

ALMA MATER STUDIORUM - UNIVERSITA' DI BOLOGNA

CAMPUS DI CESENA

SCUOLA DI SCIENZE

**CORSO DI LAUREA IN
SCIENZE E TECNOLOGIE INFORMATICHE**

Personalizzazione di percorsi accessibili con
GraphHopper

Relazione finale in
Sistemi Multimediali

Relatore:

Prof.ssa Paola Salomoni

Presentata da:

Lisa Cattalani

Correlatore:

Dott.ssa Catia Prandi

Sessione III

Anno Accademico 2013/2014

*L'importante non è la strada scelta,
ma chi si incontra durante il percorso*

(Y.Togashi)

Indice

INDICE	I
INTRODUZIONE	1
1 OPEN STREET MAP	5
1.1 INTRODUZIONE (STORIA).....	5
1.1.1 <i>Definizioni e utilità</i>	7
1.1.2 <i>La community e la raccolta di informazioni</i>	8
1.2 HUMANITARIAN OPENSTREETMAP TEAM.....	9
1.2.1 <i>Mapping Party</i>	10
1.3 LA STRUTTURA DEL DATABASE OPENSTREETMAP	10
1.3.1 <i>JOSM</i>	14
1.4 RENDERING	16
1.4.1 <i>Elenco dei principali Software per il Rendering</i>	17
1.4.2 <i>Mapnik</i>	18
1.4.3 <i>OpenStreetMap 3D</i>	18
1.5 IL ROUTING IN OPENSTREETMAP.....	20
1.5.1 <i>Le tipologie di Routing</i>	20
1.5.2 <i>I profili presenti in OpenStreet Map</i>	20
1.6 OPEN SOURCE E IL SOFTWARE PROPRIETARIO.....	22
2 OSM SOFTWARE E ACCESSIBILITÀ URBANA	25
2.1 LE BARRIERE ARCHITETTONICHE.....	25
2.1.1 <i>Le barriere architettoniche presenti in Openstreetmap</i>	26
2.2 SOFTWARE PER COMPUTER DESKTOP.....	30
2.2.1 <i>Routino</i>	31
2.2.2 <i>OpenRouteService</i>	32
2.3 SOFTWARE PER MOBILE.....	35
2.3.1 <i>Wheelmap – iPhone/Android</i>	36
2.3.2 <i>OpenTripPlanners</i>	38
2.3.3 <i>Altri mobile software</i>	39
3 IL PROFILO WHEELCHAIR IN GRAPHHOPPER	41

3.1	GRAPHHOPPER	41
3.1.1	<i>GraphHopper versione desktop</i>	42
3.1.2	<i>Come è strutturato</i>	44
3.1.3	<i>Richieste in locale o al server</i>	46
3.2	I TAG PER IL PROFILO WHEELCHAIR	47
3.2.1	<i>Come devono essere etichettati i principali servizi</i>	48
3.3	LE MODIFICHE APPORTATE A GRAPHHOPPER	51
3.3.1	<i>Come creare il profilo wheelchair</i>	51
3.3.2	<i>Lettura dati dal file XML</i>	53
	CONCLUSIONI	57
	BIBLIOGRAFIA	I

Introduzione

L'obiettivo di questo documento è quello di descrivere il progetto di tesi che è consistito nella modifica del software GraphHopper, per calcolare un percorso che prendesse in considerazione le barriere architettoniche presenti in mappe OSM(OpenStreetMap), così da fornire percorsi accessibili a tutte quelle persone che hanno delle disabilità motorie permanenti o temporanee.

Questo è un tema fondamentale da investigare per permettere a chiunque di potersi spostare in tutta tranquillità in zone non conosciute indipendentemente dalle necessità e preferenze dell'utente evitando eventuali problemi legati all'accessibilità urbana. Questo è un tema di particolare interesse soprattutto nel contesto Smart City in cui l'obiettivo principale è quello di migliorare la qualità della vita dei cittadini attraverso l'uso delle tecnologie e della partecipazione dei cittadini stessi.

Il calcolo del percorso si basa su mappe che possono essere scaricata gratuitamente dal sito di OpenStreetMap; quest'ultimo, oltre ad essere un sito che permette di scaricare le mappe di tutto il mondo, è anche una community online dove vengono raccolti i dati di tutte le mappe del mondo (luoghi d'interesse, nome delle vie, ecc.) in modo partecipativo e collaborativo sfruttando il crowdsourcing. È infatti possibile scaricare, modificare, aggiornare o aggiungere punti di interesse, strade, città, ecc. di una qualsiasi mappa del mondo. È anche possibile scaricare gratuitamente una grande vastità di software utili per il calcolo di un percorso a seconda delle esigenze (in macchina, a piedi, ecc.) o di un semplice navigatore che sfrutti le mappe di OpenStreetMap. Contrariamente a Google Maps, è possibile modificare liberamente le mappe di tutto il mondo e usare/modificare i software presenti.

Esistono diversi software che permettono di calcolare percorsi sfruttando i dati messi a disposizione da OpenStreetMap. Solitamente questi software di routing non

restituiscono un unico percorso ma sono configurabili a seconda di specifici profili che identificano le proprietà che un percorso deve avere. I profili più comuni sono: pedone; macchina; bici e wheelchair. Quest'ultimo permette di calcolare un percorso che tenga in considerazione la presenza di eventuali barriere o facility architettoniche che potrebbero ostacolare o bloccare persone con mobilità ridotta nel raggiungere una certa destinazione. Solo alcuni tra i software disponibili permettono di gestire il profilo wheelchair, molti dei quali però risultavano incompleti o specifici per nazione. Si è quindi optato per l'implementazione del profilo wheelchair nel software GraphHopper. GraphHopper è un software totalmente gratuito e open source: si può scaricare tutto il codice da repository disponibile su GitHub e installarlo nel proprio computer. GraphHopper è disponibile sia in versione desktop che in versione mobile per Android. È molto veloce ed efficiente nel calcolo di un percorso inoltre è scritto in un linguaggio ad oggetti e di facile comprensione (Java) e completamente modificabile.

Per implementare il profilo wheelchair in GraphHopper si è dovuto principalmente studiare ed analizzare quali elementi (tag), definiti in OSM, devono essere considerati (o meno) nel calcolo di un percorso accessibile, in modo tale da fornire una soluzione alternativa nel caso in cui si incontrino barriere inevitabili che bloccherebbero l'utente con disabilità motorie.

Alcuni esempi di queste barriere sono: (i) cancelli con una difficile apertura: la difficoltà di questo tipo di cancello, sta nella posizione della maniglia che risulta troppo alta e quindi non accessibile a tutti; (ii) cordolo: costituisce una barriera sia per le biciclette che per persone con disabilità motoria, se l'altezza è di 1 centimetro potrebbe risultare accessibile; (iii) passaggi stretti: sono caratterizzata da un accesso molto ristretto che rappresenta una vera e propria barriera architettonica; (iv) percorsi con pavimentazione sdruciolevole, irregolare o sconnessa, e/o con eccessiva estensione; (v) caletta: permette ai pedoni, ma non alle persone con disabilità motorie, di scavalcare un muro o una recinzione.

Una parte del sistema, ancora in versione prototipale, è stata presentata al MAMbo in occasione della presentazione del DISI, suscitando l'interesse dei partecipanti che ci han fatto riflettere sulle modifiche da apportare per migliorare il progetto svolto così da poter arrivare ad una versione stabile.

La tesi è così suddivisa:

- Nel primo capitolo viene introdotta tutta la comunità di OpenStreetMap, i progetti passati e futuri e tutto il materiale gratuito a disposizione, oltre ovviamente ad una libreria online di tutti i termini e software presenti in OpenstreetMap.

- Nel secondo capitolo verranno introdotti i concetti di barriere architettoniche e di accessibilità urbana, con una particolare attenzione a tutti quei software disponibili su OpenStreetMap, che calcolano il percorso per le persone con disabilità.

- Nel terzo capitolo si parlerà del software GraphHopper, delle modifiche apportate a quest'ultimo, per introdurre il calcolo di un percorso per il profilo wheelchair.

Verranno quindi analizzati i vari algoritmi che GraphHopper utilizza, come è strutturato, e le varie fasi che han portato alla modifica di GraphHopper per lo svolgimento del progetto.

1 Open Street Map

In questo capitolo verrà introdotta la community di Open Street Map, come è strutturata, in cosa consiste e come vengono eseguite le operazioni di rendering e routing sulle mappe prodotte e i dati raccolti grazie alla partecipazione volontaria dei contributori di tutto il mondo. Inoltre verranno descritti i vari profili disponibili nei software di routing per calcolare percorsi che tengano conto di particolari caratteristiche.

1.1 Introduzione (Storia)

OpenStreetMap (OSM) venne fondato nel luglio 2004 da Steve Coast. Nell'aprile 2006 OSM iniziò il processo per trasformarsi in una fondazione, nel dicembre dello stesso anno Yahoo! ha concesso ad OpenStreetMap l'utilizzo delle proprie ortofoto aeree come ulteriore base per la produzione di mappe.

Il 4 luglio 2007 l'azienda Automotive Navigation Data ha donato il database stradale completo dei Paesi Bassi e delle arterie principali di India e Cina al progetto, ed a luglio 2007, quando la Fondazione OpenStreetMap ha organizzato la prima conferenza internazionale di OSM (nominata "The State of the Map", Lo stato della Mappa), il progetto aveva 9.000 utenti registrati. Tra gli sponsor dell'evento figurano Google, Yahoo! e Multimap. L'agosto dello stesso anno ha visto il lancio di un progetto indipendente, OpenAerialMap, per mantenere un database di foto aeree disponibile con licenza libera, a cui è seguito in ottobre il completamento dell'importazione dei dati stradali statunitensi dal database TIGER [WOSM2015]. Nel dicembre dello stesso anno l'Università di Oxford è stata la prima importante organizzazione ad usare i dati di OpenStreetMap sul proprio sito web.

Nel gennaio 2008 è stata resa disponibile una funzionalità per scaricare i dati delle mappe nelle unità GPS per ciclisti. In febbraio si sono tenuti una serie di workshop in India. A marzo i due fondatori di OpenStreetMap hanno annunciato di aver ricevuto 2,4 milioni di euro come capitale a rischio per Cloud Made, una società commerciale nata con lo scopo di usare i dati di OpenStreetMap. In agosto 2008, poco dopo la seconda conferenza internazionale The State of the Map, gli utenti registrati sono saliti a 50.000 con oltre 5.000 contributori attivi. Nel marzo 2009, sono stati superati i 100.000 utenti. Il 5 e 6 giugno 2009 si è tenuta a Trento la prima convention italiana di OSM, denominata "OSMit 2009". Il 3 e 4 giugno 2010 è stata effettuata la seconda edizione di OSMit a Genova. La terza si è tenuta il 7 e 8 ottobre 2011 a Legnaro (PD) all'interno del festival delle libertà digitali. Si è deciso, da parte della comunità, di utilizzare un sito come referente per le edizioni a venire [WOSM2015].

OpenStreetMap (OSM) è un progetto collaborativo finalizzato a creare mappe a contenuto libero del mondo. Il progetto punta ad una raccolta mondiale di dati geografici, con scopo principale la creazione di mappe e cartografie [WOSM2015].

La caratteristica fondamentale dei dati geografici presenti in OSM è che possiedono una licenza libera, la Open Database License [WOSM2015]. È cioè possibile utilizzarli liberamente per qualsiasi scopo con il solo vincolo di citare la fonte e usare la stessa licenza per eventuali lavori derivati dai dati di OSM. Tutti possono contribuire arricchendo o correggendo i dati.

Le mappe sono create usando come riferimento i dati registrati da dispositivi mobile dotati di sensore GPS, dispositivi GPS portatili, fotografie di aeree ed altre fonti libere. Sia le immagini renderizzate che i dati vettoriali, oltre che lo stesso database di geodati sono diffusi sotto licenza Open Database License.

OpenStreetMap è stato ispirato da siti come Wikipedia: la pagina in cui la mappa è consultabile espone in evidenza un'etichetta "Modifica" per procedere con la modifica dei dati ed il progetto è accompagnato da un archivio storico delle modifiche (cronologia e log). Gli utenti registrati possono caricare nei database del progetto tracce GPS e modificare i dati vettoriali usando gli editor forniti. Sono infatti gli utenti, con la loro partecipazione volontaria nella raccolta dati tramite crowdsourcing e creazione delle mappe, che hanno permesso la crescita, in termini di dati, e la diffusione del progetto.

Una alternativa proprietaria che riprende le pratiche di crowdsourcing introdotte dalla comunità di OpenStreetMap è Google Map Maker, nel quale i contributi dell'utente possono venir inseriti su Google Maps a fronte di validazione, impedendone il riutilizzo da parte di terzi. In questo capitolo introdurrò i diversi tipi di utilizzo di OpenStreetMap e le sue funzionalità.

1.1.1 Definizioni e utilità

Open Street Map raccoglie dei dati cartografici di tutto il mondo, rendendoli però disponibili liberamente per utenti ed eventuali sviluppatori. Infatti, offre numerosi vantaggi sia per gli sviluppatori, che possono modellare a proprio piacimento le mappeOSM, sia per gli utenti di tutto il mondo che, grazie ai vari software che si basano su OSM, possono fruire di una quantità davvero variegata di informazioni sull'ambiente urbano e sui punti di interesse.

Le mappe e i dati di Openstreetmap si possono anche consultare dal sito web del progetto stesso [WOSM2015], dove si può navigare nella mappa visualizzata come mostrato in figura 1.1.1, oppure iscriversi e entrare a far parte della community di OSM.

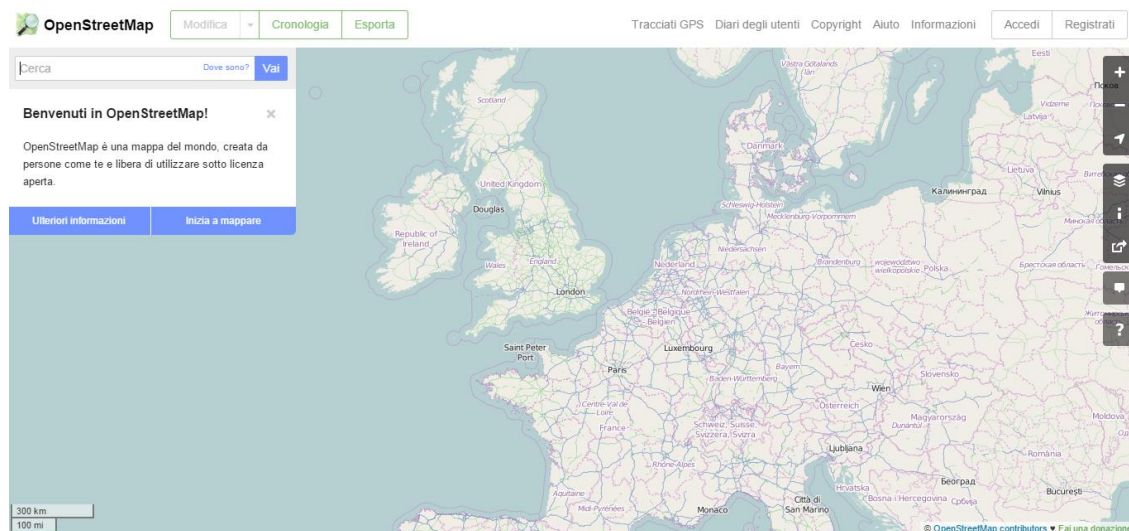


Figura 1.1.1 Mappa di OpenStreetMap

Il principale sistema di posizionamento utilizzato per ottenere dati geolocalizzati è il GPS che, attraverso una rete satellitare dedicata di satelliti artificiali in orbita, fornisce ad un terminale mobile o ricevitore GPS informazioni sulle sue coordinate geografiche ed orario, in ogni condizione meteorologica, ovunque sulla

Terra o nelle sue immediate vicinanze ove vi sia un contatto privo di ostacoli con almeno quattro satelliti del sistema. La localizzazione avviene tramite la trasmissione di un segnale radio da parte di ciascun satellite e l'elaborazione dei segnali ricevuti da parte del ricevitore [WP2015].

1.1.2 La community e la raccolta di informazioni

Per entrare a far parte della community, basta iscriversi al sito [WOSM2015], l'iscrizione permette di ricevere per mail gli aggiornamenti riguardanti discussioni attive nel forum e domande non risolte. Oltre a chiedere e/o a rispondere alle domande, si può anche partecipare attivamente, tramite il proprio account, a mappare nuovi posti oppure ad aggiungere nuove informazioni su posti già presenti. Vi sono due modi per contribuire attivamente al progetto:

- **Senza GPS:** Si può collaborare alla mappatura anche senza il GPS, l'importante è avere una connessione ad internet. Si possono: aggiungere i nomi delle vie dove sono mancanti, inserire punti di interesse quali mete turistiche, negozi, fontane, servizi ecc. oppure correggere eventuali errori. Si può utilizzare le foto aeree disponibili sul Portale Cartografico Nazionale, distribuiti tramite servizi online WMS (Web Map Server), per derivare dati per il progetto OpenStreetMap (si può fare solo con due editor specifici per OSM: JOSM [JOSM2015] o Merkaator [WOSM2015]). Si può anche utilizzare Walking Papers [WP2015], questo permette di stampare una zona e poi segnare su questa le modifiche da fare. Ovviamente si può usare questo strumento dove sono presenti già dati sul database da utilizzare come base ed è molto utile per aggiungere punti di interesse nei centri urbani. Per sfruttare al meglio il foglio modificato è bene avere uno scanner per importarlo nel PC e utilizzarlo con altri software in primis JOSM.

- **Con GPS:** Le tracce GPS non possono essere salvate direttamente nel database di OpenStreetMap, sono però estremamente utili come base su cui ricalcare le vie e i nodi mediante i software a disposizione, come Potlatch[WOSM2015] o JOSM, quest'ultimo software permette di sincronizzare le proprie tracce GPS con la registrazione audio. Il registratore può essere usato in due modi, il primo è di accendere il registratore in concomitanza dell'inizio della registrazione della traccia GPS e lasciarlo acceso; bisogna ricordarsi di prendere un punto di interesse per ogni

elemento registrato vocalmente in modo tale da potersi spostare facilmente da un punto all'altro; questa metodologia può essere molto comoda quando si è alla guida di un mezzo di trasporto come la bicicletta o l'automobile. L'altro metodo consiste nell'accendere il registratore solo in concomitanza della mappatura di un punto di interesse.

Ovviamente non solo le strade sono importanti per OpenStreetMap ma anche i punti d'interesse che verranno poi salvati nella mappa con specifici attributi, quali:

- `Segnavia(tourism=information & information=guidepost);`
- `bivacchi (amenity=shelter), rifugi (tourism=alpine hut);`
- `fontane d'acqua potabile (amenity=drinking water) e molto altro`

ancora.

Dopo di che, grazie a JOSM, si potrà caricare le nostre tracce GPS direttamente dal sito [WOSM2015].

L'aggiornamento delle mappe non sarà immediato, perché verranno poi analizzate e renderizzate.

1.2 Humanitarian OpenStreetMap Team

L'Humanitarian OpenStreetMap Team (HOT) è un gruppo di OSMapper (così viene chiamato chi partecipa a OpenStreetMap) che utilizza il progetto (e i dati raccolti) per scopi umanitari.

La prima volta che OpenStreetMap è stato utilizzato per queste finalità è avvenuto alla ripresa delle ostilità tra Israele e Palestina nel 2009: la comunità si è autofinanziata per acquistare le ortofoto recenti della Striscia di Gaza in modo tale da poter digitalizzare i dati.

Il caso più eclatante, invece, è stato in concomitanza di una delle più grandi catastrofi naturali negli ultimi anni, il terremoto ad Haiti. In questa occasione Google ha sovvenzionato l'acquisto delle ortofoto della situazione post terremoto, e gli utenti hanno provveduto celermente alla digitalizzazione, segnalando tra le altre cose la presenza di campi di soccorso, i ponti distrutti e altri elementi utili ai soccorritori; inoltre sono stati messi in piedi diversi servizi per fare in modo che si potessero utilizzare facilmente i dati presenti sul database aggiornati quasi in tempo reale, vi era la possibilità di trovarli in formato Garmin (per gli operatori che si dovevano spostare

da un posto all'altro), in formato immagine per essere stampata (per coordinare gli aiuti dai campi di soccorso) inoltre erano presenti diversi siti online che avevano creato strati informativi dedicati all'isola caraibica. Questa tragica esperienza ha mostrato al mondo come OpenStreetMap possa essere utile e come i dati raccolti dagli utenti sono in certi casi essenziali, non a caso infatti erano gli unici aggiornati al post terremoto e utilizzabili durante la situazione d'emergenza [DEL2012].

1.2.1 Mapping Party

I mapping party sono eventi legati al progetto, durante i quali un certo numero di OSMapper, sceglie una zona, solitamente poco mappata oppure da completare, incomincia a pubblicizzare l'evento all'interno della comunità e all'esterno contattando enti pubblici, associazioni e media per diffondere la manifestazione. Il contatto esterno alla comunità è molto importante per cercare di coinvolgere nuove persone all'interno del progetto. Solitamente i mapping party si tengono nel corso del fine settimana per cercare di far affluire più persone possibili. Uno dei più importanti Mapping Party avvenuto in Italia, è stato quello di Pompei, con scopi archeologici all'interno dei resti romani della nota località napoletana [WOSMMP2015].

1.3 La Struttura del database OpenStreetMap

Nel database OpenStreetMap, possono essere inseriti degli elementi (come strade, negozi, ecc.), tramite alcuni software che vedremo in seguito: gli elementi sono di quattro tipologie:

Punti (node): è l'elemento di base di tutti gli elementi di OSM, i nodi sono definiti tramite la latitudine e la longitudine ed ha un id univoco.

Linee (way): un insieme di ordinato che contiene almeno 2 fino ad un massimo di 2000 nodi, un esempio di way è una via.

Aree (polygon): un insieme di punti dove il primo nodo corrisponde all'ultimo nodo, ad esempio il sentiero di un lago, questo viene rappresentato con il tag `area=yes`.

Relazioni (*relation*): un insieme degli elementi precedenti, per esempio una linea degli autobus che è composta da più strade e dalle sue fermate.

Oltre a questi elementi, vi sono altre due tipologie molto importanti e usate: i tag e le relazioni (*relation*)

I Tag

Le etichette (*tag*) servono per descrivere le caratteristiche dei vari elementi. I tag sono sempre composti da una coppia di nomi. Il primo è detto *key*, il secondo *value*.

Solitamente la *key* descrive una famiglia di caratteristiche, mentre il *value* va più nello specifico. Ad esempio la *key* `highway` indica la famiglia delle strade di qualsiasi tipo, dalle autostrade ai sentieri. Ecco alcuni esempi nella tabella 1.3.1:

Key	Value	Descrizione
highway	Motorway	Autostrada
	Trunk	Superstrada
	Primary	Strada di importanza nazionale
	Secondary	Strada di importanza regionale
	Tertiary	Strada di importanza locale
	Unclassified	Strada del reticolo di base
	Residential	Strada per abitazioni
	Service	Strada di servizio
	Tranck	Strada agricola o forestale
	Pedestrian	Via pedonale cittadina
	Footway	Sentiero
	Cycleway	Pista ciclabile
	Steps	Scala
	Bus_stop	Fermata dell'autobus
	Stop	Segnale stop
Traffic_Signals	Semaforo	

Tabella 1.3.1 Key highway

I tag presenti non rappresentano solo un elemento, ma possono essere usati per più elementi: per esempio `highway` è prevalentemente associato alle linee ma, come si può notare nella tabella 1.3.1, vi sono alcuni casi in cui è utilizzato con i nodi `highway=bus_stop` o `highway=traffic_signals`.

I tag usati sono tantissimi e continuano ad aumentare o migliorare, permettendo di mappare qualsiasi elemento possa essere rappresentato da una coppia di coordinate geografiche. Una vasta lista è disponibile nel sito [WOSMMAP2015]. Inoltre è possibile controllare, discutere e votare i nuovi tag proposti tramite la pagina dedicata [WOSMPROP2015]. Oltre ai tag per le strade, esistono anche molti tag per elementi puntuali, lineari e areali.

Eccone alcuni mostrati nella *Tabella 1.3.1.2*:

Elemento	Key	Value	Descrizione
Puntuale	Amenity	Pub	Pub
		Bank	Banca
	Shop	Supermarket	Supermercato
		Bakery	Panificio
Tourism	Hotel	Albergo o hotel	
	Information	Punto informazioni turistiche	
Railway	Station	Stazione ferroviaria	
	Level_crossing	Passaggio a livello	
Lineare	Aerialway	Cable_car	Funivia
		Chair_lift	Seggiovia
Waterway	River	Fiume	
	Canal	Canale	
Railway	Rail	Ferrovia	
	Tram	Linea tram	
Areale	Natural	Water	Fiume molto largo o lago
		Wood	Foresta
	Leisure	Playground	Parco giochi
Sport_center		Stadio	
landuse	Residential	Zona residenziale	
	vineyard	vigneto	

Tabella 1.3.1.2 Elemento-Key

Per ogni elemento della *Tabella 1.3.1.2* è anche possibile assegnare più di un tag in modo da descriverlo nel miglior modo possibile come nella *Tabella 1.3.1.3*:

Key	Value
highway	unclassified
name	Via roma
foot	yes
bicycle	no
oneway	yes

Tabella 1.3.1.3 Altri tag

Le Relation

Le relation servono per mettere in relazione più elementi singoli, diverse vie oppure vie e punti. Esistono svariate tipologie di relation, nella *Tabella 1.3.1.4* ne mostreremo alcune [WIKIOSMTYPES].

Tipo	Descrizione
associatedStreet	Serve per mettere in relazione le vie con i numeri civici
multipolygon	Serve per creare poligoni complessi che si compongono di più d una way
restriction	Serve per vietare le soste
boundary	Serve per raggruppare aree e creare enclave ed exclavi
route	Serve per creare dei percorsi, posso essere pedonali (sentieri di montagna), ciclabili, linee di trasporti pubblici ecc.
enforcement	Serve per inserire elementi per misurare e documentare le violazioni veicolari
destination sign	Serve per dare informazioni sulle destinazioni agli incroci e alle svolte (molto utile per i programmi di routing)

Tabella 1.3.1.4 Relation

La Relation la `route` è probabilmente la più utilizzata ed importante.

Come visto nella *Tabella 1.3.1.4*, la relation `route` serve per creare dei percorsi di diverso tipo: pedonali (sentieri di montagna o urbani), ciclabili, linee di trasporti pubblici ecc.

Inoltre, viene utilizzata anche per indicare che più percorsi (`way`) diversi fanno parte di una stessa rotta. Gli unici due campi obbligatori sono `type` e `route`; gli altri, pur non essendo richiesti, sono ugualmente importanti e, quando presenti, è bene utilizzarli.

Key	Value
<code>type</code>	Route
<code>route</code>	road-bicycle-foot-hiking-bus-ferry-canal-pilgrimage - detour - railway - tram - trolleybus -mtb (mountainbike) - roller skate - running - horse -parade - protest march (recurring)
<code>ref</code>	codice identificativo se presente
<code>operator</code>	nome dell'operatore se presente
<code>name</code>	nome se presente
<code>symbol</code>	simbolo se presente

Tabella 1.3.1.5 Type route

Per creare una route bisogna quindi aggiungere i tag e poi selezionare quali elementi già presenti sul database ne fanno parte. Ad esempio, come potremo vedere in Figura 1.3.1 dove vi è la mappa di Cesena:

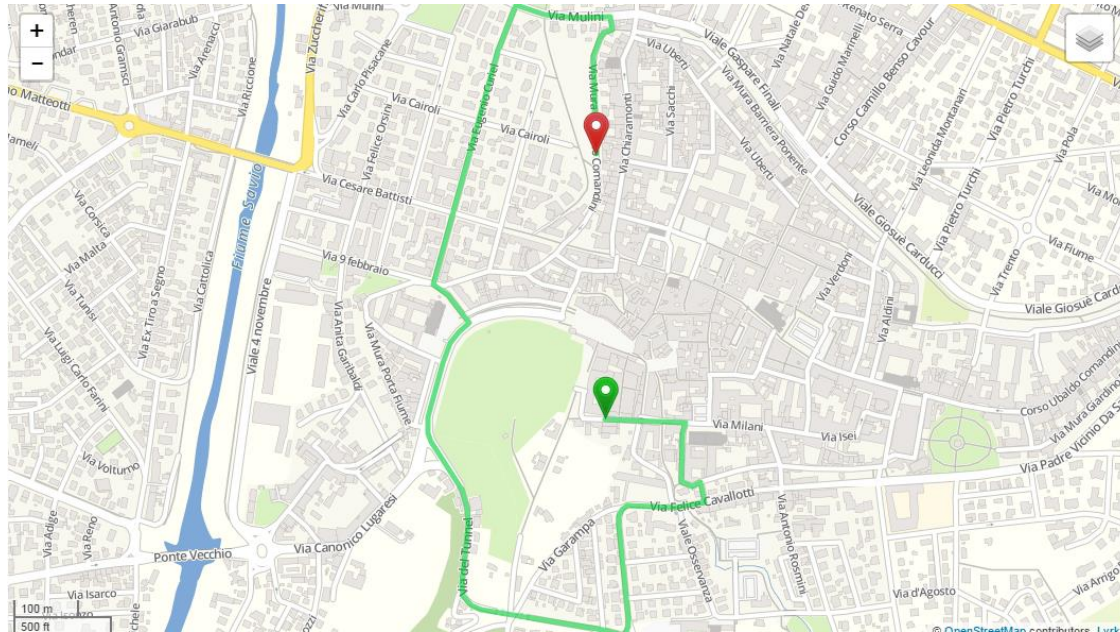


Figura 1.3.1 Calcolo percorso

la figura sarà associata al Codice 1.3.1.6, dove più nodi faranno parte di una stessa route, e i tag “” si riferiscono ai tag che abbiamo visto sopra.

1.3.1 JOSM

JOSM (Java OpenStreetMap Editor) è l’editor per OpenStreetMap più utilizzato, sviluppato da Immanuel Scholz e attualmente mantenuto da Dirk Stöcker. Come linguaggio usa Java e lavora principalmente offline infatti tutto ciò che fate non sarà visibile a nessuno finché non lo caricherete nel server [JOSMWIKI2015].

Oltre a JOSM, ci sono altri programmi di editing per i dati di OpenStreetMap, come l’editor in linea Potlatch oppure Merkaartor. Quando dunque si dovrebbe utilizzare JOSM e quando un altro editor? JOSM è un editor ricco di funzionalità per gli utenti che hanno già una certa esperienza nella mappatura. Esso richiede alcune nozioni per la sua installazione e configurazione. Se si ha intenzione di diventare un buon mappatore di OSM, allora vale la pena spendere un po’ di tempo per imparare ad utilizzarlo[WOSM2015].

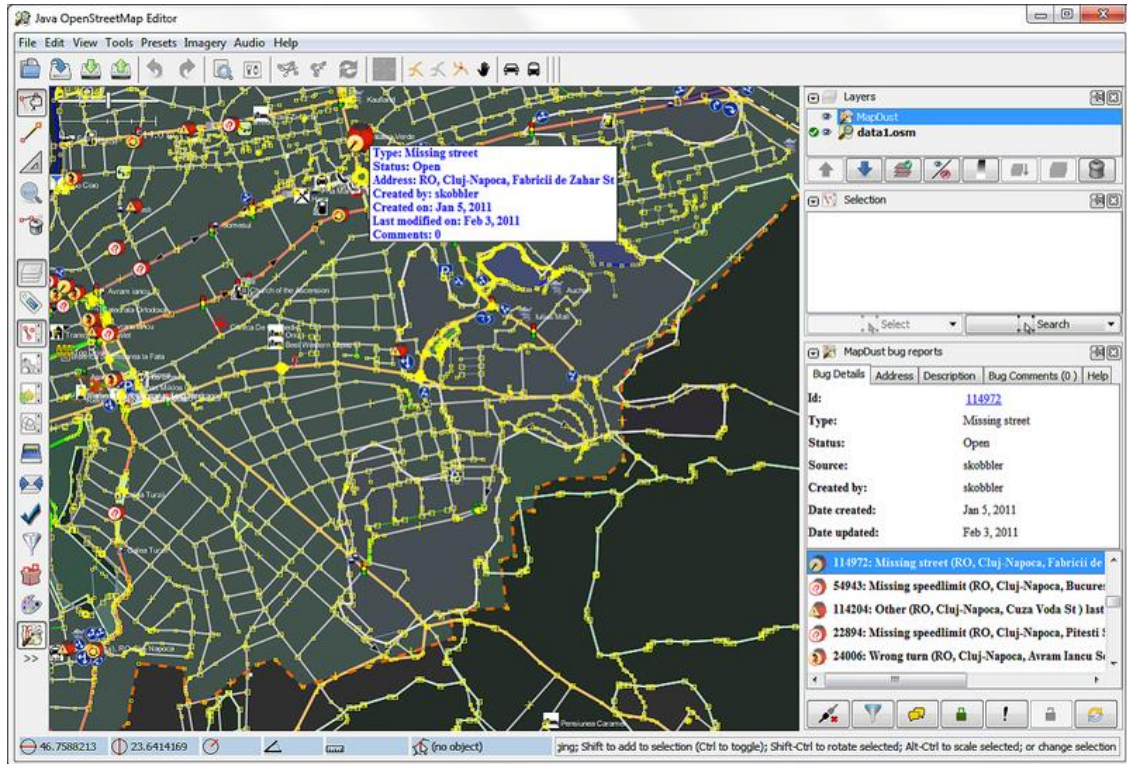


Figura 1.3.1.1 JOSM

Ora vedremo più nello specifico come usare JOSM e i suoi *plugin*. Esiste un tipo di formato di file di JOSM che serve per salvare/caricare file di dati. Questi dati hanno un tipo di estensione `.osm` che è lo stesso che si può ottenere semplicemente con una query di richiesta al server, ma nel file `.osm` vi sono qualche info in più. Un esempio di file `.osm` è presente nel *Codice 1.3.2.1 File osm*:

```
<?xml version='1.0' encoding='UTF-8'?>
<osm version='0.5' generator='JOSM'>
<bounds minlat='51.5076478723889' minlon='-0.127989783553507'
maxlat='51.5077445145483' maxlon='-0.127774884645096'
origin='OpenStreetMap server' />
<node id='26821100' timestamp='2009-02-16T21:34:57+00:00'
user='dankarran' visible='true' lat='51.5077286' lon='-0.1279688'>
<tag k='created_by' v='Potlatch 0.10f' />
<tag k='name' v='Nelson&apos;s Column' />
<tag k='tourism' v='attraction' />
<tag k='monument' v='statue' />
<tag k='historic' v='monument' />
</node>
<node id='-1' visible='true' lat='51.507661490456606' lon='-
0.1278000843634869' />
<node id='346364767' action='delete' timestamp='2009-02-
16T21:34:44+00:00' user='dankarran' visible='true' lat='51.5076698'
lon='-0.1278143' />
</osm>
```

Codice 1.3.2.1 File osm

Come abbiamo visto nella struttura di un database OSM, i vari tag presenti nel *Codice 1.3.2.1* hanno un significato ben preciso e verranno poi interpretati dal database per stilare le mappe.

I tag vengono salvati come una coppia di testo, chiamate chiavi e valori e, se si seleziona un oggetto in JOSM, si potrà vedere tutti i tag ad esso associati, visibili nel riquadro “Proprietà”. Oltre a questo si può aggiungere, modificare o cancellare un tag. Chi utilizza JOSM, si affiderà anche ad alcuni plug-in molto utili per salvare le mappe, i più importanti sono:

`Restart`, aggiunge un pulsante riavvia.

`Mirrored Download`, permette di scaricare più velocemente i dati di OSM, così invece di usare il server centrale di OSM, possiamo configurare un mirror, che è una replica esatta dei dati, ma che potremo interrogare direttamente e quindi avere delle risposte in un lasso di tempo più breve.

`Direct Upload`, permette di inviare i dati del proprio GPS al server OSM direttamente tramite JOSM, quindi permette una geolocalizzazione immediata della propria posizione che potremo condividere con gli altri utenti.

`Editgpx`, permette di modificare i dati del proprio GPS prima di inviarli ad OSM. Serve nel caso in cui vogliamo eliminare dei dati che riteniamo inutili, il tutto può essere fatto nel totale anonimato.

1.4 Rendering

Genericamente effettuare il *Rendering* significa prendere dei dati geospaziali grezzi e creare da questi una mappa visualizzabile. Spesso la parola viene applicata più specificatamente alla produzione di una immagine di tipo *raster* o di più parti di immagini raster (dette *tiles*), ma può anche indicare la produzione di mappe in formato vettoriale. E' anche possibile fare un Rendering tri-dimensionale ("3D rendering") sempre usando dati geospaziali come input. La possibilità di effettuare il rendering di mappe in nuovi e differenti stili, o la possibilità di evidenziare caratteristiche di particolare interesse, è uno degli aspetti più eclatanti quando si ha accesso a dati geografici liberi. Gli sviluppatori legati alla comunità di OpenStreetMap hanno creato una grande varietà di software per il rendering dei dati OSM. Avendo una struttura

nota, i dati OpenStreetMap possono anche essere convertiti in altri formati per poter essere quindi utilizzati con altri software di rendering già esistenti[WOSM2015].

1.4.1 Elenco dei principali Software per il Rendering

Esistono due tipi di rendering: uno 2D e l'altro 3D.

Le principali scelte per un rendering 2D su mappe di tipo raster sono:

- *Mapnik* - Richiede *PostgreSQL* e diverse librerie *C++*. E' progettato per essere veloce ed adatto alla generazione delle *tiles* su servers di fascia alta. La sua installazione non è particolarmente semplice, verrà poi approfondito in seguito[MAPNIK2015].

- *Osmarender* - Utilizza *XSLT* per trasformare dati OSM in formato *XML* in formato vettoriale *SVG*. Il formato *SVG* permette ulteriori elaborazioni e correzioni successive (vedi si seguito). *Osmarender* può essere una valida alternativa per creare renderings unici[WOSMOAR2015].

Altri principali renderers, inclusi i renders vettoriali sono:

- *Maperitive* è una applicazione desktop per il rendering di dati su file locali (.osm, .osm.bz2 and GPX), con regole di rendering definite con files di testo. Genera mappe in formato BMP, PNG e SVG. Non è progettato per rendering di alte prestazioni né per processare grandi volumi di dati, ma è facile da installare, è flessibile ed è valido per rendering personalizzati (ad esempio la mappa della propria città). E' anche un generatore di tiles, curve di isolivello, elevazioni colorate e con molte altre caratteristiche.Funziona su Windows, Linux, Mac.E' scritto in C#[MAPERITIVE2015].

- *CartoType* è un sistema di rendering multi-OS, con una libreria di routing con licenza d'uso. Funziona su sistemi Windows desktop e mobile e su sistema operativo Symbian. E' scritto in C++ [CARTO2015].

- *Mapsforge* - set di strumenti gratuiti che permette alla comunità di creare facilmente nuove applicazioni basate su OpenStreetMap. Gli strumenti e le API offrono soluzioni per: il rendering, il calcolo delle rotte e la navigazione, indicizzazione e ricerca di POI, overlay per le mappe e altro. Scritto in Java, può essere usato anche nei dispositivi Android [MFORGE2015].

- *Mapweaver* - Successore di mapgen.pl è un sistema di rendering in Perl per convertire dati OSM in SVG e con Inkscape in PNG or PDF. Supporta elenchi di

strade e POI, e overlay di griglie e coordinate. Permette l'estrazione automatica di place desiderati da file OSM. Non funziona su Windows [WOSMMW2015].

- *osmrender.pl* - Programma in Perl per generare semplici mappe. Usa dati *.osm come input [WOSMOR2015].
- *osmbrowser* - Può caricare ampie zone di dati OSM ed eseguire il rendering in grafica vettoriale [FOSM2015].
- Per il 3D Rendering, le principali scelte sono:
- *OSM2World* - Tool Java per la creazione di modelli 3D da dati OSM [OSM2W2015].
- *ShugenDoMap* - Programma di rendering 3D in tempo reale e multi-piattaforma. Progettato in particolare per piattaforme mobili come smartphones e tablets [WOSMSDM2015].
- *Kendzi3d* - Plugin per JOSM - Consente la visualizzazione dei dati modificati in 3D [WOSMK3D2015].
- *OpenStreetMap 3D* - OSM in 3D tramite web service [OSM3D2015].

1.4.2 Mapnik

Mapnik è scritto in C++ e include associazioni di alto livello in Python [MAPNIK2015]. Usa la libreria AGG e offre rendering anti-aliasing con accuratezza al sottopixel. Può leggere i file di tipo ESRI, PostGIS, raster TIFF, file .osm e ogni formato supportato da GDAL o OGR. I pacchetti sono disponibili per molte distribuzioni Linux e i file binari sono disponibili anche per Mac OS X e Windows. Mapnik è un toolkit open source per la creare le mappe, per farlo utilizza principalmente i file .osm, infatti i file vengono usati per delineare le mappe. OSM utilizza Mapnik per disegnare dei riquadri 256x256 pixel, che sono poi forniti dal server dei tile (tile.openstreetmap.org). I dati OSM sono convertiti in un formato utilizzabile da Mapnik da *osm2pgsql*, che carica i dati derivati su un database PostGIS.

1.4.3 OpenStreetMap 3D

Il progetto OSM-3D mira a fornire una visione degli elementi OpenStreet Map in 3

dimensioni. Attualmente il progetto è gestito dall'Università di Heidelberg, GIScience Grup (<http://gisscience.uni-hd.de>). È usato per dimostrare le potenzialità delle mappe in 3D e come le 3D Spatial Data Infrastructures (3D-SDI) possono essere impostate utilizzando interfacce standardizzate e aperte. La visualizzazione della mappa va oltre la semplice rappresentazione piastrelle cache che sono in qualche modo distorte in modo da creare un effetto di prospettiva. È basato su un modello grafico che incorpora un modello elevato di molte caratteristiche OSM che si possono elaborare. Questo grafico di scena può essere esplorato utilizzando il visualizzatore *XNavigator*. La preparazione dei dati OSM per essere visualizzati in OSM-3D richiede un sacco di passaggi di pre-elaborazione che sono fatti sui server di GIScience.

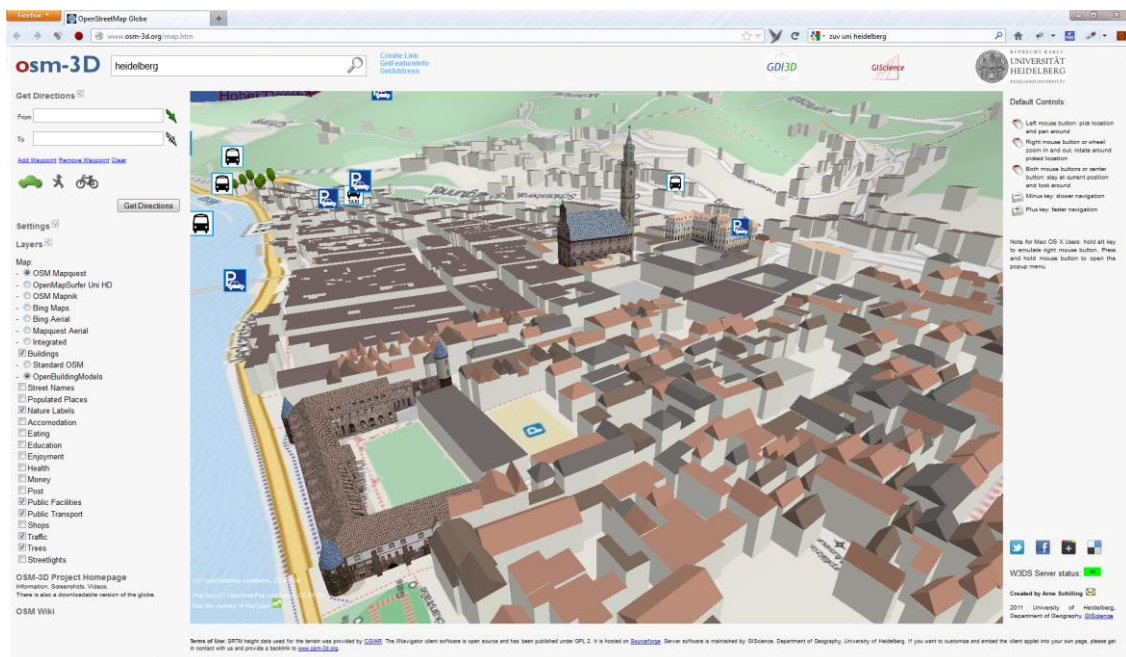


Figura 1.4.2

Gli edifici che sono visibile nella *Figura 1.4.2* sono renderizzati come poliedri, i footprint vengono estrusi con il tetto piatto. L'altezza base viene interpolata dal DEM (Modello digitale di elevazione). In OSM, gli edifici sono modellati come vie rappresentanti il footprint alla base o come relazioni, con un anello esterno e 0..n anelli interni con cortili. Gli anelli interni vengono elaborati correttamente e vengono visualizzati come buchi nel poliedro. Gli anelli esterni costituiti da più di un via non possono essere elaborati. La parte inferiore e la parte superiore sono generalmente piatte. Qualora le informazioni del tetto siano disponibili, l'algoritmo cerca di ricostruire la geometria del tetto dal tipo di tetto (ad esempio a due spioventi) e dall'allineamento. In alcuni casi, gli edifici su pendii ripidi possono causare problemi e

le parti possono essere appese sopra il suolo. Per questo motivo, le pareti non devono essere etichettate come edifici [OSM3D2015].

1.5 Il Routing in OpenStreetMap

I servizi di Routing aiutano le persone nel muoversi da un posto ad un altro. I dati contenuti nel database di OpenStreetMap includono informazioni utili per diversi tipi di routing: macchine, a piedi, bicicletta, mountain bike e a cavallo. Esistono molti servizi di routing, sia offline che online che utilizzano i dati di OpenStreetMap. Esistono anche software per creare nuove applicazioni e di continuo nuovi dati sono aggiunti al database delle mappe per permettere un uso migliore. Sono inoltre disponibili varie mailing-list per avere più informazioni [WOSM2014].

1.5.1 Le tipologie di Routing

Esistono differenti tipologie di Routing e anche differenti tipologie di Software che permettono di fare routing.

Principalmente il routing può essere fatto offline, online (usando principalmente web-server) oppure tramite applicazioni di navigazione.

I principali linguaggi utilizzati sono:

- Java, l'applicazione *Traveling Salesman* contiene un navigatore di mappe OSM e una libreria base, ma non un'interfaccia grafica. Tutte le parti più importanti possono essere cambiate con plugin.
- C/C++, il software di routing *Gosmore* è anche un visualizzatore di dati OSM XML come i planet.osm o i file di Geofabrik.
- C#, IMORTIS (Intermodal Transport Routing Informations-System) utilizza un approccio che tende all'ottimizzazione della navigazione di differenti veicoli utilizzando un algoritmo A*.

1.5.2 I profili presenti in OpenStreet Map

I profili in OpenStreetMap sono importanti sia per decidere quali tag usare in una mappa, sia per come deve essere strutturato il software che verrà poi realizzato.

Vi sono 4 categorie principali che vengono prese in considerazione dai principali software che permettono di calcolare un percorso che tiene conto delle caratteristiche (tag) presenti su archi e nodi.

Questi profili sono:

- **Automobile:** la maggior parte delle query per calcolare un percorso vengono fatte per spostarsi in automobile. La richiesta può esser fatta in modo statico oppure attivando il navigatore, quindi runtime. In quest'ultimo caso occorrerà avere un software che, permette il ricalcolo automatico e dinamico del miglior percorso. Considerando le automobili come un profilo, bisogna considerare che non hanno accesso a diversi luoghi, come le zone pedonali, parchi, sentieri di montagna ecc. . Il tag che lo caratterizza è *motorcar* che può antecedere il tag *barrier* con i seguenti valori: *yes/no/limited*; questi valori saranno gli stessi anche per tutti gli altri profili.

- **Bicicletta:** la bicicletta necessita di un software che, come l'automobile, possa permettere il ricalcolo automatico e dinamico del miglior percorso con però una maggiore attenzione su eventuali piste ciclabili che normalmente non sono praticabili da un'autovettura. Il tag che lo contraddistingue è il tag *bicycle*.

- **Pedone:** i software che calcolano dinamicamente un percorso per un pedone, considerano molti elementi in più che non vengono invece presi in considerazione nelle altre categorie. Un pedone può principalmente spostarsi a suo piacimento in qualsiasi luogo (tranne ovviamente in superstrada/autostrada), può quindi avere accesso a sentieri di montagna, a zone con una barriera architettonica che normalmente non sono praticabili dalle altre categorie. Il tag che lo contraddistingue è il tag *foot*.

- **WheelChair:** avrà le stesse limitazioni del profilo pedone con una maggiore attenzione, per calcolare un percorso, delle eventuali barriere architettoniche presenti. Il tag che lo caratterizza è il tag *wheelchair*.

Ovviamente oltre a questi tag sono presenti molti altri tag che non si riferiscono esplicitamente a profili ma che sono comunque riconducibili per caratteristiche e limitazioni alle categorie sopracitate. Come detto prima, le limitazioni sono dovute alla presenza del tag *barrier*, ma possono essere anche dovute al tag *access* che caratterizza l'accesso ad una determinata tipologia di strada (caratterizzata dal tag *highway* principale tag per etichettare le strade). Ad esempio in una superstrada, caratterizzata dal tag *highway=motorway*, vi sarà anche il tag

`access=motor_vehicle`, significa che vi possono circolare tutti i veicoli a motore quindi esclude pedoni, biciclette ecc..

1.6 Open Source e il Software Proprietario

Il database di OpenStreetMap è pubblicato secondo la licenza Open Database License. La cartografia contenuta nelle tavole e la documentazione sono invece rese disponibili sotto licenza Creative Commons Attribuzione - Condividi allo stesso modo 2.0 (CC-BY-SA) [WIKI2015]. Tutti i dati inseriti dagli utenti devono essere caricati secondo una licenza compatibile con la Creative Commons Attribuzione-Condividi allo stesso modo. In questo modo possono essere inclusi anche dati nel pubblico dominio o sotto altre licenze compatibili. Tutti i contributori devono essere registrati al progetto ed accettare di sottoporre i propri dati con la licenza Creative Commons BY-SA 2.0 specificare che la licenza con cui i loro dati sono forniti è compatibile con questa. Con il passare del tempo, ciò ha portato ad esaminare le licenze con cui i governi concedono i propri dati per l'utilizzo pubblico, per determinare la compatibilità di tali dati con il progetto. L'uso di dati non liberi è un problema particolarmente gravoso nella cartografia, in quanto l'unica maniera per provare che non sono stati utilizzati dati non liberi sarebbe quella di riportare l'intera area interessata (anche se molto estesa) alla condizione in cui era prima dell'importazione dei dati in questione.

Il software usato nella preparazione e presentazione dei dati di OpenStreetMap è stato originato da molti progetti differenti e ognuno di questi può avere una licenza differente. Il core software per la gestione del database, l'interazione utente e le API sono disponibili sotto GNU General Public License.

Una parte dei contributori ha deciso di rendere disponibili i propri contributi con licenza di pubblico dominio [WIKI2015].

Nonostante OpenStreetMap sia Open Source e cioè è possibile utilizzare liberamente tutti i dati presenti, con il solo vincolo di citare la fonte e usare la stessa licenza per eventuali lavori derivati da OSM, molti software, risultano con licenze non opensource.

Il costo dei software, può variare da 1 euro fino a 500 euro, l'elenco completo dei software, con le relative caratteristiche e i relativi prezzi è disponibile nel sito [WOSMSOFT2015]:

Eccone alcuni esempi presi dal sito:

- Autosputnik, ha una Licenza propria e un costo di 15 euro, può essere usato in Windows Mobile, Windows CE 5/6, Windows XP/Vista/7 [ASN2015].
- Spatial Manager Desktop, è un applicazione desktop progettata per elaborare dati spaziali in modo semplice, veloce e non dispendioso [SMANAGER2015].

2 OSM software e accessibilità urbana

In questo capitolo verranno illustrati i vari desktop e mobile software che utilizzano i dati e le mappe di OSM per fornire servizi che riguardano l'accessibilità in ambiente urbano, analizzando in particolare il profilo wheelchair. Esso permette infatti di richiedere all' algoritmo di routing di calcolare un percorso che tenga conto delle specifiche necessità di un utente che vuole evitare barriere architettoniche ben definite. Verranno introdotte quindi le diverse tipologie di barriere architettoniche e le problematiche che ne derivano quando si incontrano in percorsi urbani.

2.1 Le barriere architettoniche

Per introdurre il profilo *wheelchair*, bisogna introdurre il discorso delle barriere architettoniche, che devono essere sempre prese in considerazione durante il calcolo di un percorso nel caso l'utente abbia necessità specifiche.

Con il termine barriere architettoniche si indica l'insieme di quegli elementi che non permettono la completa mobilità alle persone, temporaneamente o permanentemente, in condizioni limitate di movimento, quindi costituiscono un problema non solo per le persone con disabilità, ma per tutti coloro (bambini, gestanti, anziani, cardiopatici, ecc.) che, per diversi motivi, non possono muoversi liberamente. Ciò che accomuna le barriere architettoniche al problema dell'accessibilità, sta proprio nella definizione di quest'ultima, infatti secondo lo standard ISO TS 16071, l'accessibilità è "l'usabilità di un prodotto, servizio, ambiente o strumento, per persone col più ampio raggio di capacità" [ISO16071]; quindi le barriere architettoniche

rendono di fatto alcuni luoghi inaccessibili a tutte le persone in condizioni limitate (temporanea o permanente) di movimento.

E' un diritto di tutti muoversi, anche di chi ha limitate capacità motorie: lo stabilisce anche la legge. Infatti la legge n.13 del 1989 stabilisce i termini e le modalità con le quali deve essere garantita l'accessibilità a tutti soprattutto negli ambienti pubblici. La legge (ed altri interventi successivi) stabiliscono anche la concessione di contributi per l'abbattimento delle barriere su immobili privati.

Le barriere architettoniche vanno dunque considerate come una caratteristica negativa dell'habitat dell'uomo, riguardante la generalità dei cittadini. Si avrà un maggior beneficio per l'accessibilità urbana se, nell'ambiente urbano, vi saranno poche barriere architettoniche. L'accessibilità va quindi intesa come la possibilità, da parte di una utenza ampliata, di accedere agevolmente a spazi, edifici, attrezzature e sistemi di trasporto. Per accessibilità, al di là dell'esigenza di poter disporre di spazi privi di barriere architettoniche, si intende anche il comfort ambientale e la sicurezza urbana, come la presenza di attraversamenti pedonali ben visibili e segnalati opportunamente, in situazioni di totale autonomia, anche per chi ha ridotte capacità motorie o sensoriali, per periodi brevi o in maniera permanente. Quest'aspetto urbano deve essere preso in considerazione quando si affronta qualsiasi tipo di progetto.

2.1.1 Le barriere architettoniche presenti in Openstreetmap

Openstreetmap considera, attraverso opportuni tag, i seguenti elementi come barriere architettoniche, tutte caratterizzate da due tag principali: `barrier="tipo di barriera"`, `wheelchair="yes/no/limited"`.



Figura 2.1.1.1 Pavimentazione irregolare

- Percorsi con pavimentazione sdruciolevole, irregolare o sconnessa, e/o con eccessiva estensione come presenti in *figura 2.1.1.1* , sono caratterizzate dal tag `barrier="debris"` e molte volte limitano, o nei peggiori dei casi negano, l'accesso al profilo *wheelchair*, quindi sarà presente il tag `wheelchair="limited/no"`.



Figura 2.1.1 Scaletta

- Scaletta(come presente in figura 2.1.1): permette ai pedoni, ma non alle persone con disabilità motorie, di scavalcare un muro o una recinzione, ma senza 'aprire' la barriera (non ci sono parti in movimento). Si utilizza il tag `barrier="style"` e `stile_type="stepover"` per specificare il tipo di scaletta.



Figura 2.1.2 Cordolo

- Cordolo (come si può vedere nella figura 2.1.1), costituisce una barriera per biciclette e per persone con disabilità motorie che sono sulla sedia a rotelle. L'importante è indicare, se nota, l'altezza del cordolo tramite il tag `height=*` così da determinare se rappresenta una barriera o meno per diversi profili, ad esempio l'altezza di 1 cm potrebbe comunque risultare accessibile per un profilo wheelchair. Il lato destro rappresenta la base, il lato sinistro la sommità, ovviamente saranno anche caratterizzate dal tag `barrier="kerb"`.



Figura 2.1.3 Passaggio stretto

- Passaggi stretti (come presenti in figura 2.1.3): sono caratterizzati da un accesso molto ristretto che rappresenta una vera e propria barriera per il profilo wheelchair, quindi sarà presente il tag `wheelchair="no"` il tag utilizzato per distinguerlo è il seguente `stile="ladder"`.



Figura 2.1.4 Cancelli con una difficile apertura

- Cancelli con una difficile apertura: la difficoltà di questo tipo di cancello, sta nella posizione della maniglia che risulta troppo alta e quindi non accessibile a tutti, il tag che caratterizza questo tipo di barriera è `barrier="gate"`.



Figura 2.1.1.5 Scale prive di scorcimano

- Scale prive di corrimano, come presente in figura 2.1.1.5, sono identificate dal tag `barrier="step"` dove è possibile specificare anche se vi è un solo scalino, tramite il tag `step_count=1`, così questo caso potrebbe risultare accessibile.



Figura 2.1.4 Rampe con forte pendenza

- Rampe con forte pendenza o troppo lunghe, come presente in figura 2.1.4, saranno caratterizzate dal tag `ramp: wheelchair="no"`.



Figura 2.1.5 Ruscello

- Ruscelli, fiumi, un fossato o una gola, solitamente con un corso d'acqua sul fondo, dove non sono presenti ponti che collegano due sentieri e quindi non sono facili da attraversare, specialmente non a piedi. Il tag che lo caratterizza è `barrier="ditch"`, e viene specificato con il tag `waterway="stream"` o `waterway="drain"`.

Ora invece vedremo nello specifico i diversi software che permettono di calcolare un percorso, tenendo in considerazione tutte i tag riguardanti le barriere.

2.2 Software per computer desktop

Vi è una grande varietà di software disponibili e usati da Open Street Map nelle mappe e nei siti: ogni software è specifico per una determinata categoria. Principalmente i software per il desktop sono quelli più utilizzati e più elaborati. La maggior parte dei software funziona su una grande varietà di sistemi operativi ma la maggior compatibilità si ha con Windows, MAC OS, Linux, dove sono presenti numerose guide sulla installazione e su eventuali errori riscontrati.

Il miglior Software da utilizzare è specifico per ogni utente, o meglio, a seconda delle esigenze sono disponibili una grande varietà di desktop Software che si possono dividere in due categorie principali:

- Offline: la maggior parte dei Software lavora online e vengono direttamente utilizzati da alcuni siti internet (come Openstreetmap) per

permettere di calcolare un percorso o guardare semplicemente la mappa di una città. I software offline devono essere invece installati nel proprio computer, anche se però questo non sempre permette di utilizzarlo al meglio, infatti si potrà calcolare un percorso, ma non si potrà visualizzare la mappa, perché per farlo bisogna installare Mapnik[MAPNIK2015].

- **Inline:** i software inline sono preferiti dalla maggior parte dei programmi, così ogni programma permetterà all'utente che lo utilizza, di: calcolare il percorso e visualizzare e/o spostarsi nella mappa. Le richieste di calcolo di un nuovo percorso vengono direttamente fatte ai server online che risulteranno più rapidi nell'elaborazione di dati rispetto al proprio computer.

Di seguito parleremo dei software desktop specifici per il profilo *wheelchair*. Esso permette infatti di richiedere all'algoritmo di routing di calcolare un percorso che tenga conto delle specifiche necessità di mobilità di un utente che vuole evitare determinate barriere architettoniche (precedentemente descritte) che altrimenti lo bloccherebbero nel raggiungimento della sua destinazione.

2.2.1 Routino

Questo calcolatore d'itinerario utilizza i dati di osm e permette di calcolare un percorso tra due punti nel modo più corto possibile o nel modo più rapido possibile. Per ottimizzare i calcoli, vengono creati dei file specifici che contengono le strade suddivise in segmenti ma che verranno poi unite con le stesse proprietà.

Il processo di elaborazione dei file di input come XML, PBF o O5m si basa su delle regole di un file di configurazione che trasforma automaticamente i tag autostradali in tag che vengono elaborati da Routino. La generazione dei file di output (HTML e GPX) utilizza tag scelti da un altro file di configurazione che permette l'output in diverse lingue.

L'itinerario può essere calcolato per tutti i principali tipi di veicoli con i loro specifici parametri. Si può specificare, per tipo di veicolo, i limiti di velocità, peso, larghezza e anche la presenza di un senso unico.

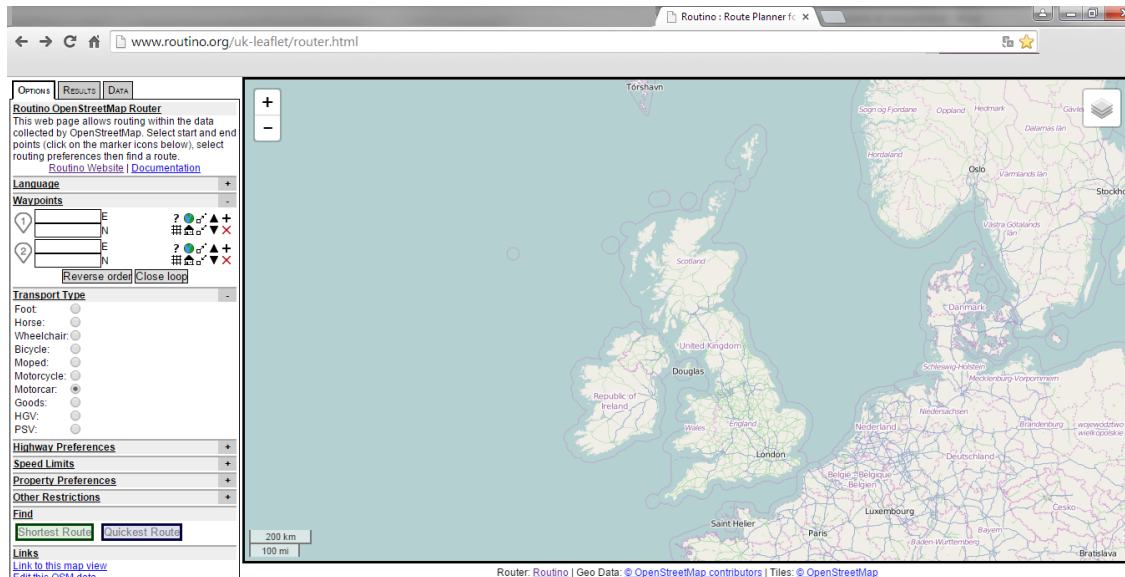


Figura 2.2.1 Routino HomePage

Nella figura 2.2.1, è possibile vedere che l'unica versione disponibile online di Routino, può essere utilizzata solo per il calcolo di un percorso in Inghilterra[RROUTINO2015]. Tramite la scelta del “Transport Type” è possibile decidere per che tipo di persona o veicolo si vuole calcolare il percorso. Oltre al “Transport Type” è presente anche il tipo di percorso che si desidera scegliere, cioè se si preferisce il percorso più breve (quindi con meno chilometri) oppure quello più rapido (che può avere più chilometri ma che risulta più veloce in minuti). Nel calcolo del percorso per il profilo wheelchair, verranno considerate tutte le barriere architettoniche già presenti nel database di openstreetmap. Routino può anche essere scaricato gratuitamente [OL2015] ed essere installato nel proprio computer ed essere utilizzato, insieme a Mapnik (che fornisce le immagini della mappa), per calcolare il percorso di una qualsiasi nazione si voglia.

2.2.2 OpenRouteService

È un sito online specifico per il calcolo di un percorso per i seguenti profili: autovetture, pedoni, biciclette e wheelchair. È basato sui standard OpenStandars[OS2015] e OpenGeodata[WOG2015]. OpenRouteService è molto più di un servizio di routing: utilizza una vasta gamma di servizi basati su dati OSM e che potrebbero anche essere utilizzati in altre applicazioni o scopi. Attualmente i seguenti

servizi sono stati implementati all'interno della struttura di OpenRouteService e sono basati su dati OSM.

- Il *Directory Service* è un servizio che fornisce l'accesso a una cartella online per trovare la posizione o un punto vicino, di un determinato luogo, prodotto o servizio.
- Il *Location Utility Service* offre un servizio di *Geocoder/Reverse Geocoder*; il *Geocoder* trasforma il nome di un luogo, come ad esempio il nome di una città, l'indirizzo di una via o un codice postale, in una descrizione normalizzata della localizzazione tramite un insieme di punti, più comunemente, trasforma i dati di input in coordinate.
- Il *Route Service* determina i percorsi e le indicazioni stradali a seconda del criterio selezionato. Questo servizio è stato realizzato per calcolare il percorso dei seguenti profili: auto (il percorso più breve o il più veloce), biciclette (ogni tipo), pedoni, *wheelchair*.

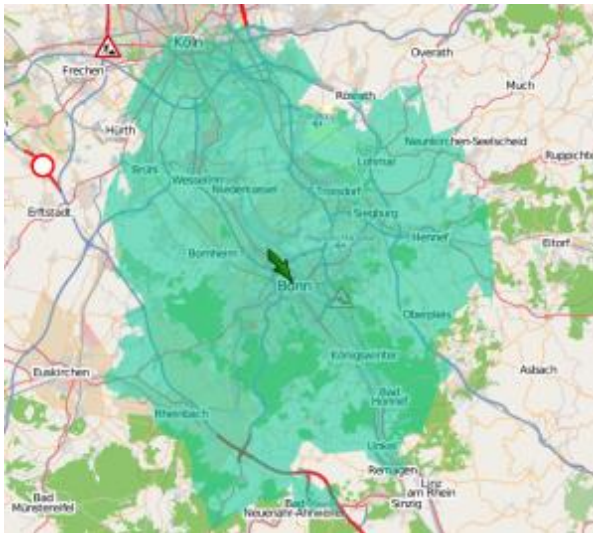


Figura 2.2.2.1 Accessibility Analysis

- L'*Accessibility Analysis Service* (AAS) calcola un poligono, come mostrato nella Figura 2.2.3.1, che rappresenta l'area che è raggiungibile tramite una rete stradale, con un certo intervallo di tempo, da un determinato luogo [WOSMORS2015].



Figura 2.2.2.2 Emergency Route Service

- Tramite l' *Emergency Route Service*, si può selezionare l'area/le aree da evitare per il calcolo di un percorso, come mostrato in *figura 2.2.3.2*.

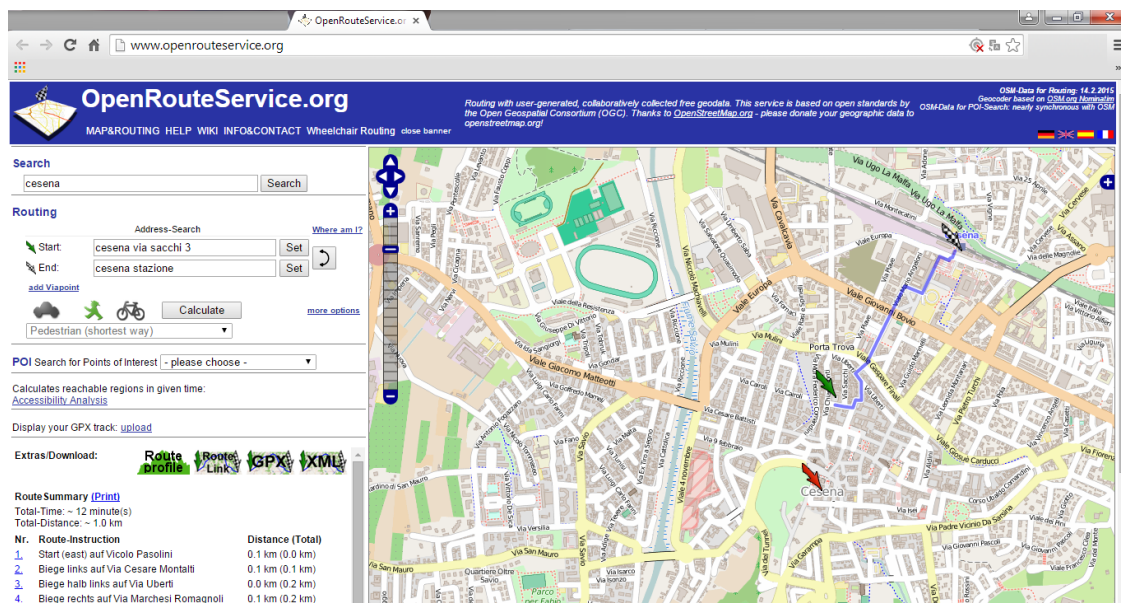


Figura 2.2.2.3 OpenRouteService.org

Come si può vedere dalla schermata principale del sito, presente in figura 2.2.3.3, se si vuole calcolare il percorso specifico per il profilo wheelchair bisogna cliccare sul pulsante “Wheelchair Routing” che ci renderizzerà al sito Rollstuhlrouting che è un’estensione di RouteService ma è specifico per il profilo wheelchair.

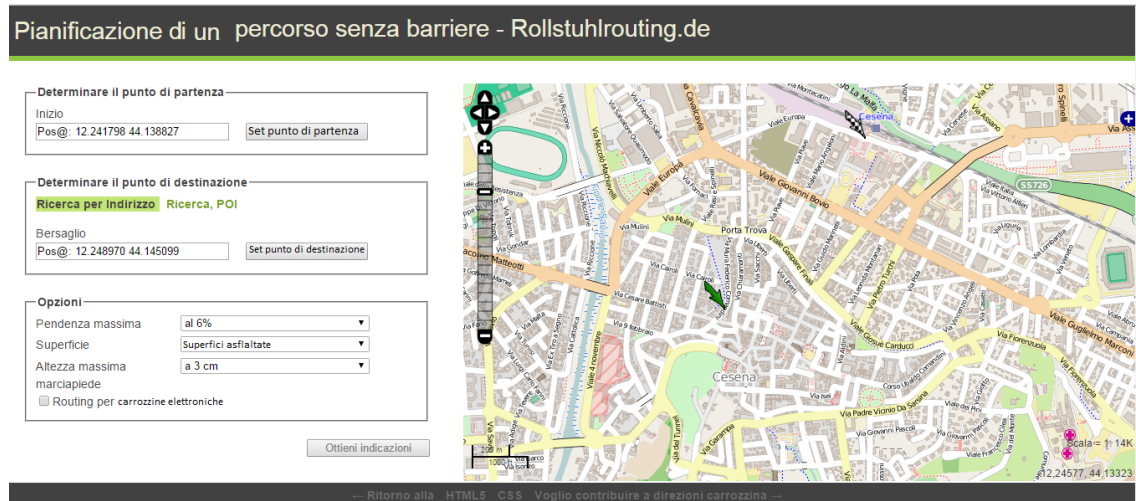


Figura 2.2.3.4 Rollstuhlrouting

Come potremo vedere nella figura 2.2.3.4, bisognerà selezionare un punto di inizio e un punto di fine per poter calcolare il percorso, il sito ci permette anche di selezionare altri dati che potrebbero tornarci utili, come ad esempio: massima pendenza del terreno, superficie (da preferire superfici asfaltate/senza ciottoli/tutte le superfici), altezza massima del marciapiede, routing per carrozzine elettriche (si può selezionare anche la velocità).

2.3 Software per mobile

I software mobile, come anche i desktop, possono essere a loro volta divisi in:

- **Offline:** Molti software mobile sono progettati per funzionare offline, infatti utilizzano il GPS per ricavare la posizione dell'utente e calcolare dinamicamente il percorso.
- **Inline:** i software che lavorano inline, utilizzano il GPS ma, tramite la connessione dati, permettono una geolocalizzazione più immediata.

La differenza tra i software mobile e i desktop software, sta nel modo in cui vengono elaborati i dati e nel loro utilizzo. Recentemente si preferisce l'utilizzo di software mobile, perché risultano sempre a "portata di mano", contrariamente a quelli desktop che richiedono l'utilizzo di un computer e, nella maggior parte delle volte, anche una connessione ad internet. L'utilizzo dei software mobile risulta anche molto utile quando si vuole avere in tempo reale le indicazioni per un determinato luogo, infatti il percorso viene calcolato dinamicamente anche a seconda degli spostamenti che

vengono effettuati. Di seguito vedremo alcuni esempi di software specifici per il profilo *wheelchair*.

2.3.1 Wheelmap – iPhone/Android



Figura 2.3.1 Wheelmap Application

Wheelmap è un software che mette a disposizione una mappa in cui sono mappate le barriere architettoniche specifiche per il profilo *wheelchair*. Esiste sia una versione desktop (e anche il sito online), che l'applicazione per iPhone e Android come presente in *Figura 2.3.1*. Entrambe permettono di sapere se un luogo d'interesse è accessibile o meno. È anche possibile ricercare e/o aggiornare i luoghi con maggiori informazioni.

L'applicazione, così come anche il sito internet sono disponibili nei seguenti linguaggi: Francese, Inglese, Tedesco, Spagnolo, Islandese, Italiano, Giapponese, Greco.

Il tag maggiormente utilizzato sia da OpenStreetMap che da Wheelmap, che contraddistingue il profilo wheelchair, è il tag *wheelchair="yes/no/limited"*.

Il tag *wheelchair="yes"* permette il totale accesso ad una zona, il tag "*limited*" permette un accesso parziale alla zona, ad esempio non si può accedere al bagno, invece il tag "*no*", non permette nessun tipo di accesso alla zona.

Oltre alla ricerca, si può anche contribuire attivamente al progetto tramite delle segnalazioni, sia in rete che con lo smartphone, dei luoghi che presentano degli ostacoli per una persona disabile in sedia a rotelle [WHEELMAP2015].

Tramite diversi tipi di bandierine colorate, si possono classificare gli elementi nel seguente modo, e come rappresentato anche nella *figura 2.3.2*:



Figura 2.3.2 Bandierine Wheelmap

- Bandierina verde: segnala che non ci sono problemi e che tutte le località sono raggiungibili per le persone disabili in sedia a rotelle, il gradino d'ingresso è più basso di sette centimetri ed è anche provvisto di bagno.
- Bandierina gialla: segnala che il luogo è parzialmente accessibile, sono raggiungibili solo i luoghi più importanti e l'ingresso ha un gradino inferiore a sette centimetri.
- Bandierina rossa: segnala che le località sono inaccessibili, quelle principali non sono raggiungibili e l'ingresso principale ha un gradino più alto di sette centimetri.
- Bandierina grigia: luogo d'interesse non ancora mappato, viene comunque mostrata la bandierina in modo tale da invogliare gli utenti ad assegnarli una bandierina specifica.

Oltre a classificare gli elementi, si può anche modificare la mappa, infatti dopo aver aggiunto un luogo si può anche specificare/modificare:

- il nome
- il tipo di accessibilità per il profilo wheelchair
- una descrizione sull'accessibilità, cioè specificare più nel dettaglio cosa potremmo trovare
- tipo di luogo: ad esempio se è un luogo di interesse, un bar, un museo ecc.

2.3.2 OpenTripPlanners

È uno sforzo di collaborazione tra: Trimet (l'agenzia di trasporto pubblico che opera in Portland, OR, USA) [Trimet2015], OpenPlans [OP2015], gli sviluppatori di FivePoints [FPDEV2015], OneBusAway [OBUSW2015], Graphserver [GSERVER2015], e anche altri sviluppatori indipendenti; per sviluppare un software open-source, che permette di calcolare un viaggio multimodale, cioè ad esempio utilizzando l'autobus e/o il treno, per avvicinarsi in modo più rapido e più pratico alla destinazione.

Caratteristiche principali:

- È progettato per utilizzare i dati di OpenStreetMap e il General Transit Feed Specification [GTFS2015].

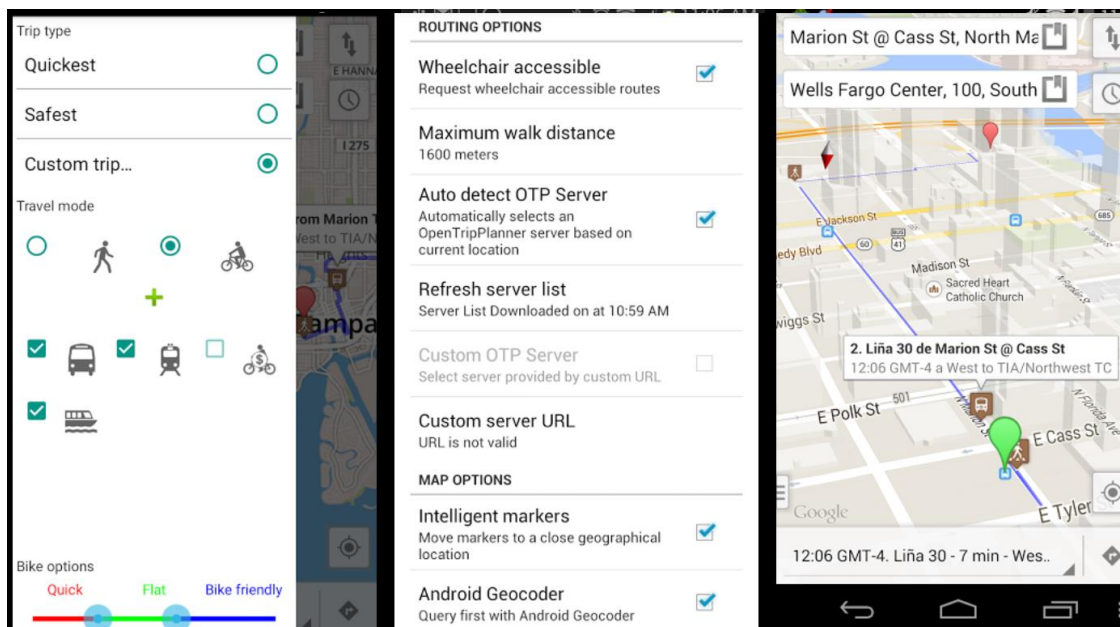


Figura 2.3.2 Scelte Applicazione

- Consente agli utenti di pianificare un viaggio che può utilizzare diverse modalità di trasporto (come mostrato nella figura 2.3.2), come andare in bicicletta oppure utilizzare diversi mezzi pubblici.
- I diversi tipi di percorsi da seguire per passare da un mezzo pubblico ad un altro.
- L'utente può specificare gli orari di partenza o di arrivo per un determinato trasporto pubblico.

- Si può specificare anche diverse opzioni di routing, come ad esempio il viaggio più corto, il viaggio più breve, il viaggio con il minor numero di utilizzo di trasporti pubblici.
- Accessibilità per le persone disabili.
- L'utilizzo del Geocoder per impostare la partenza e l'arrivo.

Per quando riguarda il design, OpenTripPlanner è costituito principalmente da tre software: un costruttore grafico, un modulo di routing e una interfaccia per l'utente. Mentre l'interfaccia utente principale (ad esempio, la mappa e la rappresentazione visiva del percorso) sono scritte in linguaggio *Javascript*, gli altri componenti sono scritti in Java. Il modulo di routing OpenTripPlanner è integrato in un Web Service API RESTful che può essere utilizzato con altre applicazioni personalizzate per il client. Il codice sorgente OpenTripPlanner è sotto la licenza GNU Lesser General Public License [LGPL2015], e può essere scaricata gratuitamente dal sito [OTP2015].

OpenTripPlanner utilizza due metodi diversi per il routing: l'algoritmo *A** [Algorithm2015] e il *Contraction Hierarchies* [CH2015]. Originariamente, OTP utilizzava solo il semplice algoritmo di *Dijkstra* e *A**, questi algoritmi però venivano eseguiti in modo molto lento su grandi grafi. L'algoritmo *Contraction Hierarchies* risultava quindi migliore per grafi molto grandi. L'idea base di questo algoritmo è che un grande grafo può essere contratto tramite la rimozione, passo per passo, di uno dei vertici e, sostituendo qualsiasi percorso, attraverso il vertice rimosso, con un collegamento che rappresenta tale percorso.

I tag di accesso (come ad esempio `bicycle/foot= "yes/no/designated"`) possono essere usati per escludere dei parametri nel calcolo del grafo e quindi del percorso. Come da default, non è consentito l'accesso, da parte di pedoni o biciclette, alle strade dove sono presenti i tag `highway= "trunk/motorway"`. Entrambi sono però ammessi nelle strade caratterizzate dai seguenti tag `highway= "pedestrian/cycleway/footway"`.

2.3.3 Altri mobile software

Principalmente i mobile software che utilizzano le mappe di OpenStreetMap e che considerano il profilo wheelchair, sono disponibili unicamente per iPhone e Android. Tra questi, oltre a quelli elencati sopra, troviamo:

- <20Mb, 512, 2 gb maps(iPhone) sono le mappe specifiche di alcune città disponibili per iPhone, iPod e iPad. Lo store di OpenStreetMap trasforma i dati in vettori e disegna la mappa che desideriamo, il contenuto della mappa è memorizzato in un file compresso così anche aree molto vaste possono essere memorizzate all'interno dell'applicazione [MBMAPS2015].

- OSMGpxUploader(Android): è un'applicazione che serve per caricare dei file GPX dalla propria sdcard nel server OSM. Si possono quindi inviare dei file dove all'interno saranno presenti dei vari pezzi di mappa con i rispettivi tag, come ad esempio i tag sull'accessibilità o meno di un luogo [GSTORE2015].

- Navigon Urban(iPhone): è un navigatore online prodotto da NAVIGON, una compagnia di Garmin. Questa applicazione utilizza i dati del database di OSM (solo quelli riguardanti la cartina della Germania) ed è ottimizzato per lo spostamento dei pedoni e dei ciclisti in città. Il "Pedestrian Routing" ha una guida vocale che può essere utilizzato dalle persone non vedenti [OSMNU2015].

3 Il profilo wheelchair in GraphHopper

In questo capitolo verranno descritte le modifiche che abbiamo apportato al progetto GraphHopper in modo che potesse calcolare un percorso anche per il profilo wheelchair.

3.1 GraphHopper

GraphHopper è un software open source di routing veloce ed efficiente scritto in Java e rilasciato con licenza Apache [GRAPH2015]. Fornisce sia la parte server, che quella client disponibile sia per il desktop e le applicazioni per Android e iOS. Utilizza diversi algoritmi come per calcolare il percorso più veloce come Dijkstra, A* e Contraction Hierarchies.

La mappa di GraphHopper utilizza un servizio chiamato *Directions API for Business*, per calcolare un percorso, con le relative informazioni necessarie per andare da un punto A ad un punto B, attraverso una richiesta http. Oltre a ciò, per mostrare la mappa, utilizza dei tiles prendendoli da Lyrk, il tutto è disponibile gratuitamente e, tramite https, quindi con una maggiore sicurezza.

Il sito online, permette il calcolo di un percorso solo per i profili “pedone/autovettura/bicicletta”, come mostrato in *figura 3.1*, senza però considerare il profilo *wheelchair*.

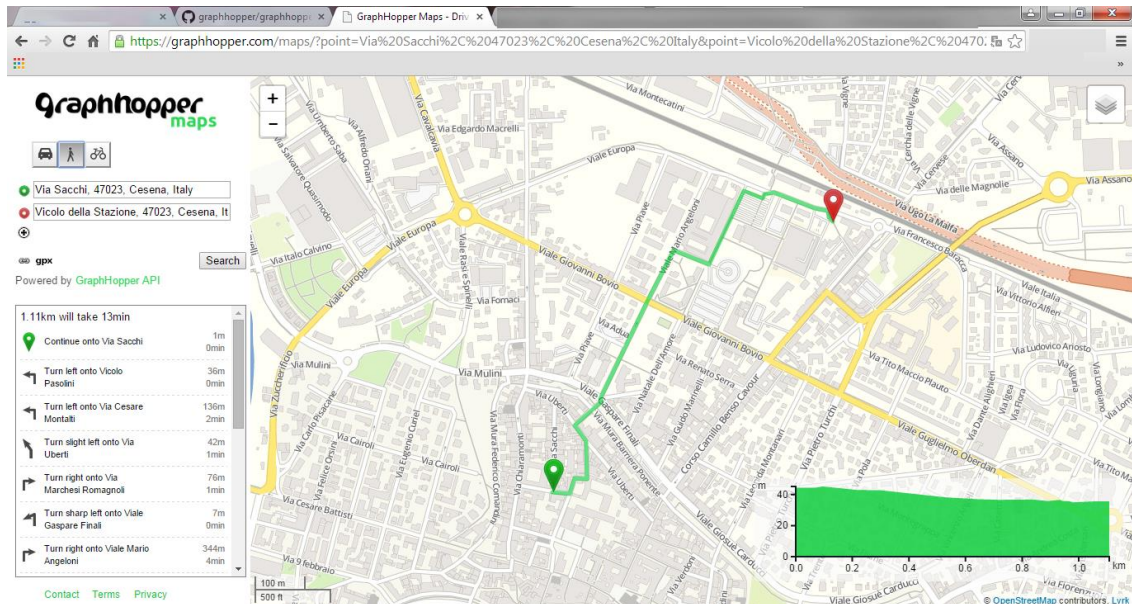


Figura 3.1 GraphHopper Website

Come vedremo in seguito, il software verrà poi modificato per permettere il calcolo di un percorso anche per il profilo wheelchair.

3.1.1 GraphHopper versione desktop

È possibile scaricare gratuitamente tutto il codice di GraphHopper da repository disponibile su GitHub [GitHGraph2015] e installarlo nel proprio computer.

A seconda del sistema operativo nel quale viene installato, cambierà anche il modo con cui avviarlo. Ad esempio per i sistemi linux, bisognerà farlo partire da terminale seguendo alcuni semplici steps. Per prima cosa, bisogna posizionarsi nella cartella dove è presente il progetto di GraphHopper (tramite il comando da terminale “`cd nomecartella`”).

Verrà poi usato il comando “`./graphhopper.sh web cesena.osm`” che è così costruito:

- la stringa “`./graphHopper.sh web`” prenderà la cartella dove è contenuto il progetto utilizzabile tramite web.
- `cesena.osm` invece rappresenta il file `.osm` che è stato esportato da OpenStreetMap e contiene tutti i dati relativi all’ambiente urbano di Cesena.

Verrà quindi eseguito il progetto e da terminale avremo le seguenti schermate disponibili nella figura 3.1.1.1:

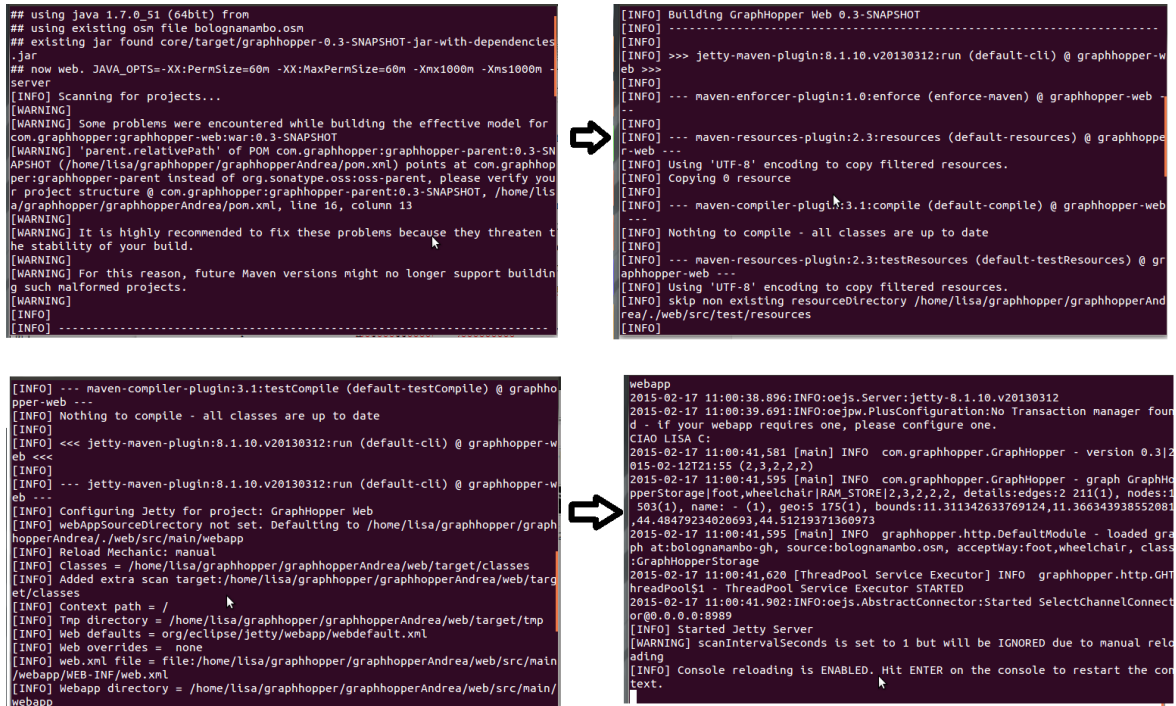


Figura 3.1.1.1 esecuzione del progetto GraphHopper

Dopo aver eseguito il progetto, si potrà visualizzare quest'ultimo in locale, infatti basterà andare in localhost:8989 (8989 è il numero della porta) per poter visualizzare ed utilizzare GraphHopper.

Come si può vedere dalla figura 3.1.1.2 saranno disponibili due profili per il calcolo del percorso, il profilo pedone e il profilo *wheelchair*.

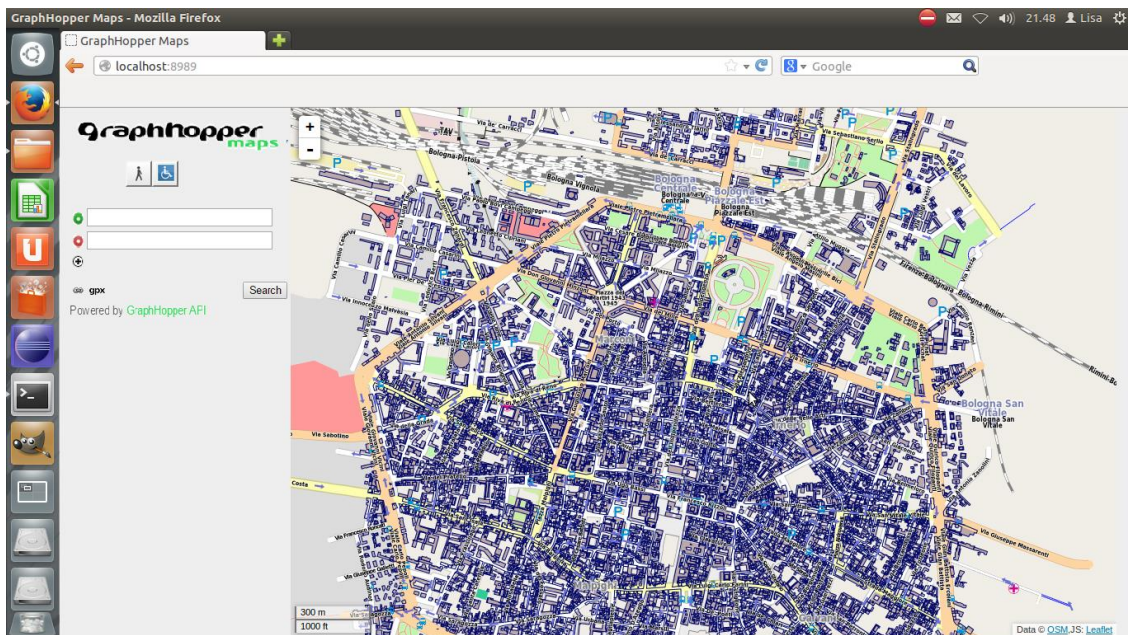


Figura 3.1.1.2 GraphHopper in local host

3.1.2 Come è strutturato

Come visto nella figura 3.1.1, GraphHopper utilizza:

- *Java*: infatti tutto GraphHopper è scritto in Java. Nella parte client per desktop vengono ovviamente utilizzati anche i linguaggi e le tecnologie tipiche del web come Javascript (HTML e CSS). L'utilizzo di Java come linguaggio base fa sì che il sistema possa essere eseguito in Linux, Mac OS X, Windows, Android, iOS ecc.
- *Maven*: è un software che usa un costrutto conosciuto come Project Object Model (POM) [WMAVEN2015]; un file XML che descrive le dipendenze fra il progetto e le varie versioni di librerie necessarie nonché le dipendenze fra di esse. In questo modo si separano le librerie dalla directory di progetto utilizzando questo file descrittivo per definirne le relazioni. Maven effettua automaticamente il download di librerie Java e plug-in Maven dai vari repository definiti scaricandoli in locale o in un repository centralizzato lato sviluppo. Questo permette di recuperare in modo uniforme i vari file JAR e di poter spostare il progetto indipendentemente da un ambiente all'altro avendo la sicurezza di utilizzare sempre le stesse versioni delle librerie [WMAVEN2015].
- *Jetty*: è un client/server HTTP unicamente basato su Java, client/server WebSocket e contenitore servlet sviluppato come software libero e progetto open source come parte di Eclipse Foundation. Jetty è progettato per essere incorporato in una applicazione Java, per fornirle tutti i servizi del protocollo HTTP che le siano necessari. Supporta AJP, JASPI, JMX, JNDI, OSGi, Socket Web e altre tecnologie Java [WJETTY2015].

Utilizza anche due server esterni quali:

- *Nominatim* per il geocoder, trova le coordinate relative ad una partenza/arrivo fornita in input.
- *Tile server* per le immagini della mappa. Se si vuole lavorare offline, si possono scaricare tutte le tiles della mappa, che risultano però molto pesanti.

Oltre a questo, GraphHopper utilizza diversi algoritmi per il calcolo del percorso che risultano molto efficienti e veloci in termini di prestazioni.

- *Contraction hierarchies*: è una tecnica per calcolare il percorso più breve precalcolando una versione contratta del grafo. CH è un estremo caso di approccio generalizzato, dove vengono generati dei nodi “multi-layered” in fase di preelaborazione. Ogni nodo del grafo è rappresentato dal suo livello nella gerarchia. Questo può essere ottenuto in diversi modi, la via più semplice è di numerare ogni nodo da 1 a n. Approcci più sofisticati potrebbero prendere in considerazione il tipo di strada (autostrada vs strada secondaria, ecc.).
- *A**: ("A Star" in inglese) è un algoritmo di ricerca su grafi che individua un percorso da un dato nodo iniziale verso un dato nodo destinazione. Utilizza una "stima euristica" che classifica ogni nodo attraverso una stima della strada migliore che passa attraverso tale nodo. L'algoritmo A* è anche un esempio di ricerca best-first.
- *Dijkstra e la versione bidirezionale*: è un algoritmo che ricerca il percorso più breve da un vertice iniziale ad una posizione finale. Compie due ricerche simultanee, la prima in avanti dal punto di partenza, e la seconda all'indietro dal punto finale, fino a quando queste due non si incontrano a metà del percorso.

Si può vedere dal codice 3.1.2, come l'algoritmo *CH* (ma anche gli altri) leggono tutti i nodi del file XML che collegano il punto di partenza e il punto di arrivo, in questo caso precisamente, inizieranno dal punto di partenza/di arrivo, per poi confrontare il miglior risultato, che sia il più breve o il più veloce.

```

void contractNodes ()
{
    meanDegree = g.getAllEdges().getMaxId() / g.getNodes();
    int level = 1;
    counter = 0;
    int initSize = sortedNodes.getSize();
    int logSize = (int) Math.round(Math.max(10, sortedNodes.getSize() / 100 * logMessagesPercentage));
    if (logMessagesPercentage == 0)
        logSize = Integer.MAX_VALUE;

    // preparation takes longer but queries are slightly faster with preparation
    // => enable it but call not so often
    boolean periodicUpdate = true;
    Stopwatch periodSW = new Stopwatch();
    int updateCounter = 0;
    int periodicUpdatesCount = Math.max(10, sortedNodes.getSize() / 100 * periodicUpdatesPercentage);
    if (periodicUpdatesPercentage == 0)
        periodicUpdate = false;

    // disable as preparation is slower and query time does not benefit
    int lastNodesLazyUpdates = lastNodesLazyUpdatePercentage == 0
        ? 0
        : sortedNodes.getSize() / 100 * lastNodesLazyUpdatePercentage;
    Stopwatch lazySW = new Stopwatch();

    // Recompute priority of uncontracted neighbors.
    // Without neighborupdates preparation is faster but we need them
    // to slightly improve query time. Also if not applied too often it decreases the shortcut number.

```

Codice 3.1.2 *ContractNode*

3.1.3 Richieste in locale o al server

Come visto in precedenza, gli algoritmi per il calcolo del percorso hanno bisogno di un nodo di partenza (inizio) e di un nodo di arrivo (fine) che vengono presi (indirettamente) dalla form.

Infatti i valori della form, vengono analizzati dalla funzione `mySubmit`: essa controllerà, che la form sia stata compilata in modo corretto (cioè non ci siano campi vuoti, o in quel caso di provvederà ad utilizzare i valori di default dei placeholder). Nel caso in cui i valori della form non corrispondano a quelli di default del placeholder significa che è stato immesso un valore e, quindi, verrà richiamata la funzione `resolveCoords` a cui verranno passati i valori `fromStr` e `toStr`.

La funzione `resolveCoords` prende in input i due valori sopracitati più una variabile “doQuery” che servirà poi per eseguire la query finale. I valori `from` e `to` verranno poi elaborati da `GHInput` che controllerà se i due valori sono stati inseriti come testo (indirizzo) o come coordinate.

- Nel caso di coordinate, verrà impostato un valore = “numero cifre della coordinata”, così da settare la latitudine e la longitudine in modo corretto, cioè da arrotondare, la latitudine dal primo termine della stringa iniziale fino alla numero di cifre stabilito, e la longitudine dal secondo termine fino al numero di cifre stabilito.

- Se non è un numero (`!isNaN`) significa che l’utente ha inserito un indirizzo, i due valori di l’input saranno convertiti come stringhe di testo e inviati, come parametri di richiesta, al server esterno di Nominatim, che fornirà poi la o le diverse coordinate a cui si riferiscono. Nominatim potrebbe fornire: più di una risposta, sia perché l’indirizzo inserito è troppo vago (ad esempio solo il nome della città senza cap) sia perché ci potrebbero essere omonimie, oppure potrebbe ritornare un errore nel caso in cui l’indirizzo non sia stato trovato.

La funzione `GHInput` passerà i due valori di coordinate alla funzione `Resolvefrom` (per l’indirizzo d’andata) e `ResolveTo` (per la coordinata di ritorno).

Le due funzioni di `Resolve`, ritorneranno un valore true e anche due marker nella mappa se la coordinata è stata trovata, altrimenti vi sarà un errore.

Le richieste al server.

Le principali richieste vengono gestite dal file `ghrequest.js` dove si possono: trovare

tutte le principali domande e risposte alle richieste http. Al suo interno, sarà presente la variabile *GHRequest*, che a sua volta sarà composta dalle variabili:

- *Host*: nome dell'host richiedente (in questo caso è in locale).
- *from*: la partenza selezionata/scritta presa dalla funzione *GHInput*.
- *To*: la destinazione, come sopracitato.
- *Vehicle*: il tipo di profilo selezionato.
- *Weightin*: il tipo di percorso selezionato, cioè il più veloce o il più breve.

Se non sono modificati tutti questi parametri sono settati di default a determinati valori, altrimenti verranno presi i valori inseriti dal richiedente.

La richiesta restituirà poi un "url" dove al suo interno saranno presenti:

- I dati richiesti in precedenza, che verranno aggiunti al url `localhost:8989/`
- le coordinate numeriche dei punti di partenza e di arrivo; tutto ciò grazie ad una richiesta a Nominatim, che ha trasformato i nomi dei punti di arrivo e di destinazione in coordinate numeriche.

3.2 I Tag per il profilo wheelchair

Per poter sviluppare un profilo wheelchair, bisogna prima di tutto considerare e quindi conoscere i vari tag che OpenStreetMap utilizza nei file contenenti le mappe. Come già detto in precedenza, i tag sono caratterizzati da due attributi, *key* e *value*. Per *key* si intende cosa rappresenta il valore, e *value* invece indica il valore che ha. Le due immagini che più rappresentano il tag d'accesso per il profilo *wheelchair*, sono le immagini 3.2.1, che sono ben visibili anche in diverse città.

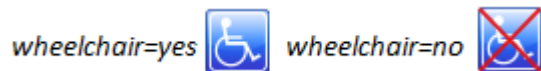


Immagine 3.2.1 Tag wheelchair

I tre valori (*value*) che forniscono informazioni sull'accesso per il profilo *wheelchair* sono:

- *Wheelchair="yes"* di solito viene usato per: ingresso senza scale, stanze senza scale, bagno (se presente) accessibile e con la cabina ampia abbastanza da contenere una sedia a rotelle vicino alla toilette, ecc.

- *Wheelchair="limited"* coloro che sono sulla sedia a rotelle hanno un accesso parziale ad alcune zone (ad esempio, ci possono essere delle zone in cui è necessario l'aiuto di un'altra persona per salire sopra un gradino ma poi l'ambiente risulta accessibile). Il tag si può riferire ad una zona con la seguente caratteristica: entrata con un gradino che supera i 7 cm d'altezza, stanze dove non sono presenti gradini, bagno (se presente) senza nessun accesso, ecc.

- *Wheelchair="no"* non si può accedere in nessun modo a questa zona(ad esempio a delle scale). Si può riferire anche a: entrate con uno scalino più alto di 7 cm, le stanze più importanti non sono accessibili, bagno (se presente) non accessibile, ecc. A questi va aggiunto il tag:

- *Wheelchair="designated"* : se vi è una zona o una via dove vi è l'intenzione d'installare delle particolari facility apposite per le persone che sono sulla sedia a rotelle (ad esempio un ascensore apposito). Questa coppia chiave-valore è però raramente utilizzata.

Per capire meglio quando questi tag vengono utilizzati, verranno spiegati, nel prossimo capitolo, tutti i servizi di maggiore importanza che utilizzano le limitazioni per il profilo wheelchair e tutte le informazioni aggiuntive che vengono utilizzate per specificare meglio il perché della limitazione oppure l'eventuale altro servizio offerto/disponibile.

3.2.1 Come devono essere etichettati i principali servizi

I principali servizi sono:

- Fermata dell'autobus:

Tag principale e 1 limitazione	Altre Limitazioni	Descrizione	Altro
Amenity=bus_stop wheelchair=yes : se il livello della pensilina è piano rispetto all'ingresso del bus o se tutti gli autobus hanno l'ascensore apposito a bordo.	wheelchair=limited :se i servizi sopra descritti sono presenti solo in alcuni casi o se è comunque necessario un aiuto esterno.	Si possono fornire più dettagli nel tag <i>wheelchair:description=*</i>	Può essere presente anche il tag <i>wheelchair=no</i>

Tabella 3.2.1.1 Fermata dell'autobus

- Bagno:

Tag principale e 1 limitazione	Altre Limitazioni	Descrizione	Altro
amenity=toilets wheelchair=yes : se sono perfettamente equipaggiate con corrimano, nessun gradino, porta ampia, ecc.	<ul style="list-style-type: none"> wheelchair=limited : se la porta è grande e senza gradini wheelchair=no , se la porta non è larga almeno 60cm e ci sono evidenti ostacoli per l'utilizzo. 	Si possono fornire più dettagli nel tag <i>wheelchair:description=*</i>	Non bisogna dimenticare eventuali orari di apertura/chiusura tramite il tag <i>opening_hours=*</i> .

Tabella 3.2.1.2 Bagno

Un esempio completo per aggiungere un tag di un bagno accessibile al profilo wheelchair(quindi con il tag “wheelchair=yes”) è presente nella tabella 3.2.1.3

Nome	Tag e Valore
Bagno	amenity="toilets"
Orario d'apertura	opening_hours="13:00-15:00"
Nome della via	Addr: street="name of road"
Numero (indirizzo)	Addr: house number="number"

Tabella 3.2.1.3 tag bagno

- Cinema:

Tag principale e 1 limitazione	Altre Limitazioni	Descrizione	Altro
amenity=cinema wheelchair=limited	nessuna	wheelchair:description =7 places (7 posti).	Se è comunque necessario un aiuto esterno, oppure se vi sono degli scalini solo in alcuni ingressi o se questi si possono varcare solo tramite dei gradini, si possono fornire più dettagli nel tag <i>wheelchair:description=*</i> .

Tabella 3.2.1.4 Cinema

- Hotel:

Tag principale e 1 limitazione	Altre Limitazioni	Descrizione	Altro
amenity=hotel <i>wheelchair=yes</i>	nessuna	<i>wheelchair:</i> <i>description=3</i> <i>rooms (3</i> <i>stanze)</i>	Se si vuole, si può anche indicare <i>website="www.example.com"</i> e <i>operator=*</i> se si dispone di informazioni più dettagliate.

Tabella 3.2.1.5 Hotel

- Museo:

Tag principale e 1 limitazione	Altre Limitazioni	Descrizione	Altro
tourism=museum <i>wheelchair=yes</i>	nessuna	<i>wheelchair:</i> <i>description=</i> <i>n=offre anche</i> <i>guide</i> <i>accessibili per</i> <i>sedie a rotelle</i>	Se si vuole, si può anche indicare <i>website="www.example.com"</i> e <i>operator=*</i> se si dispone di informazioni più dettagliate.

Tabella 3.2.1.6 Museo

- Ways (se ci sono scalini senza una rampa):

Tag principale e 1 limitazione	Altre Limitazioni	Descrizione	Altro
highway=steps <i>wheelchair=no</i>	nessuna	nessuna	Suggerimento: Bisogna sempre pensare alle pesanti sedie a rotelle elettriche che non possono superare nemmeno piccoli gradini.

Tabella 3.2.1.7 Ways

3.3 Le modifiche apportate a GraphHopper

Per poter soddisfare le nostre esigenze, si è dovuto cambiare molti aspetti di GraphHopper. I principali cambiamenti sono stati:

- Togliere i profili esistenti, e lasciare solo il profilo pedone
- Aggiungere, e quindi creare, il profilo *wheelchair*
- Apportare diverse modifiche al codice in modo tale da considerare il nuovo profilo e quindi le limitazioni/aggiunte che ne derivavano
- Permettere di aggiungere una barriera ad un nodo specifico presente in un file XML
- Ricalcolare il percorso con la barriera

Vedremo più nel dettaglio le operazioni elencate nei sottostanti paragrafi.

3.3.1 Come creare il profilo wheelchair

In GraphHopper, sono già presenti tre profili: pedone, auto, bicicletta; ma non era presente il profilo wheelchair che si è creato.

Per farlo si è dovuto creare una classe Java dove al suo interno vi fossero tutte le specifiche per il nuovo profilo. Nel file Java, sono presenti, oltre ad alcune librerie di Java, anche le librerie di GraphHopper relative alle `highway` (strade) e ai tag dei nodi per il calcolo del percorso.

I quattro valori principali che si sono settati sono visibili nel *codice 3.3.1.1*:

```
static final int SLOW=1;
static final int MEAN=5;
static final int FERRY=10;
private int safeWayBit=0;
```

Codice 3.3.1.1 Velocità e senso di marcia

`safeWayBit=0`, si intende che un *wheelchair* può andare in qualsiasi senso di marcia, quindi non vi sono limitazioni dovute ad un senso unico.

`Slow`, `Mean`, `Ferry` sono caratteristiche per la velocità e rappresentano in ordine, la velocità minima, la velocità media, la velocità massima.

Vi sono tre valori per la velocità in quanto, in una strada senza ostacoli e con una pavimentazione regolare, verrà applicata la velocità massima, nei casi in cui la pavimentazione e la strada è irregolare a tratti, verrà applicata una velocità media.

La velocità minima, o detta anche sconosciuta, verrà invece applicata nel caso in cui non si conosce bene, per via della mancanza di tag, la pavimentazione stradale, o nel caso in cui si selezionino sentieri di montagna e quindi più difficoltosi.

Oltre alla velocità, ci sono anche le direzioni che si possono prendere nel marciapiede, come presente nel *codice 3.3.1.2*. Sono ovviamente tutte, infatti si può, oltre ad accedere al marciapiede, decidere di stare in uno dei due lati della strada dove è presente il marciapiede (se presenti), girare a destra o a sinistra per poi immettersi in una nuova strada.

```
sidewalks.add("yes");  
sidewalks.add("both");  
sidewalks.add("left");  
sidewalks.add("right");
```

Codice 3.3.1.2 Direzioni

Bisogna poi indicare le restrizioni che caratterizzano l'accesso ad una determinata zona, tramite i due tag `access` e il tag `wheelchair` (ad esempio `access:wheelchair="yes/no/limited"`).

Le limitazioni d'accesso, comprendono anche i tag `private`, `no`, `restricted` tutti valori specifici di `restrictedValue`. Le limitazioni d'accesso possono essere dovute anche alle barriere architettoniche, bisognerà quindi aggiungere tutte le varie barriere architettoniche che riguardano il profilo `wheelchair`, tramite il comando `"potentialBarriers.add("tipobarriera")"`.

Oltre alle limitazioni d'accesso caratterizzate da barriere, vi possono anche essere limitazioni d'accesso per il profilo `wheelchair`, anche se non è direttamente presente il tag `wheelchair=no`. Le superstrade e le autostrade sono caratterizzate dal tag `highway = autocar`, che indica l'accesso solo ed esclusivamente ad autovetture, precludendo quindi l'accesso a tutti gli altri profili. Stessa cosa invece, in senso contrario, per le zone caratterizzate dal tag `pedestrian`, che caratterizza zone urbane, accessibili quindi oltre ai pedoni, anche al profilo `wheelchair`, tutto ciò è ben visibile nel *codice 3.3.1.3*.

```

@Override
public long acceptWay( OSMWay way )
{
    String highWayValue=way.getTag("highway");
    if(highWayValue==null)
    {
        if (way.hasTag("route", ferries))
        {
            String wheelchairTag=way.getTag("wheelchair");
            if(wheelchairTag==null||"yes".equals(wheelchairTag))
                return acceptBit|ferryBit;
        }
        return 0;
    }
    String sacScale = way.getTag("sac_scale");
    if (sacScale != null)
    {
        return 0;
    }
    if (way.hasTag("sidewalk", sidewalks))
        return acceptBit;

    if(way.hasTag("wheelchair", intended))
        return acceptBit;

    if(!allowedHighwayTags.contains(highWayValue))
        return 0;

    if (way.hasTag("motorroad", "yes"))
        return 0;

    if (way.hasTag("highway", "ford")||way.hasTag("ford"))
        return 0;
    if (way.hasTag("bicycle", "official"))
        return 0;

    if(way.hasTag(restrictions, restrictedValues))
        return 0;

    if (way.hasTag("railway")&&!way.hasTag("railway", acceptedRailways))
        return 0;
    return acceptBit;
}

```

Codice 3.3.1.3 Strade e accessi

Tutti i tag e le limitazioni, saranno prese in considerazione per il calcolo di un percorso a seconda del profilo scelto.

3.3.2 Lettura dati dal file XML

Per poter segnalare un'eventuale barriera e tenere in considerazione anche i vari tag del file *.osm*, si sono creati due file chiamati XMLopener e XMLcloser entrambi scritti in Java, che permettono di leggere un file *.xml*, e che cercano, nei vari nodi e tag della mappa, un tag di una barriera con `key="wheelchair"` e con `value=yes/no` così da poter elaborare questo dato nella fase del calcolo di un percorso.

Come visto in precedenza, se vi è un luogo dove il valore del tag *wheelchair* è *no*, significa che in quel luogo/via un profilo *wheelchair* non ha alcun modo di andare, e quindi bisogna calcolare un percorso alternativo.

Oltre alla lettura del file, si è aggiunto anche un comando per poter aggiungere un'eventuale barriera non segnalata in precedenza, in questo caso quindi, verrà aggiunto (se non già presente), dopo il tag *barrier* del nodo selezionato(*node*), il campo “*wheelchair=no*”, come si può vedere dal *codice 3.3.2.1*

```

else
{
    Element node=document.createElement("tag");
    node.setAttribute("k", "wheelchair");
    node.setAttribute("v", "no");
    nodeList.item(i).appendChild(node);
    modifyFile(args[0]);
    System.exit(WITHOUTATTRIBUTESCREATED);
}
}
}
try
{
    TransformerFactory transformerFactory=TransformerFactory.newInstance();
    Transformer transformer=transformerFactory.newTransformer();
    DOMSource domSource= new DOMSource(document);
    StreamResult streamResult=new StreamResult(args[0]);
    transformer.transform(domSource, streamResult);
} catch (Exception e)
{
    e.printStackTrace();
}

```

Codice 3.3.2.1

La stessa cosa può essere fatta per un'eventuale barriera non più presente. Il codice di XMLcloser sarà lo stesso di XMLopener, con l'unica differenza che ricercherà il tag “*wheelchair=no*” di un nodo specifico, e lo cambierà in “*wheelchair=yes*”. Si è dovuto importare i pacchetti *xml.sax* che sono dei pacchetti specifici per leggere i file *xml*.

Il problema della lettura dei file è che nel calcolo del percorso, viene prima creata una cartella contenente i nodi della mappa e poi viene calcolato il percorso. Quindi si è dovuto creare un file Java che, prima di calcolare il percorso, aggiungesse/rimosse la barriera, cancellasse la cartella contenente i nodi della mappa, ricreasse la cartella dei nodi della mappa con il tag modificato e mi ricalcolasse di nuovo il percorso.

La differenza del calcolo di un percorso con o senza barriera, lo si può notare dalle *figure 3.3.2.2 e 3.3.2.3*, nella prima infatti viene calcolato un percorso che risulterà uguale sia per il profilo pedone che per il profilo *wheelchair*, nella *figura 3.3.2.3* è stata aggiunta una barriera e quindi il calcolo del percorso per un profilo *wheelchair*, seppure minimo, risulterà differente.

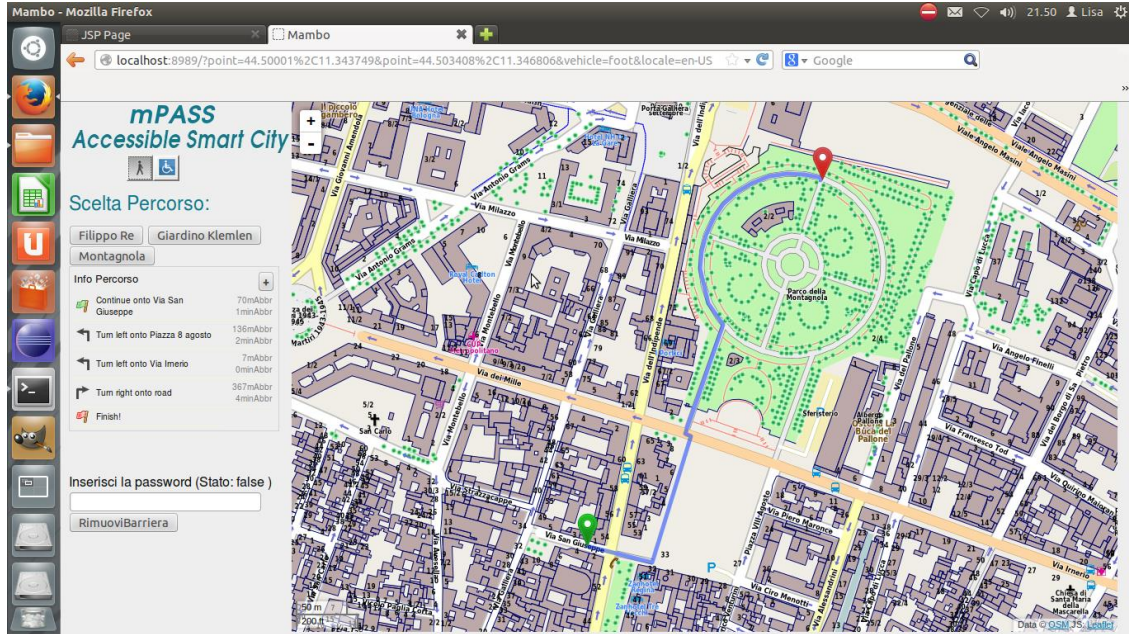


Figura 3.3.2.4 Calcolo percorso senza barriera

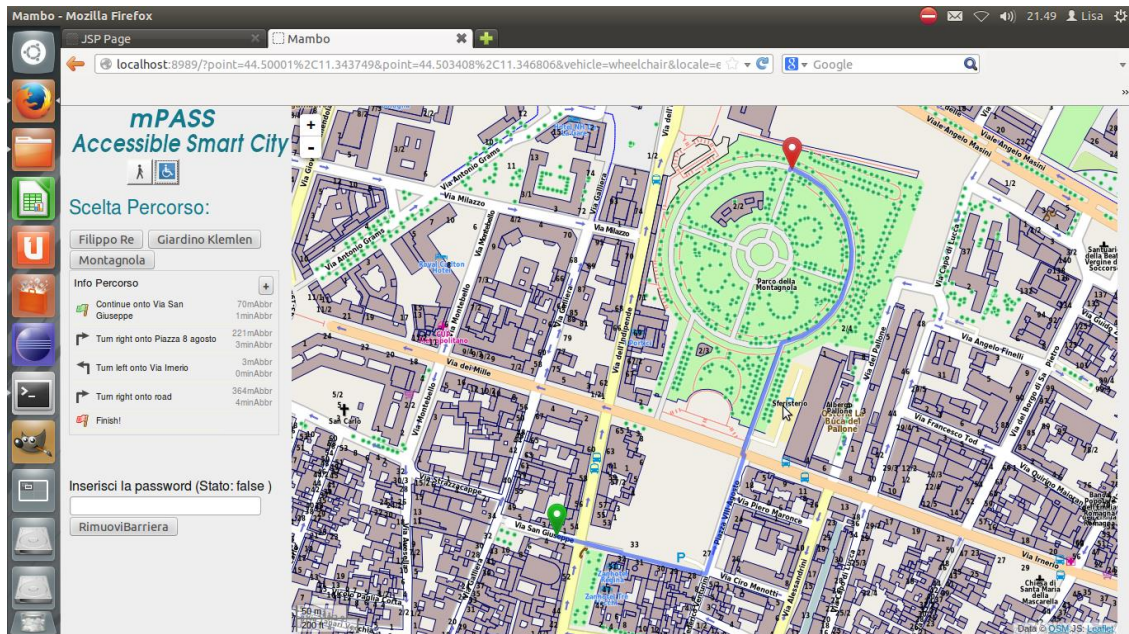


Figura 3.3.2.5 Calcolo percorso con barriera

Conclusioni

In questo lavoro di tesi sono stati introdotti i principali temi correlati con la community di OpenStreetMap e con il calcolo di un percorso per una persona con disabilità.

Si è proceduto effettuando un'introduzione sui termini e sui software che utilizzano le mappe presenti in OpenStreetMap per il calcolo di un percorso, oltre anche alle specifiche e ai termini caratteristici utilizzati per etichettare gli elementi presenti in una mappa.

Inoltre è stato illustrato il concetto di barriera architettonica e di accessibilità urbana, utilizzando degli esempi per ogni tipo di barriera presente e etichettata in OpenStreetMap.

Infine è stato analizzato il software di GraphHopper che è stato poi modificato per poter permettere il calcolo di un percorso per il profilo wheelchair.

Al momento i software che rendono disponibili il calcolo di un percorso per il profilo wheelchair, sono veramente pochi e molte volte sono specifici per alcuni paesi. Sono infatti presenti molti software che calcolano un percorso, con anche informazioni molto dettagliate sul percorso da seguire, ma sono tutti molto efficienti per le mappe della Germania oppure per alcune mappe dell'America. In questi ultimi due paesi sono stati organizzati anche diversi Mapping Party dove sono state raccolte informazioni molto dettagliate su molte città così da facilitarne lo spostamento. In Italia non sono ancora presenti molte informazioni sull'accessibilità urbana e ancora non sono state segnalate la maggior parte delle barriere architettoniche.

Con il calcolo di un percorso per il profilo wheelchair e con la rilevazione di una barriera architettonica si può aiutare attivamente tutte quelle persone che, in modo

permamente o temporaneo hanno delle disabilità motorie. Se ognuno di noi, potesse rilevare una barriera architettonica semplicemente camminando, senza quindi bisogno di ricordarsi sia il luogo che il tipo di barriera, sarebbe il modo più efficace e più immediato per segnalare tutte quelle barriere che prima non erano state segnalate, così da poter anche e eventualmente trovare un accordo con il comune di appartenenza per trovare una possibile soluzione al problema.

Bibliografia

- [WIK2015] Wikipedia, open street map,
<http://www.wikipedia.org/openstreetmap> , 2015
- [OSM2015] OpenStreet Map wiki
<http://wiki.openstreetmap.org/wiki/IT:About> , 2015
- [WPA2015] Walking Papers,
<http://walking-papers.org/> , 2015
- [WOS2015] Wiki Open Street Map, Traces:
<http://www.openstreetmap.org/traces> , 2015
- [WOSMMP2015]. Wiki Open Street Map, Mapping Party a Pompei
<http://wiki.openstreetmap.org/wiki/Pompeii> , 2015
- [WP2015] Wikipedia, GPS
https://it.wikipedia.org/wiki/Sistema_di_Posizionamento_Globale ,
2015
- [DEL2012] Luca Delucchi, Maurizio Napolitano, Alessio Zanol con il
contributo della Comunità italiana di OpenStreetMap,
“*Introduzione a OpenStreetMap*” Marzo 2012 pp 4-5
- [WOSMMAP2015] Wiki OpenStreetMap, Map Features
] http://wiki.openstreetmap.org/wiki/Map_Features , 2015
- [WOSMPROP2015] Wiki OpenStreetMap, Proposed Features
5] http://wiki.openstreetmap.org/wiki/Proposed_features , 2015
- [WOSMTYPES]. Wiki OpenStreetMap, i tipi di relazione
http://wiki.openstreetmap.org/wiki/Types_of_relation , 2015
- [JOSMWIKI2015] Wiki OpenStreetMap, il software JOSM
<http://josm.openstreetmap.de/wiki/It%3AWikiStart> , 2015

- [MAPNIK] Mapnik
<http://mapnik.org/> , 2015
- [WOSMOAR] Wiki OpenStreetMap, il software Osmarender
<http://wiki.openstreetmap.org/wiki/Osmarender> , 2015
- [MAPERITIVE2015] Maperitive
<http://maperitive.net/> , 2015
- [CARTO2015] Cartotype
<http://www.cartotype.com/> , 2015
- [MFORGE2015] MapsForge
<http://mapsforge.org/> , 2015
- [WOSMMW2015] Wiki OpenStreetMap, il software Map weaver
<http://wiki.openstreetmap.org/wiki/Mapweaver> , 2015
- [WOSMOR2015] Wiki OpenStreetMap, Osmrender
<http://wiki.openstreetmap.org/wiki/Osmrender.pl> , 2015
- [FOSM2015] Question OpenStreetmap
<http://forum.openstreetmap.org/viewtopic.php?id=7901> , 2015
- [OSM2W2015] Osm2World
<http://osm2world.org/> , 2015
- [WOSMSDM2015] Wiki OpenStreetMap, ShugenDoMap
<http://wiki.openstreetmap.org/wiki/ShugenDoMap> , 2015
- [OSM3D2015] OpenStreetMap 3D
<http://www.osm-3d.org> , 2015
- [MAPNIK2015] Mapnik
<http://mapnik.org/> , 2015
- [WOSMSOFT2015] Wiki OpenStreetMap, I desktop software
<http://wiki.openstreetmap.org/wiki/Software/Desktop> , 2015
- [ASN2015] Autosputnik
<http://autosputnik.com/> , 2015
- [SMANAGER2015] Spatial Manager desktop version
<http://www.spatialmanager.com/spm-desktop/> , 2015
- [SOZIALH2015] Sozial Helden
<http://www.sozialhelden.de/> , 2015
- [WHEELMAP2015] WheelMap
<http://wheelmap.org/> , 2015

- [ROUTINO2015] Routino : Router for OpenStreetMap Data
<http://www.routino.org/> , 2015
- [OL2015] OpenLayers
<http://openlayers.org/> , 2015
- [OS2015] Wikipedia OpenStandars
http://it.wikipedia.org/wiki/Standard_aperto , 2015
- [WOG2015]. OpenGeodata
<http://www.opengeodata.it/> , 2015
- [MBMAPS2015] Wiki OpenStreetMap, 20MB 512 MB, 2GB Maps
[http://wiki.openstreetmap.org/wiki/20MB, 512MB, 2GB Maps](http://wiki.openstreetmap.org/wiki/20MB,_512MB,_2GB_Maps) ,
2015
- [WOSMORS2015] Wiki OpenStreetMap, OpenRouteService
] <http://wiki.openstreetmap.org/wiki/OpenRouteService> , 2015
- [Trimet2015] Wikipedia, Trimet
<http://en.wikipedia.org/wiki/TriMet> , 2015
- [OP2015] OpenPlans
<http://openplans.org/> , 2015
- [FPDEV2015] Welcome to Fire and Police Development
<http://www.fpdev.org/> , 2015
- [OBUSW2015] One Bus a Way
<http://onebusaway.org/> , 2015
- [GSERVER2015] Github - graphserver
<http://bmander.github.io/graphserver/> , 2015
- [LGPL2015] GNU licenses
<http://www.gnu.org/licenses/lgpl.html> , 2015
- [OTP2015] Open Trip Planner
<http://openTripPlanner.org> , 2015
- [Algorith2015] Wikipedia, A*
http://en.wikipedia.org/wiki/A*_search_algorithm , 2015
- [CH2015] Wikipedia, Contraction hierarchies
http://en.wikipedia.org/wiki/Contraction_hierarchies , 2015
- [GTFS2015] Google developers, Transit
<https://developers.google.com/transit/gtfs/reference?csw=1> , 2015

- [GSTORE2015] Google Play Store, Osm Upload
<https://play.google.com/store/apps/details?id=net.anzix.osm.upload>
, 2015
- Wiki OpenStreetMap, Navigon Urban
- [OSMNU2015] http://wiki.openstreetmap.org/wiki/Navigon_Urban , 2015
- [GRAPH2015] GraphHopper
<https://graphhopper.com/> , 2015
- [GithGraph2015] Github, GraphHopper
<https://github.com/graphhopper/graphhopper/> , 2015
-