



simple

Strumenti e Modelli Per La mobilità sostenibile

R.2.3 Repository tool open source (aggiornamento nel mese M25)



UNIONE EUROPEA
Fondo europeo di sviluppo regionale



REPUBBLICA ITALIANA



REGIONE AUTONOMA DE SARDIGNA
REGIONE AUTONOMA DELLA SARDEGNA



SARDEGNA
RICERCHE

Progetto finanziato con fondi *POR FESR 2014/2020 - ASSE PRIORITARIO I*
"RICERCA SCIENTIFICA, SVILUPPO TECNOLOGICO E INNOVAZIONE".

INFORMAZIONI SUL PROGETTO

Numero del progetto	N/A	Acronimo	SIMPLE
Titolo completo	Strumenti e Modelli Per La mobilità sostenibile		
Soggetto	Progetto CLUSTER ICT		
Data inizio	01/02/2018		
Durata in mesi	30		
Coordinatore	UniCA – Università degli Studi di Cagliari		
URL del progetto	http://www.simple-cluster.it		

INFORMAZIONI SUL DOCUMENTO

Numero del Deliverable	2.2	Titolo	R.2.3 Repository tool open source (aggiornamento nel mese M25)
Numero del Workpackage	2	Titolo	Raccolta esigenze e animazione
Data di scadenza del deliverable	29/02/2020		
Data di sottomissione del deliverable	27/02/2020		
Autore/i responsabile/i	Lucia Pintor		
Livello di diffusione	Non applicabile		

MODIFICHE DEL DOCUMENTO

Data	Autore	Modifiche	Versione
17/01/2020	Lucia Pintor	Prima stesura del documento	v0.0
23/01/2020	Lucia Pintor	Correzioni e aggiunte minimali	v0.1
04/02/2020	Lucia Pintor	Capitolo riguardo il routing	v0.2
05/02/2020	Lucia Pintor	Conclusioni	v1.0
10/02/2020	Lucia Pintor	Correzione di alcuni dettagli	v1.1
27/02/2020	Luigi Atzori	Correzioni finali	v1.2

Tavola dei contenuti

Sommario	4
I prototipi	4
Beep4Me	4
PoolBus	4
I linguaggi di programmazione utilizzati	4
Python	4
Java	4
Swift	5
Gli strumenti open source utilizzati	5
Django	5
PostgreSQL	5
Open Trip Planner	5
Docker container	6
Nuovi tool inseriti dopo il mese M14	6
Librerie di routing Google OR-Tools	6
Conclusioni	6

1 Sommario

Il presente documento riporta tutti gli strumenti open source utilizzati per lo sviluppo dei due prototipi, includendo quelli già descritti nel primo repository tool. Il repository tool dei software open source è accessibile in formato online nel portale del progetto all'interno della sezione Risultati.

2 I prototipi

2.1 Beep4Me

Questo prototipo ha come obiettivo quello di incentivare gli utenti ad utilizzare il trasporto pubblico e supportare in modo semplice la ripartizione degli introiti (clearing) tra le aziende che forniscono i servizi di mobilità. Le sperimentazioni prevedono l'uso delle tecnologie Bluetooth e lo sviluppo di nuove funzionalità all'interno di applicazioni per tablet e smartphone. Il team SIMPLE sta sviluppando dei moduli da integrare ad applicazioni iOS e server Django.

2.2 PoolBus

Questo prototipo consentirà agli utenti di prenotare dei servizi su richiesta, in modo semplice e con costi contenuti. L'obiettivo è quello di sostenere quelle aree della Sardegna in cui il trasporto di linea tradizionale non è sufficiente per soddisfare le necessità della popolazione. Il prototipo sarà la base per un sistema completo che include diversi tipi di servizi di mobilità integrati: sharing, pooling, trasporto pubblico e servizi a chiamata. Attualmente sono in fase di sviluppo delle applicazioni Android e dei moduli server Django.

3 I linguaggi di programmazione utilizzati

3.1 Python

Python è un linguaggio di programmazione interpretato, di alto livello e generico. Creato da Guido van Rossum e pubblicato per la prima volta nel 1991, Python ha una filosofia di design che enfatizza la leggibilità del codice. Python supporta diversi paradigmi di programmazione, inclusi object-oriented, imperativo, funzionale e procedurale. Questo linguaggio è usato lato server per sviluppare applicazioni Web con Django e Django Rest Framework.

Viene utilizzato l'ambiente di sviluppo Pycharm di JetBrains: www.jetbrains.com/pycharm

3.2 Java

Java è un linguaggio di programmazione general-purpose basato su classi, orientato agli oggetti, e specificamente progettato per avere il minor numero possibile di dipendenze di implementazione. Si tratta di un linguaggio che una volta compilato può essere eseguito su tutte le piattaforme che supportano Java, senza necessità di ricompilazione. Il linguaggio Java è utilizzato anche per sviluppare applicazioni Android.

Viene utilizzato l'ambiente di sviluppo Android Studio di Google: developer.android.com/studio

3.3 Swift

Swift è un linguaggio di programmazione general-purpose, multi-paradigma e compilato sviluppato da Apple Inc. Questo linguaggio è nato dall'esigenza di semplificare e rendere più sicuro lo sviluppo di applicazioni iOS, che precedentemente venivano scritte in linguaggio Objective-C.

Viene utilizzato l'ambiente di sviluppo Xcode di Apple Inc. : developer.apple.com/xcode

4 Gli strumenti open source utilizzati

4.1 Django

Django è un framework Web Python di alto livello che agevola notevolmente lo sviluppo Web, fornendo diversi pacchetti e moduli adattabili a qualunque applicazione. Utilizza la filosofia DRY (Don't Repeat Yourself): in questo modo si realizzano funzioni e comportamenti generici che consentono di evitare ripetizioni, rendendo il codice più snello. Il codice può essere sviluppato velocemente e in maniera modulare, in modo che ciascuna funzione possa essere utilizzata e aggiornata indipendentemente dalle altre. Inoltre dispone del toolkit Django Rest Framework, che semplifica la creazione di API (*Application Programming Interface*) e servizi web complessi e altamente personalizzabili.

Link: www.djangoproject.com, www.django-rest-framework.org

4.2 PostgreSQL

PostgreSQL è un potente sistema di database relazionale a oggetti open source con oltre 30 anni di sviluppo attivo che gli è valso una solida reputazione per affidabilità, robustezza delle funzionalità e prestazioni. PostgreSQL è dotato di molte funzionalità volte ad aiutare gli sviluppatori a creare applicazioni. Oltre ad essere gratuito e open source, PostgreSQL è altamente estensibile. Segue la filosofia ACID (Atomicità, Consistenza, Isolamento, Durabilità) e consente l'accesso multiutente. PostgreSQL inoltre può essere combinato con le librerie PostGIS per gestire oggetti spaziali.

Link: www.postgresql.org, postgis.net

4.3 Open Trip Planner

OpenTripPlanner (OTP) è una famiglia di progetti software open source che forniscono servizi di analisi delle reti di trasporto e dei passeggeri. Il componente Java core sul lato server trova gli itinerari che combinano i segmenti di transito, pedone, bicicletta e auto attraverso reti basate su dati OpenStreetMap e GTFS standard aperti e ampiamente disponibili. È possibile accedere a questo servizio direttamente tramite la sua API Web o utilizzando una vasta gamma di librerie client. Non è compatibile con molte versioni di Java, però è possibile risolvere il problema utilizzando un container per OTP con una versione Java supportata.

Link: www.opentripplanner.org

4.4 Docker container

Docker offre agli sviluppatori la libertà di creare, gestire e proteggere le applicazioni evitando gran parte dei problemi dovuti all'incompatibilità del software. Docker consente di creare dei container che contengono un'applicazione specifica e tutte le sue dipendenze, in modo che l'applicazione funzioni rapidamente e in modo affidabile da un ambiente di elaborazione a un altro. Un'immagine di container Docker è un pacchetto software leggero, autonomo e eseguibile che include tutto il necessario per eseguire un'applicazione. Esiste un repository ufficiale di immagini Docker già pronte (Docker Hub).

Link: www.docker.com , hub.docker.com

5 Nuovi tool inseriti dopo il mese M14

In seguito ad alcuni incontri con i partner si è scelto di modificare alcuni dettagli dei due prototipi, in particolare le scelte di sviluppo del prototipo PoolBus hanno richiesto l'utilizzo di nuove librerie di routing specifiche per risolvere Vehicle Routing Problems.

5.1 Librerie di routing Google OR-Tools

OR-Tools è un software open source dedicato alla ricerca operativa, che include una libreria di routing specializzata nel risolvere i problemi di instradamento dei veicoli, in cui l'obiettivo è trovare percorsi efficienti per il trasporto di persone o merci attraverso una rete complessa.

Il prototipo PoolBus utilizza l'algoritmo di routing VRPTW (Vehicle Routing problems With Time Windows) di questa libreria, aggiungendo la clausola che ciascun drop-off deve avvenire in seguito al pick-up corrispondente. Il risolutore cerca il percorso ottimale per soddisfare tutte le richieste di passaggio all'interno delle finestre temporali associate a ciascuna fermata. I dati iniziali sono la matrice di origine destinazione, una lista di finestre temporali, il numero di veicoli disponibili e l'indicazione del deposito.

La libreria contiene i wrapper per i linguaggi Python, Java e C#, che consentono di usare in modo semplice e intuitivo la libreria anche in linguaggi diversi da quello nativo.

Link <https://developers.google.com/optimization/routing/vrptw>

6 Conclusioni

L'avanzamento dello sviluppo dei prototipi implica continue lievi modifiche per adattarsi alle esigenze del progetto. Esse in alcuni casi sono soddisfatte dagli strumenti già utilizzati, in altri casi invece no, per cui è necessario cercare nuovi tool specifici.

Il prototipo Beep4Me si appoggia su software dei partner per la bigliettazione smart (piattaforma Teseo di CTM). Il modulo sviluppato continua ad utilizzare i software open source indicati nel precedente deliverable "R.2.3 Repository tool open source", ovvero le librerie Swift per lo sviluppo iOS e il framework Django

abbinato a un database PostgreSQL per lo sviluppo lato server. Le applicazioni lato server sono virtualizzate tramite container Docker.

Il prototipo PoolBus invece è un sistema completo che ha la possibilità di integrarsi con altre applicazioni di pooling e sharing tramite API. Il prototipo continua ad utilizzare i software open source indicati nel precedente deliverable “R.2.3 Repository tool open source”, ovvero le librerie Android per lo sviluppo mobile e il framework Django abbinato a un database PostgreSQL e ad un’istanza di Open Trip Planner per lo sviluppo lato server. Le applicazioni lato server sono virtualizzate tramite container Docker.

In seguito alle ultime riunioni di progetto si è deciso di modificare il servizio su richiesta gestito interamente dal sistema PoolBus: per avere una maggiore flessibilità di orari e di percorsi si è deciso di utilizzare algoritmi di routing e, dopo un’attenta analisi, è stata selezionata la libreria di routing di OR-Tools VRPTW (Vehicle Routing problems With Time Windows).