noi non abbiamo preso in considerazione. I rimanenti dati sono stati ottimizzati per essere inseriti.

I campi presi dal database di Liberi di Muoversi sono:

- lastUpdate;
- name;
- geo;
- comment.

Queste sono le informazioni più utili raccolte dopo il processo d'integrazione. Il campo *geo* salva le coordinate nel formato KML quindi è pienamente compatibile con il servizio Google Maps. I commenti sono ben fatti e descrivono a pieno la caratteristica della barriera architettonica. Il problema maggiore è che non si è riuscito a leggere il campo che descrive il livello di accessibilità perché ci sono state delle incompatibilità nel momento in cui sono stati scaricati i dati dal sito del servizio. Questo problema è stato ovviato aggiungendo un collegamento all'interno del commento che permette di aprire direttamente la segnalazione presente nel sito web.

### 3.4 Creazione applicazione

Vediamo quali sono i passaggi seguiti per creare una nuova applicazione su Foursquare. Per prima cosa è stata portata a termine una registrazione al sito web di Foursquare ed in seguito eseguito l'accesso alla sezione per sviluppatori. A questo punto per creare una nuova applicazione è bastato cliccare nel bottone “crea una nuova applicazione”. Fatto ciò, è comparso un *form* nel quale è sufficiente inserire i dati che andranno a descrivere i dettagli della nostra applicazione. Il nome con il quale si vuole chiamare l'applicazione, gli indirizzi web dove la nostra applicazione risiede, la possibilità di attivare o no il servizio *Push API* e infine dei dettagli descrittivi. Inoltre abbiamo dei campi da compilare se per caso la nostra applicazione è creata per *smartphone* o *tablet*.

L'applicazione creata su Foursquare è accessibile tramite dispositivo mobile dal quale verrà aperta la pagina web in cui la nostra applicazione risiede.

Dopo aver creato l'applicazione, avremo delle informazioni tecniche che serviranno per permettere l'autenticazione e la personalizzazione. Come:

- informazioni di carattere amministrativo (client id, client secret, push secret);
- l'indirizzo web dove risiede l'applicazione e il *Redirect URI* l'indirizzo dove le API di Foursquare inviano i dati di risposta;
- informazioni dettagliate sul Push API;
- icone ed immagini.

Per quanto riguarda il Push API Foursquare mette a disposizione una console, con la quale l'utente può verificare il corretto funzionamento della sua applicazione. Dopo aver creato l'applicazione vediamo come viene integrata la funzionalità per richiedere l'autenticazione e l'autorizzazione.

### 3.5 Autenticazione e permessi in Foursquare

Per permettere di autenticare un utente nel sistema e di avere accesso garantito alle funzionalità di Foursquare, l'utente deve registrarsi al servizio oppure, se già in possesso di un account, eseguire il login e accettare le condizioni che gli vengono poste.

Nel sito web è inserita una sezione dove l'utente può iniziare la procedura di autenticazione tramite il click di un apposito pulsante. Questo pulsante è messo a disposizione tra gli strumenti per il social media nella pagina delle risorse di Foursquare. Per utilizzare la funzionalità basta integrare il codice nella propria pagina web avendo l'accortezza di inserire il *client id* dell'applicazione creata.

Dopo aver eseguito il login o portato a termine la registrazione, Foursquare chiede di dare autorizzare all'applicazione per accedere all'account, leggere i dati e le informazioni.

La richiesta è effettuata concatenando il *client id* e il *redirect URI* all'indirizzo dove risiede il sistema di rilascio dei token OAuth. Nell'immagine seguente vediamo cosa accade quando si clicca nel pulsante.



Figura 3.1 Finestra di autorizzazione per applicazioni su Foursquare.

Accettando la richiesta (figura 3.1) diamo il permesso all'applicazione di interrogare i nostri dati presenti su Foursquare, quindi dati anagrafici, *check-in*, punteggi e quant'altro sia presente nell'account.

In questa fase è creata anche una chiave, che sarà successivamente utilizzata per generare il *token* OAuth.

Dopo il primo passaggio di autenticazione è richiamata una pagina apposita, dove viene generato e salvato il *token*. Questa pagina riceve in ingresso una *HTTP request* da Foursquare che contiene la chiave. La chiave viene utilizzata per una nuova richiesta al server OAuth. Vediamo man mano i passaggi effettuati.

Come prima cosa sono stati impostati *client id*, *client secret* e *redirect URI* con i valori che sono presenti nel sito degli sviluppatori Foursquare. È stata letta la chiave in entrata tramite la funzione `$GET [ ]`. Poi è stata costruita la nuova richiesta da inoltrare concatenando i valori appena elencati al seguente indirizzo web.

```
https://fousquare.com/oauth2/access_token? ...
```

In questo modo, se tutto è andato a buon fine, abbiamo nella risposta il nostro *token* OAuth. Tramite il *token* possiamo utilizzare le API di Foursquare per poter effettuare qualsiasi operazione necessaria.

Vediamo come interrogare le API per ottenere le informazioni di un utente.

```
https://api.foursquare.com/v2/users/self?oauth_token="..."
```

Eseguendo una richiesta a questo indirizzo abbiamo come risposta un JSON che contiene tutte le informazioni sull'utente che ha fatto la richiesta e anche la lista dettagliata degli amici.

Queste informazioni servono per catturare l'id di un utente in modo da poterlo abbinare al suo *token* OAuth e salvarlo in file XML. User id e token sono utilizzati da altre parti dell'applicazione, dove c'è bisogno di interrogare le API di Foursquare, per questo motivo sono salvati in un file appositamente creato.

Il file XML è stato strutturato nel seguente modo.

```
<?xml version="1.0"?>
<users>
  <user>
    <id> userID </id>
    <token> access_token </token>
  </user>
  ...
</users>
```

Il file è salvato nel server in modo da poter essere riutilizzato in seguito dalle altri componenti del sistema.

## 3.5 Applicazione connessa

Vediamo come s'interagisce con il server API di Foursquare in modo da inoltrare delle notifiche nel momento in cui l'utente fa un *check-in*. Come vedremo in seguito l'applicazione consiste in un file residente nel server che attende un oggetto di tipo request contenente i dati relativi a un check-in effettuato dall'utente. All'interno dell'oggetto troviamo anche delle informazioni sull'utente che saranno utilizzate per autenticare la risposta. Infine le informazioni sono inviate dal server di Foursquare.

### 3.5.1 Utilizzo Server Push

Tramite il protocollo HTTP-Push, dopo che un utente ha eseguito il check-in e quindi inviato una richiesta al server di Foursquare, è trasmessa una richiesta push alla nostra applicazione inviando una richiesta HTTP POST all'URL fornito. Che contiene due parametri.

Il primo parametro in formato JSON che contiene i seguenti dettagli riguardanti:

- check-in (id, data creazione, time zone);
- utente (id, nome, cognome, foto, sesso, città residenza);
- luogo (indirizzo, latitudine, longitudine, dettagli anagrafici);
- categoria (id, nome, icona);
- statistiche (numero check-in, numero utenti, numero commenti).

Il secondo parametro contiene il *push secret* che è possibile trovare e ripristinare in caso vogliamo dei dettagli sul consumatore. Questo segreto è aggiunto in modo da poter verificare l'autenticità delle richieste ricevute (cioè che sono arrivate dai server di Foursquare invece che da terzi).

Queste informazioni sono raccolte nella nostra applicazione lato server, che le utilizza per formare una nuova richiesta da inviare. L'applicazione è scritta in PHP e utilizza le funzionalità messe a disposizione dal linguaggio, che è molto consolidato per lo sviluppo web delle applicazioni.

Per utilizzare le Push API, il nostro sistema deve risiedere in un dominio criptato (HTTPS), in modo da rendere sicuro lo scambio d'informazioni. È creato un canale di comunicazione criptato tra il client e il server attraverso uno scambio di certificati; una volta stabilito questo canale al suo interno viene utilizzato il protocollo HTTP per la

comunicazione. Questo tipo di comunicazione garantisce che solamente il client e il server siano in grado di conoscere il contenuto della comunicazione.

### 3.5.2 Applicazione Real-Time

In questo paragrafo vediamo in che modo è stata sviluppata la parte del sistema che interagisce con il server push di Foursquare.

Entriamo nei dettagli implementativi, come prima cosa vediamo come leggere la richiesta che ci viene inviata.

```
$obj = json_decode($_REQUEST['checkin']);
```

Con questa riga di codice sono raccolte le informazioni in formato JSON, per poter accedere e salvare le informazioni si utilizza la seguente sintassi.

```
$checkinid = $obj->id;...
```

Nel codice riportato sopra vediamo come viene estrapolato l'id di un check-in eseguito. In questo modo salviamo tutte le informazioni che ci interessano.

Per inviare la richiesta al server di Foursquare è necessario autenticarsi. È stato creato un file XML che contiene *userId* e *token* OAuth di ogni utente che ha dato il permesso all'applicazione di accedere ai propri dati. Quindi dobbiamo caricare e accedere al file in modo da poter ricercare, tramite il valore *userId* di utente, il *token* OAuth ad esso abbinato. Entriamo adesso nella parte dove viene costruita la richiesta.

I parametri che servono per costruire l'URL sono:

- *url*, in questo campo viene messo l'indirizzo della pagina web dove vogliamo che la nostra applicazione si colleghi;
- *text*, campo testuale che viene associato all'applicazione nel momento che appare all'utente;
- *OAuth token*, token per l'autenticazione.

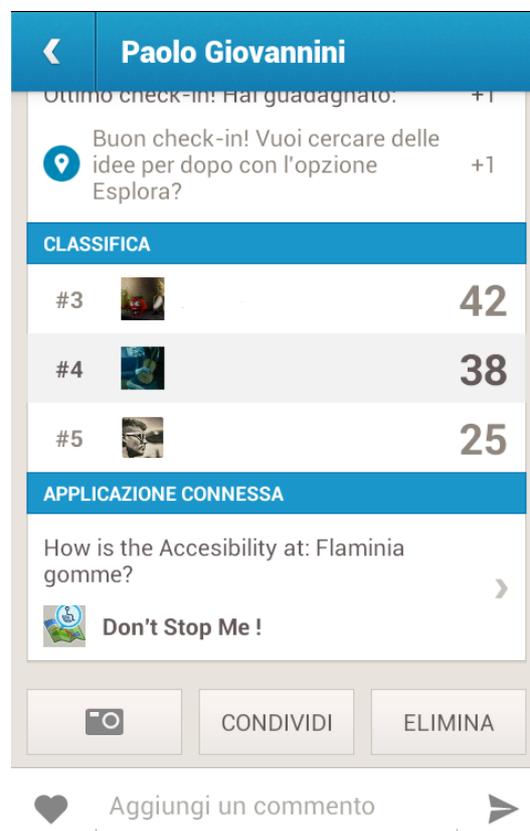
In questo modo è tutto pronto per essere inviato a Foursquare così da far visualizzare all'utente la richiesta della nostra applicazione. Questa richiesta è inviata tramite metodo HTTP POST attraverso il seguente codice.

```

$ch = curl_init();
    curl_setopt($ch,CURLOPT_URL, $replyURL);
    curl_setopt($ch,CURLOPT_POST, count($fields));
        curl_setopt($ch,CURLOPT_POSTFIELDS,
$fields_string);          $data = curl_exec($ch);
curl_close($ch);

```

All'interno della variabile `$replyURL` c'è un array contenente l'URL, come descritto in precedenza. Vediamo con un'immagine cosa accade all'utente quando fa un check-in e interagisce con la nostra applicazione.



**Figura 3.2** Applicazione connessa su schermata Android

Nella figura 3.2 possiamo notare una foto della schermata dell'applicazione Foursquare su sistema operativo Android. Vediamo come dopo un *check-in* Foursquare visualizzi una pagina contenente: dettagli sul luogo, punteggi di badge e classifica utenti.

Nel riquadro *applicazione connessa* si vede come l'applicazione *Don't Stop Me!* pone una domanda all'utente. La domanda chiede “com'è l'accessibilità nel luogo...?”,

il luogo è dove l'utente si è appena registrato. Nel momento che è cliccata la zona di schermo creata dall'applicazione, è caricato un quiz che contiene delle domande inerenti all'accessibilità, l'utente può compilare tale questionario e contribuire così alla raccolta d'informazioni.

### 3.6 Questionario

In questo paragrafo è descritto com'è stato pensato e costruito il questionario tramite il quale sono raccolte le informazioni inserite da un utente. La procedura funziona nel seguente modo.

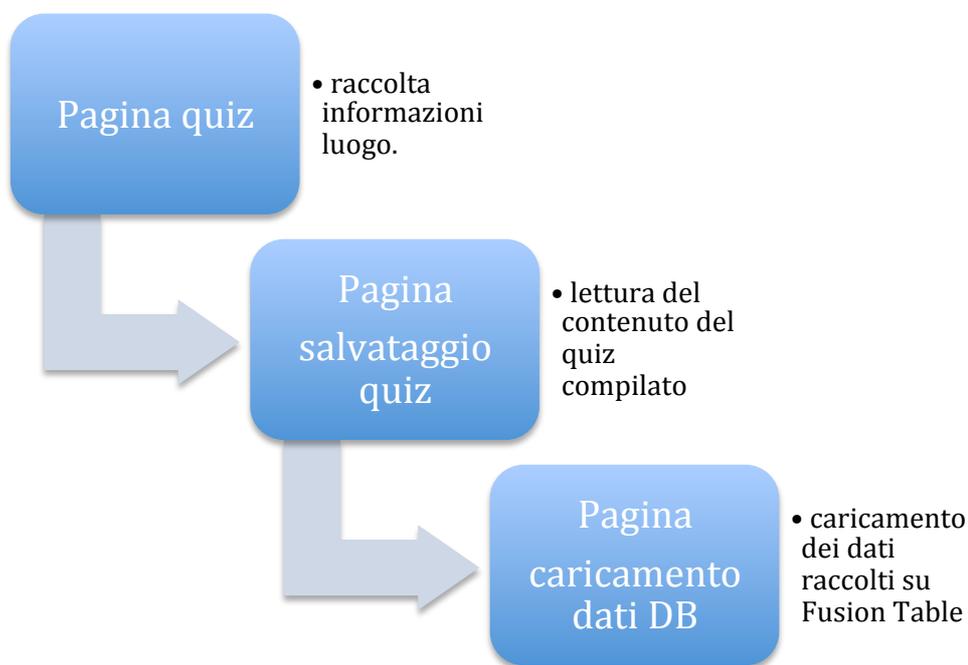


Figura 3.3 Diagramma procedurale di creazione del questionario.

In Figura 3.3 vediamo com'è stata pensata l'architettura per implementare il questionario:

- pagina quiz, in questa pagina viene visualizzato il quiz che l'utente potrà compilare scegliendo tra diverse risposte. Sono inoltre raccolte le informazioni del luogo in cui è stato fatto il *check-in* e memorizzate in un oggetto creato con un'apposita classe;
- pagina salvataggio quiz, in questa pagina vengono raccolte le informazioni sul quiz compilato dall'utente, il commento inserito viene poi aggiunto alla pagina Foursquare del luogo nella sezione apposita *Consigli*;

- pagina caricamento DB, in questa pagina vengono salvate tutte le informazioni raccolte nella tabella Fusion Table.

### 3.6.1 Raccolta dei dati sul check-in

Come prima cosa si deve fare una richiesta di autenticazione al server di Foursquare, in questo momento viene richiesto il token OAuth. Tramite le API Foursquare prendiamo il token OAuth come descritto nel paragrafo 3.5.

Dopo aver ottenuto l'accesso vengono salvati i dati relativi all'utente: userId, nome, cognome che saranno utilizzati in seguito.

Il token e i dati utente vengono salvati nel file XML che contiene i dati di accesso per gli utenti. Tramite le API Foursquare raccogliamo le informazioni sui check-in effettuati dall'utente autenticato tramite richiesta al seguente indirizzo, concatenando il token.

```
https://api.foursquare.com/v2/users/self/checkins?oauth_token="..."
```

La richiesta restituisce in formato JSON tutti i dati dettagliati sui check-in che sono stati fatti dall'utente. L'applicazione raccoglie i dati riguardanti l'ultimo check-in effettuato e li salva in una classe creata appositamente per mantenere i dati salvati in memoria. La classe contiene tutti i campi descritti nel paragrafo 3.2. In questa fase sono salvati:

- l'identificatore del check-in;
- il valore della time zone;
- la data, in formato "Y-m-d H:i:s", attuale in modo da gestire l'aggiornamento e/o i duplicati dei dati inseriti;
- la categoria del luogo di interesse;
- le coordinate geografiche;
- anagrafica dettagliata del luogo.

L'oggetto creato è salvato nella sessione in modo da poter essere utilizzato e modificato dalle altre pagine. Per gestire la sessione è utilizzato un metodo presente nella libreria del linguaggio PHP.

```
$_SESSION['venueData'] = $venueData;
```

Dove `$venueData` è l'oggetto che contiene le informazioni che sono state salvate. Vediamo in seguito com'è stato composto e implementato il questionario.

### 3.6.2 Domande

Le domande presenti nel questionario sono state scelte dopo un'attenta analisi del problema che vogliamo risolvere, rispettando dei parametri da noi imposti.

Il questionario deve avere le seguenti proprietà:

- essere di facile e immediata visibilità, in modo da facilitare la lettura;
- semplice in modo da non stancare l'utente sottoponendolo a troppe domande;
- completo deve cogliere con poche domande tutte le proprietà che ci interessa descrivere, riguardanti il concetto di accessibilità.

In certe situazioni è necessario raccogliere più informazioni. Questo viene fatto dando la possibilità all'utente di aggiungere un parere personale e ulteriori informazioni compilando una sezione apposita, dove può inserire un commento testuale. Nella figura 3.4 vediamo un esempio di come il questionario è posto all'utente dall'applicazione di Foursquare sotto piattaforma Android, andremo poi in dettaglio a commentare le domande poste.

Figura 3.4 Visualizzazione del quiz su schermata Android.

In figura 3.4 vediamo come in cima alla schermata troviamo il nome dell'applicazione al momento utilizzata. Poi in evidenza il nome del luogo in cui l'utente ha eseguito il check-in e a seguire il questionario.

La pagina web contenente il questionario è scritta in codice HTML5[HTML13] utilizzato una struttura dati apposita denominata *form* [HFI13]. Un *form* HTML è utilizzato per il passaggio d'informazioni ad un server. Contiene degli elementi d'input come: campi di testo, *checkbox*, *radio-button* e *submit button* .

L'aspetto stilistico e grafico è curato utilizzando CSS3[CSS13] come di solito avviene per l'implementazione di una pagina web. Nell'implementazione della grafica si è posta attenzione sul fatto che la pagina deve essere visualizzabile soprattutto da smartphone e tablet.

Tramite l'interfaccia mobile è possibile selezionare una delle risposte a scelta multipla, le possibili risposte sono visualizzate tramite *radio-button* . Nella parte finale troviamo il pulsante di conferma, utilizzato per inviare i dati del questionario appena compilato. È stato scelto di scrivere il questionario in lingua inglese in modo da poter raccogliere più utenti possibili.

Le domande presenti nel questionario sono le seguenti:

1. *il luogo è accessibile?* A questa domanda è possibile rispondere: Accessibile, Parzialmente Accessibile, Non Accessibile;
2. *l'entrata principale è accessibili?* Si può rispondere SI oppure NO;

3. *ci sono scale mobili?* Si può rispondere SI oppure NO;
4. *sono presenti ascensori?* Si può rispondere SI oppure NO;
5. *è disponibile un parcheggio per disabili?* Si può rispondere SI oppure NO;
6. nella sesta domanda è presente una casella di testo dove è possibile inserire un commento di 220 caratteri.

Dopo che l'utente clicca sul pulsante di conferma, l'applicazione raccoglie tutti i dati e li inoltra alla pagina di salvataggio del quiz.

### 3.6.3 Salvataggio questionario

I dati raccolti sono inviati alla pagina di salvataggio del quiz mediante il form HTTP. Il form invia i dati tramite metodo POST, questi dati sono raccolti nella pagina ricevente utilizzando un'apposita funzione PHP.

```
$_POST['nome del campo nel form'];
```

Questo metodo va applicato per ogni campo presente nel form della pagina dove è presente il questionario. In questo modo abbiamo raccolto i dati del questionario compilato.

Il prossimo passo è quello di controllare il livello di accessibilità scelto dall'utente. Secondo il quale è impostato un valore, che sarà utilizzato per visualizzare il livello di accessibilità dei luoghi sulla mappa:

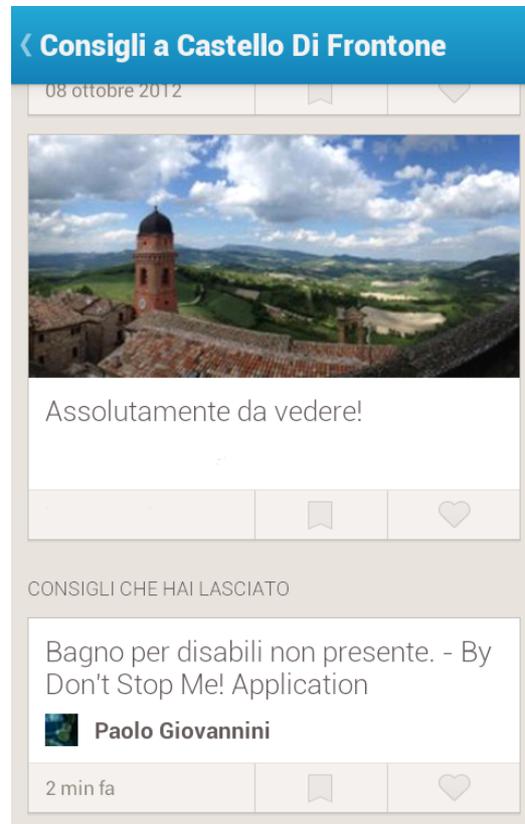
- accessibile: il valore viene settato come *blue*;
- parzialmente accessibile: il valore viene settato come *yellow*;
- non accessibile: il valore viene impostato *red*.

In seguito vediamo come inserire un *consiglio* su Foursquare contente il commento redatto dall'utente.

### 3.6.4 Inserimento Tip

Nella terminologia Foursquare un *Tip* è un consiglio, che un utente lascia dopo aver fatto check-in in un luogo. Per evitare di dover far fare all'utente passaggi aggiuntivi, il sistema inserisce in automatico il Tip utilizzando le API di Foursquare.

Il messaggio contenuto nel Tip contiene il testo del commento che l'utente ha lasciato durante la compilazione del quiz, se l'utente non ha lasciato nessun commento, il Tip viene inserito in automatico in base al livello di accessibilità scelto.



*Figura 3.5 Esempio di un Tip inserito in modo automatico dall'applicazione.*

Vediamo nella Figura 3.5 un esempio di come l'applicazione ha inserito in automatico il Tip in base al commento aggiunto da un utente.

A questo punto sono state raccolte tutte le informazioni necessarie per poter salvare i dati nelle Fusion Table. L'oggetto che contiene tutte le informazioni viene di nuovo salvato nella sessione così da poter essere utilizzato nella apposita pagina.

### 3.7 Scrittura su Fusion Table

In questo paragrafo vedremo come sono memorizzati i dati raccolti. Utilizzando il servizio Google Fusion Table. Come prima cosa si deve accedere alla Google API Consolle [GAC13], è una consolle che gestisce tutti i permessi che sono utilizzati per le API di Google.

Si crea un nuovo progetto, tramite la procedura guidata e s’inseriscono tutti i dettagli del caso. Al termine di questa procedura, viene rilasciata una API key che sarà utilizzata all’interno del progetto per l’autenticazione ai server di Google.

Per creare i permessi necessari all’applicazione si crea un nuovo “client ID” che conterrà tutti i dati utili per accedere ai servizi come descritto in figura 3.6.

#### Client ID for web applications

Client ID:	517233656841-2vpqc8nvsnv6lcv6kdpjgt6q8lv7is7d.apps.googleusercontent.com
Email address:	517233656841-2vpqc8nvsnv6lcv6kdpjgt6q8lv7is7d@developer.gserviceaccount.com
Client secret:	hxXTgLllyTssGWPpqi3PkQ
Redirect URIs:	https://dontstopme.ssl.altervista.org/oauth2callback
JavaScript origins:	https://dontstopme.ssl.altervista.org

*Figura 3.6 Dettaglio creazione di un Client ID.*

Adesso si devono attivare i servizi che Google mette a disposizione, selezionandoli in un’area apposita della consolle. Vediamo nella figura 3.7 i servizi che sono stati abilitati nel nostro caso.

Service	Status	Notes
 Fusion Tables API 	<input checked="" type="checkbox"/> ON	Courtesy limit: 25,000 requests/day
 Google Maps API v3 	<input checked="" type="checkbox"/> ON	Courtesy limit: 25,000 requests/day • <a href="#">Pricing</a>
 Latitude API 	<input checked="" type="checkbox"/> ON	Courtesy limit: 1,000,000 requests/day
 Places API 	<input checked="" type="checkbox"/> ON	Courtesy limit: 1,000 requests/day • <a href="#">Pricing</a>

*Figura 3.7 Servizi API attivati nella consolle Google.*

Fatto ciò, c’è tutto quello che serve per lavorare e interagire con i servizi Google. Possiamo interrogare le Fusion Table utilizzando query SQL così da poter modificare il database della nostra applicazione.

### 3.7.1 Autenticazione

La fase di autenticazione consiste nell’inviare una richiesta a un apposito indirizzo per ottenere un token OAuth con il quale si possono interrogare le API. L’indirizzo è il seguente.

```
https://www.googleapis.com/auth/fusiontables
```

È costruito l'URL che sarà poi inviato al server Google, utilizzando anche il client ID e la API key.

Per quanto riguarda l'implementazione si è scelto di utilizzare il linguaggio di *scripting* JavaScript per la sua compatibilità e il supporto della comunità degli utenti web. Si è scelto di utilizzare le *Google APIs Client Library* [GACL13] per gestire l'autenticazione. Riguardo l'accesso ai dati privati, la nostra applicazione ha bisogno di eseguire autenticazione e avere autorizzazioni da Google. Sono definiti due livelli di sicurezza *simple* (utilizza solamente la API key) o *Authorized* (utilizza sempre la API key ma in più anche i token OAuth 2.0).

Per accedere alle informazioni personali la nostra applicazione deve utilizzare il token OAuth, come abbiamo detto OAuth è un meccanismo che permette di far autenticare utenti e applicazioni. La libreria Javascript effettua tutte le complesse operazioni per ottenere il token, è solo necessario avere in client ID che si riceve quando si registra l'applicazione e sapere quali dati andremo ad utilizzare.

La libreria procede all'autenticazione in questo modo:

1. l'utente clicca nel link login;
2. il browser visualizza un finestra dove chiede all'utente di approvare il permesso, per l'applicazione, di accedere alle sue informazioni;
3. dopo l'approvazione, il browser rimanda l'utente dalla pagina da cui proveniva;
4. il *callback* salva il token di autorizzazione e chiude la finestra.

Fatto ciò l'applicazione è autorizzata ad accedere ai dati.

Vediamo com'è implementato a livello codice:

```
function handleClientLoad() {
    gapi.client.setApiKey(apiKey);
    window.setTimeout(checkAuth, 1);
}

function checkAuth() {
```

```
    gapi.auth.authorize({client_id: clientId, scope:
scopes, immediate: true}, handleAuthResult);
}

function handleAuthResult(authResult) {
    var authorizeButton =
document.getElementById('authorize-button');
    if (authResult && !authResult.error) {
        authorizeButton.style.visibility = 'hidden';
        makeApiCall();
    } else {
        authorizeButton.style.visibility = '';
        authorizeButton.onclick = handleAuthClick;
    }
}

function handleAuthClick(event) {
    gapi.auth.authorize({client_id: clientId, scope:
scopes, immediate: false}, handleAuthResult);
    return false;
}
```

La funzione `handleClientLoad` viene chiamata durante il caricamento. Il metodo setta la API key e chiama la funzione `checkAuth`. Quando un utente effettua il login viene richiamata una funzione chiamata `handleAuthClick` che chiama il metodo di autorizzazione presente nella libreria, la risposta viene gestita nel metodo `handleAuthResult`.

Se l'autorizzazione ha avuto successo, il server ritorna il token OAuth alla funzione `authorize`, gli oggetti ritornati sono passati alla funzione `handleAuthResult` che controlla se il token è valido.

Vediamo ora come sono utilizzate le API Fusion Table.

### 3.7.2 Operazioni su Fusion Table

A questo punto l'applicazione ha tutte le informazioni su un luogo, pronte per essere inserite. Vediamo le strategie implementative utilizzate:

- come prima cosa viene effettuata una ricerca nel database per vedere se la tupla che stiamo per aggiungere sia già presente o meno;
- se è presente faremo un aggiornamento dei dati;
- se non è presente viene inserita una nuova tupla;
- avvio della procedura di salvataggio nella Fusion Table.

La sintassi utilizzata è una query SQL concatenata in un URL. Vediamo in seguito com'è costruito.

#### 3.7.2.1 Selezione

La selezione di una tupla avviene cercandola tramite il suo Foursquare ID, quindi avremo una cosa simile a:

```
SELECT ROWID FROM `table id` WHERE `condizione`
```

Dove ROWID è l'identificatore di default della tupla, e *table id* è l'id univoco associato alla tabella. In questo modo possiamo sapere se il luogo è già stato segnalato da altri utenti.

#### 3.7.2.2 Aggiornamento

Se il luogo è stato segnalato, ci limitiamo ad aggiornare i dati, se non erano presenti sono aggiunti e il resto viene sovrascritto.

Vediamo in seguito come.

```
UPDATE `table id`  
SET `nome campo`, `valore`  
WHERE ROWID = `rowid della tupla da aggiornare`
```

In questo modo viene aggiornata la tupla.

### **3.7.2.3 Inserimento**

Se la tupla non è presente nella Fusion Table allora ne viene inserita una nuova, vediamo in seguito come.

```
INSERT INTO `table id`  
  `tutti i nomi dei campi`  
VALUES `tutti i valori da inserire`
```

### **3.7.2.4 Invio a Fusion Table**

Arriviamo dove vengono inviate le informazioni al server. Tutto ciò è fatto costruendo un URL dove viene concatenata la query SQL a seconda del tipo di richiesta che vogliamo fare. La risposta arriva in formato JSON così è molto intuitivo raccogliere i risultati delle operazioni svolte.

## **3.8 Visualizzazione dei dati**

Per la visualizzazione dei dati è stato implementato un sito web, in cui è descritto in cosa consiste il progetto. È richiesto, agli utenti che vogliono partecipare alla raccolta d'informazioni, di creare un account su Foursquare in modo da poter interagire con l'applicazione. Inoltre tramite l'utilizzo delle mappe Google sono visualizzate tutte le segnalazioni effettuate dagli utenti.

### **3.8.1 Google Maps**

Google Maps permette di visualizzare tutte le segnalazioni che sono state compiute dagli utenti, consentendo modifiche sia stilistiche sia di presentazione dei dati.

Vediamo in seguito i dettagli implementativi e le scelte intraprese.

#### **3.8.1.1 Stili e proprietà**

È possibile personalizzare la presentazione delle mappe standard, modificando come sono visualizzati elementi tipo: strade, parchi e aree edificate.

Nella sintassi ci sono due tipi di stili:

- “Map features”: è possibile selezionare una degli elementi che caratterizzano una mappa, così da evidenziarlo rispetto agli altri;
- “Stylers” in questo modo può modificare tutte le caratteristiche visive della mappa come il colore, la saturazione, la luminosità, il tipo di visibilità e la grandezza in pixel.

Per il settaggio degli stili è utilizzato un array simile al seguente.

```
var styleArray = [  
  {  
    featureType: "all",  
    stylers: [  
      { saturation: -80 }  
    ]  
  }, {  
    featureType: "road.arterial",  
    elementType: "geometry",  
    stylers: [  
      { hue: "#00ffee" },  
      { saturation: 50 }  
    ]  
  }, {  
    featureType: "poi.business",  
    elementType: "labels",  
    stylers: [  
      { visibility: "off" }  
    ]  
  }  
];
```

Dal codice si nota come sono settati tutti parametri descritti in precedenza.

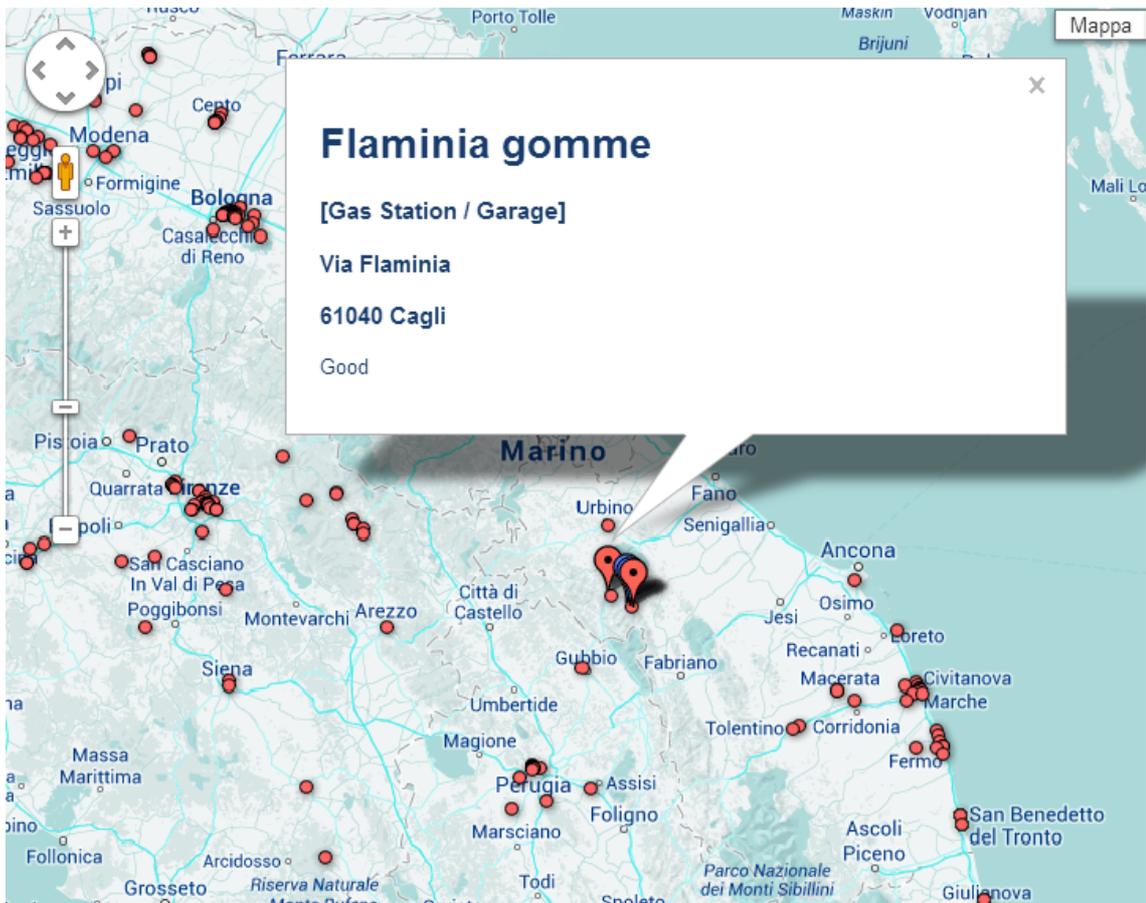
### 3.8.1.2 Caricamento Layer da Fusion Table

Per visualizzare le informazioni salvate nelle Fusion Table, è utilizzato il concetto di “layer”, cioè strato d’informazioni caricato sopra la mappa. Lo strato visualizza tutti i punti che rappresentano un luogo segnalato dagli utenti, ma contiene anche le informazioni caricate dalle Fusion Table.

Il codice riportato in seguito descrive il caricamento del layer.

```
layer = new google.maps.FusionTablesLayer({
  query: {
    select: '\Geocodable address\'',
    from: 'table id'
  }
});
layer.setMap(map);
```

Nella figura 3.8 possiamo vedere il risultato dell’applicazione degli stili alla mappa e del layer Fusion Table. Nel dettaglio possiamo notare come nel riquadro a fumetto siano visualizzate le informazioni dettagliate riguardanti un luogo. Per visualizzare ciò è bastato integrare il codice JavaScript che crea la mappa, nella pagina web tramite l’ausilio del linguaggio HTML.



*Figura 3.8 Dati visualizzati nella mappa.*

La mappa è centrata automaticamente tramite codice, richiedendo le coordinate al browser web. Se questo non è possibile, è stata definita una località di default. In questo modo l'utente può consultare le segnalazioni più vicine.

Con questo si conclude il terzo capitolo dove sono stati descritti tutti i passaggi implementativi compiuti. Vedremo nel prossimo capitolo le considerazioni finali e i possibili sviluppi futuri del progetto.



---

# Conclusioni

Il progetto sviluppato vuole cercare di sensibilizzare il cittadino al problema delle barriere architettoniche, cercando di renderlo partecipe alla raccolta d'informazioni sugli impedimenti presenti nei luoghi da lui visitati.

La barriera architettonica può essere una scala, un gradino, una rampa troppo ripida, qualunque elemento architettonico può trasformarsi in un impedimento. Lo scopo è poter utilizzare le novità in ambito tecnologico, come smartphone molto potenti, sensori e visori artificiali e metterli al servizio dei cittadini in modo da renderli partecipi e sensibilizzarli allo sviluppo del progetto.

Sono già presenti dei servizi web e mobile, che consentono la raccolta d'informazioni sulle barriere architettoniche. I servizi sono stati confrontati e con un'attenta analisi e integrazione dei dati si è riusciti a cogliere informazioni da due servizi molto attivi, di cui uno con molte segnalazioni in Italia e un altro con molte segnalazioni in Nord America. Comunque le segnalazioni raccolte sono un buon numero, per essere un punto di partenza. In questo modo il progetto ha già una base solida d'informazioni.

Per raccogliere dati sui luoghi visitati dagli utenti, si è scelto di utilizzare Foursquare. Questo ha consentito da una parte di avvantaggiarsi di un bel numero di "rilevatori" potenziali, dall'altra di diffondere il progetto tra le persone e sensibilizzare maggiormente il tema. Foursquare ha delle API molto ben progettate e funzionali che non hanno permesso una veloce fase di sviluppo e testing.

Per quanto riguarda la memorizzazione delle informazioni si è scelto di utilizzare Google Fusion Tables: un innovativo servizio di Google che consente la creazione e la gestione di grandi quantità di dati on-line. Esso non si configura però, come un database tradizionale in quanto si concentra sulla condivisione e sulla collaborazione. L'obiettivo concreto di Fusion Tables consiste nel permettere a più utenti di lavorare via web sullo stesso database, rimanendo aggiornati in tempo reale sulle modifiche apportate da altri utenti. I dati di più database, inoltre, possono essere correlati. Fusion Tables si è dimostrato un valido e utile strumento, con delle API molto

ben documentate. Anche se il servizio è in fase sperimentale, non ci sono stati molti problemi in fase di sviluppo.

Tutte le informazioni raccolte sono presentate geograficamente tramite Google Maps. La mappa con le segnalazioni è disponibile in un'apposita pagina web che è stata implementata per diverse ragioni: permettere a un utente di collegare l'applicazione al suo account Foursquare; avere una descrizione dettagliata del progetto e rendere disponibili le segnalazioni raccolte alla comunità.

I possibili sviluppi futuri sono diversi. Per ciò che riguarda la fase di data integration, si potrebbero impiegare risorse per raccogliere le informazioni dagli altri servizi, cercando anche di implementare dei sistemi automatici e periodici che raccolgono e aggiornano le informazioni prese dagli altri servizi. Inoltre si potrebbe ampliare il servizio di raccolta dei dati utilizzando altri servizi come Google Places, Gowalla e Facebook Places.

Un altro sviluppo molto interessante è legato alla creazione di un'applicazione mobile in modo da poter personalizzare la raccolta d'informazioni e avere un supporto con cui l'utente può visualizzare le barriere architettoniche nelle immediate vicinanze.

Si auspica che il progetto sviluppato possa sensibilizzare qualsiasi cittadino a prestare il suo contributo nell'obiettivo di ridurre, il più possibile, le barriere architettoniche.

---

# Bibliografia

- [BA06] F. Vescovo, “Barriere Architettoniche”, 2006, [http://www.treccani.it/enciclopedia/barrierearchitettoniche\\_\(Enciclopedia\\_Italiana\)/](http://www.treccani.it/enciclopedia/barrierearchitettoniche_(Enciclopedia_Italiana)/)
- [BAEP13] Disabili.com Testata giornalistica on-line, “Barriere Architettoniche nell’Edilizia Pubblica”, 2013, <http://www.disabili.com/mobilita-auto/speciali-mobilita-a-auto/barriere-architettoniche-e-disabilita/23936-barriere-architettoniche-normativa-e-leggi>
- [L.13/89] COSSIGA DE MITA Presidente del Consiglio dei Ministri, FERRI Ministro dei lavori pubblici, JERVOLINO RUSSO Ministro per gli affari sociali, “Disposizioni per favorire il superamento e l’eliminazione delle barriere architettoniche negli edifici privati.”, Legge 9 gennaio 1989, n. 13
- [L.104/92] Presidente del Consiglio dei Ministri, “Legge-quadro per l’assistenza, l’integrazione sociale e i diritti delle persone handicappate”, 17 Febbraio 1992
- [DM236/89] Ministro FERRI, “Prescrizioni tecniche necessarie a garantire l’accessibilità, l’adattabilità e la visitabilità degli edifici privati e di edilizia residenziale pubblica sovvenzionata e agevolata, ai fini del superamento e dell’eliminazione delle barriere architettoniche”, Decreto Ministeriale - Ministero dei Lavori Pubblici 14 giugno 1989, n. 236.
- [DPR503] SCALFARO presidente della repubblica, “Regolamento recante norme per l’eliminazione delle barriere architettoniche negli edifici, spazi e servizi pubblici.”, Decreto del Presidente della Repubblica 24 luglio 1996, n. 503.
- [DPR380] C. A. CIAMPI presidente della repubblica, “Testo unico delle disposizioni legislative e regolamentari in materia edilizia”, Decreto del Presidente della Repubblica 6 Giugno 2001, n.380
- [CI13] Wikipedia, “Città Intelligente”, 2013, [http://it.wikipedia.org/wiki/Citt%C3%A0\\_intelligente](http://it.wikipedia.org/wiki/Citt%C3%A0_intelligente)

- [PSC13] C. Froghieri e A. M. Sismondi, “Il Paradigma Smart City”, Edizioni FORUM PA, Aprile 2013
- [PS12] E. de Cristofaro e C. Soriente, “Partecipatory Privacy: Enabling Privacy in Participatory sensing”, Università politecnica di Madrid, 23 Gennaio 2012
- [CP13] Wikipedia, “Cloud Computing”, 2013, [http://it.wikipedia.org/wiki/Cloud\\_computing](http://it.wikipedia.org/wiki/Cloud_computing)
- [QRC13] Wikipedia, “Codice QR”, 2013, [http://it.wikipedia.org/wiki/Codice\\_QR](http://it.wikipedia.org/wiki/Codice_QR)
- [AN13] Android, sistema operativo mobile, 2013, <http://www.android.com/about/>
- [PZ13] Pazienti, servizio online, 2013, <http://www.pazienti.it/about>
- [PNT10] Pinterest, servizio online, 2010, <http://about.pinterest.com/>
- [SHN13] Shinynote, servizio online, 2013, <http://www.shinynote.com/>
- [CRF13] Wikipedia, “Crowdfunding”, 2013, <http://en.wikipedia.org/wiki/Crowdfunding>
- [OPP13] Openpolis, servizio online, <http://www.openpolis.it/chi-siamo/>
- [GIS00] M. Catudella, “GIS”, 2000, [http://www.treccani.it/enciclopedia/gis\\_\(Enciclopedia\\_Italiana\)/](http://www.treccani.it/enciclopedia/gis_(Enciclopedia_Italiana)/)
- [TIDE13] D. Beneventano, “Tecniche per l’integrazione dei dati eterogenei”, Università degli studi di Modena, 2013, <http://www.dbgroup.unimo.it/SIA/IntegrazioneDeiDati/IntroduzioneIntegrazioneDeiDati.pdf>
- [DGC13] Wikipedia, “Digital City”, 2013, [http://en.wikipedia.org/wiki/Digital\\_city](http://en.wikipedia.org/wiki/Digital_city)
- [IOT13] Wikipedia, “Internet Of Things”, 2013, [http://en.wikipedia.org/wiki/Internet\\_of\\_Things](http://en.wikipedia.org/wiki/Internet_of_Things)
- [IEE13] Iee 802 LAN/MAN standard committee, 2013, <http://www.ieee802.org/>
- [QIS13] Wikipedia, “query”, 2013, <http://it.wikipedia.org/wiki/Query>
- [IPP13] Wikipedia, “Internet Protocol”, 2013, [http://en.wikipedia.org/wiki/Internet\\_Protocol](http://en.wikipedia.org/wiki/Internet_Protocol)
- [ACS13] Wikipedia, “Client-server model”, 2013, [http://en.wikipedia.org/wiki/Client%E2%80%93server\\_model](http://en.wikipedia.org/wiki/Client%E2%80%93server_model)
- [BMM13] Microsoft, Bing mappe, 2013, <http://it.bing.com/maps/?mkt=it-it>
- [EAW13] Vodafone, servizio online Easy Way, 2013, <http://www.easyway.vodafone.it/>

- 
- [GEA13] Google, Google Earth API, 2013, <https://developers.google.com/earth/>
- [LDM11] Liberi di muoversi, servizio online, 2011, <http://www.liberidimuoversi.it/map/>
- [WHM13] Wheel Map, servizio online, 2013, <http://wiki.openstreetmap.org/wiki/Wheelmap>
- [INL13] Ingresso Libero, servizio web, 2013, <http://www.ingressolibero.info/Il%20Progetto>
- [OSM13] Open street map, 2013 , <http://www.openstreetmap.it/>
- [IOS13] Apple, IOS, 2013, <https://developer.apple.com/ios7/>
- [GSW13] Google, google street view, 2013, <http://maps.google.it/intl/it/help/maps/streetview/>
- [CPT13] Comuni per tutti, servizio web, 2013, [http://www.comunipertutti.it/it/bolzano\\_per\\_tutti](http://www.comunipertutti.it/it/bolzano_per_tutti)
- [ACT13] Access together, servizio web, <http://www.accesstogether.org/>
- [DUR13] Decoro Urbano, servizio web, 2013, <http://www.decorourbano.org/>
- [FBK13] Facebook, 2013, <http://www.facebook.com>
- [GMAP13] Google, Google Maps, 2013, <https://maps.google.com/maps>.
- [FT13] Google, Servizio di supporto online, 2013 <https://support.google.com/fusiontables/answer/2571232?hl=en>
- [FSQ13] A. Maestro, “Foursquare: cos’è, come funziona, il perché di un successo”, 2013-06-07, <http://www.maestroalberto.it/2010/08/28/foursquare-cose-come-funziona-il-perche-di-un-successo/>
- [API13] Wikipedia, “Application programming interface”, 2013, [http://it.wikipedia.org/wiki/Application\\_programming\\_interface](http://it.wikipedia.org/wiki/Application_programming_interface).
- [OAF12] D. Hardt, Ed. Microsoft, “The Oauth 2.0 Authorization Framework”, Ottobre 2012, <http://oauth.net/>.
- [CSSL13] Kioskea.net, “Crittografia - SSL”, 2013, <http://goo.gl/jBFOi>.
- [IJS11] G. Romanato, “Introduzione a JSON”, 2011, <http://www.html.it/articoli/introduzione-a-json-2/>.
- [JST13] W3schools.com, “Java script Tutorial”, 2013, <http://www.w3schools.com/js/>
- [PHP13] Sito ufficiale PHP, “che cos’è il PHP”, 2013, <http://php.net/manual/it/intro-what-is.php>.

- [WGM13] Wikipedia, “Google Maps”, 2013, [http://it.wikipedia.org/wiki/Google\\_Maps](http://it.wikipedia.org/wiki/Google_Maps).
- [GFTA13] Google Developers, “Google fusion table API”, 2013, <https://developers.google.com/fusiontables/>.
- [FSP13] Foursquare Developers site, 2013, <https://developer.foursquare.com/overview/>.
- [TSP13] Wikipedia, “Push tecnologia”, 2013, [http://en.wikipedia.org/wiki/Push\\_technology](http://en.wikipedia.org/wiki/Push_technology).
- [KML13] Google, “KML Documentation”, 2013, <https://developers.google.com/kml/documentation/>.
- [JQR13] JQuery Foundation, 2013, <https://jquery.org/support/>
- [FTLM13] Google Developers, Fusion Table Layer Map, 2013, [https://developers.google.com/fusiontables/docs/samples/multiple\\_layers\\_per\\_map?hl=it](https://developers.google.com/fusiontables/docs/samples/multiple_layers_per_map?hl=it)
- [TWT13] Twitter, social network, 2013, <https://support.twitter.com/>.
- [RFC13] Request for Comments, 2013, <http://www.ietf.org/rfc.html>.
- [JSP13] Linguaggio scripting Javascript, 2013, <http://javascript.it/>.
- [JAVA13] Tecnologia Java, 2013, <https://java.net/>.
- [ANSI13] Wikipedia, “C language”, 2013, [http://en.wikipedia.org/wiki/ANSI\\_C](http://en.wikipedia.org/wiki/ANSI_C).
- [XML13] W3schools, Extensible Markup Language, 2013, <http://www.w3.org/XML/>.
- [LK13] Linked in social network, 2013, [http://it.linkedin.com/static?key=what\\_is\\_linkedin&trk=hb\\_what](http://it.linkedin.com/static?key=what_is_linkedin&trk=hb_what).
- [HTML13] W3schools, Hipertext Markup Language, 2013, <http://www.w3schools.com/html/>.
- [CSS13] W3schools, 2013, <http://www.w3schools.com/css/>
- [GAC13] Google developer, Google API consolle, 2013, <https://code.google.com/apis/console/>.
- [GACL13] Google developer, Google API Client Library, 2013, <https://developers.google.com/discovery/libraries>.