

## 1. RICONOSCIMENTI

Ringrazio il mio relatore Fabio Casati, il mio correlatore Cris-  
tian Parra e LifeParticipation<sup>1</sup> per avermi seguito durante  
tutto il percorso dell’internship che ha portato a questa tesi  
e al lavoro da me fatto; ringrazio anche Manuel Cattin Cosso  
e chi mi ha sopportato nei momenti di stress di quest’anno.

## APPENDIX

### A. IL BOOKLET COME APPLICAZIONE PER IL TESTING

Nella sezione dedicata alla soluzione è stato presentato bre-  
vemente il Booklet, un’interfaccia pensata per svolgere una  
delle funzioni di base di Reminiscens, cioè quella di mostrare  
contenuti che possano risultare familiari all’utente in modo  
da stimolare l’afflusso di ricordi e far partire la narrazione di  
episodi correlati al contenuto presentato. Tale applicazione  
è stata intesa per permettere, come sviluppo futuro, alle  
famiglie degli anziani la personalizzazione del booklet, ag-  
giungendo ad esempio immagini e musica di propria scelta;  
il tutto per riuscire a creare un “aggregatore di ricordi” che  
sia il più efficace possibile. Come anticipato in sezione ??, la  
scelta da noi compiuta è quella di offrire un’interfaccia che  
riprende un libro antico, con una grossa copertina in pelle e  
il titolo dorato, il tutto con uno sfondo che rappresenta la  
superficie in legno massiccio di un vecchio tavolo, in modo  
da mostrare un ambiente familiare che possa essere il primo  
passo per andare oltre il gap che indubbiamente esiste tra gli  
anziani e le nuove tecnologie; proprio per come è stata pen-  
sata, la UI si presta perfettamente ad un’attività preliminare  
di testing. Per verificare la bontà delle scelte di design, così  
come l’adeguatezza di alcuni dati raccolti dai moduli ETL,  
si potrebbe pensare ad un workshop avvalendosi della colla-  
borazione di un centro ricreativo per anziani, strutturato in  
questo modo: dopo aver diviso i collaboratori in gruppi, ad  
ogni gruppo verrebbe assegnato un iPad, con a bordo una  
versione del booklet con contenuti diversi; l’idea è quella di  
comporre ogni piattaforma di test con contenuti riguardanti  
un diverso periodo della vita, tenendo conto delle diverse età  
dei partecipanti e ottenendo quindi infanzia, giovinezza, età  
adulta e presente. Ultimo parametro da considerare nella di-  
visione dei gruppi e nella composizione dei booklet è quello  
del luogo in cui centrare il calcolo del contesto da presentare  
ai volontari: perchè il workshop possa restituire dei risultati  
utili, c’è bisogno che ognuno possa vedere immagini, ascol-  
tare musica e ricordare eventi che almeno in teoria possano  
essere significative (e.g. una persona cresciuta a Roma dif-  
ficilmente potrà rievocare memorie passate guardando delle  
fotografie del Monte Bondone). Il feedback ricevuto da que-  
sta e altre prove è indispensabile per capire se la strada che  
stiamo percorrendo con Reminiscens è quella giusta e, in  
caso di difetti del nostro approccio, fornirebbe importanti  
linee guida per correggerlo.

## B. DOCUMENTAZIONE API

A seguito del lavoro compiuto, viene qui rilasciata la do-  
cumentazione delle API REST che si occupano di costruire  
un contesto a partire dalle entità contenute nella Knowledge  
Base. Oltre a questo, sono state scritte delle route che per-  
mettono anche la semplice lettura dei dati dalla KB, utiliz-  
zando delle query più complesse; l’obiettivo è quello di ren-

- **lat[]**, **lon[]** : due array di coordinate geografiche, Lati-  
tudini e longitudini devono avere nome rispettivamente  
*lat[]* e *lon[]* - invece di *lat* e *lon* - per far sapere all’API  
che si tratta di due array.

mentre la terza accetta solo il primo, cioè *decade[]*.

### B.2 Lettura dei dati

- **GET** /media
- **GET** /events
- **GET** /people
- **GET** /works

Le prime tre route accettano i seguenti parametri:

- **decade**: una decade nella forma *dddd*.
- **place** : è la forma letterale rappresentante il luogo;  
quando usato, è cercato nel db; se assente, viene ef-  
fettuato il Geocoding per ottenere le coordinate cor-  
rispondenti; può essere associato a *radius*; se usato  
insieme a *lat* e *lon*, viene ignorato.
- **lat** e **lon** : ovviamente latitudine e longitudine; devono  
sempre apparire in coppia; possono essere associate a  
*radius*; se utilizzate insieme a *place*, quest’ultimo viene  
ignorato.
- **radius** : rappresenta la distanza nella da *place* / *lat*  
e *lon* che l’API deve utilizzare per considerare validi o  
no i possibili risultati; di default ha valore 0.

mentre l’ultima accetta solo il primo, cioè *decade*.

## C. SOLUZIONE

Partendo dai paragrafi precedenti, riusciamo ora a delineare quelle che sono le possibili componenti software da costruire per i nostri scopi. La prima e indispensabile è un indice di risorse storiche, da consultare per ottenere le entità più pertinenti alla vita di un individuo; proprio per decidere cosa è attinente e cosa no, un algoritmo deve essere pensato e implementato; affinché il risultato dell'implementazione sia scalabile e flessibile, deve essere scelto un meccanismo di accesso aperto e utilizzabile in più frangenti; ultima, ma non meno importante, la visualizzazione dei dati raccolti e filtrati, che devono essere presentati in una forma user-friendly, con attenzione particolare agli utenti di una certa età, che, rappresentando una fetta consistente dei possibili utilizzatori, richiedono una progettazione più attenta.

Nella realizzazione di un sistema capace di rispondere all'utente con dati relativi al tempo e al luogo in cui egli ha vissuto alcuni momenti della sua vita, la prima difficoltà è riscontrabile nella scelta riguardo quale meccanismo utilizzare per recuperare le entità con cui costruire il contesto; per ottenere questo risultato le possibilità si possono ricondurre a tre approcci principali:

- Sfruttare i linked data per recuperare a runtime le informazioni necessarie: questo è una scelta ambiziosa, che pone molta fiducia nel fatto di riuscire a scavare nel modo giusto tra i datasets per trovare abbastanza elementi da restituire, elementi che siano anche inerenti alla richiesta fatta.
- Utilizzare i linked data, collegandoci un altro dataset, formato da informazioni raccolte da noi: il problema che vogliamo risolvere riguarda la storia delle persone, e chi meglio delle altre persone può aiutarci a raggiungere questo obiettivo? per questo un grosso aiuto nella ricerca di immagini, eventi e altro potrebbe venire da un opera di crowdsourcing che, immagazzinata sotto forma di entità con un significato e, collegata ad altre sorgenti di dati, andrebbe a completare i nostri bisogni e forse quelli di altri utilizzatori del semantic web.

Quest'ultima è la proposta di soluzione più interessante, che si tuffa nel futuro di una rete nella quale informazioni da sorgenti completamente diverse tra di loro vanno a completarsi e impreziosirsi a vicenda, ma che per la natura sperimentale del nostro lavoro va purtroppo considerata solo un possibile sviluppo futuro. Per poter essere meno dipendenti dalle specifiche RDF e SPARQL, certamente ancora in evoluzione, l'approccio che abbiamo scelto è di per sé il più semplice: perchè non costruire un archivio storico in forma di database, indicizzarlo in maniera corretta e renderlo disponibile tramite un algoritmo di creazione di un contesto intorno a delle coordinate spazio-temporali? Così nasce Reminiscens, il cui funzionamento e struttura sono spiegati qui sotto.

### C.1 Architettura generale

Reminiscens è un sistema strutturato di varie parti, che lavorano insieme per fornire numerosi servizi. Le funzionalità di raccolta e memorizzazione dati sono svolte da un database, dei moduli software e un'interfaccia grafica: Il database è la componente fondamentale di Reminiscens, contenente tutto

il materiale utilizzato dal sistema per restituire un contesto (di cui si parlerà più avanti) all'utente; perchè la Knowledge Base contenga dei dati, questi devono essere raccolti da qualche parte, nel nostro caso il lavoro è fatto da dei moduli ETL, ognuno dei quali si occupa di interrogare una determinata risorsa sul web (e.g. DBpedia); la stessa funzione è svolta da CrowdMemories, una UI sviluppata da Francesco Maturi che utilizza il crowdsourcing per reperire materiale storico direttamente dagli utenti. L'accesso ai dati invece, è implementato tramite delle API con architettura REST, che svolgono quindi la funzione di intermediari nella comunicazione tra i client e la Knowledge Base. La parte di visualizzazione consiste attualmente in una semplice interfaccia in forma di libro sfogliabile, a cui nel futuro si affiancheranno altri servizi; questi ulteriori sviluppi verranno trattati alla sezione ??.

### C.2 Il mio lavoro

Durante il mio lavoro, mi sono trovato a seguire i dati per tutto il percorso, dalla loro raccolta alla loro visualizzazione, passando per tutta la fase di elaborazione. Di seguito ecco le parti che ho sviluppato:

#### C.2.1 Moduli ETL

L'implementazione da noi scelta è quella di un insieme di moduli ETL (Extract, Transform, Load) standalone che raccolgono ed elaborano dati, dipendenti quasi solo da un modulo che si occupa della comunicazione diretta con la Knowledge Base. La loro importanza è vitale perchè, oltre all'ovvia funzione di recuperare gli elementi che andranno poi restituiti all'utente in forma di contesto, il hanno anche il compito di ricostruire dove possibile le coordinate spazio-temporali dei dati estratti (e.g. tramite il Geocoding di una locazione scritta in maniera testuale). Essendo la lista delle risorse consultabili sul web sempre in crescendo, questa scelta è stata fatta per rendere semplice l'aggiunta e la sostituzione dei moduli, che attualmente supportano la consultazione di Flickr, DBpedia, Catinabib (catalogo di cartoline e incisioni a tema regionale di proprietà della biblioteca comunale di Trento) e di dataset del progetto OpenData Trentino. Proprio grazie a quest'ultima risorsa, si spera che con il passare del tempo, grazie all'aggiunta di nuovi datasets, le informazioni contenute nella Knowledge Base e di conseguenza fornite all'utente possano crescere di quantità e di qualità. Indice: Per permettere una lettura veloce dei dati raccolti, il risultato delle estrazioni dal web viene periodicamente indicizzato, così che non sia necessario leggere tutto il db per ottenere un risultato da spedire ai client visuali.

#### C.2.2 Api REST

Per permettere la consultazione della KB senza dover interrogarla direttamente, ho sviluppato delle semplici api che, tramite messaggi HTTP, permettono di fare ricerche a partire da parametri spazio-temporali. Queste si dividono in due tipi:

- Nel caso sia necessario semplicemente leggere una lista di elementi dal KB, un insieme di chiamate si assicura che al richiedente torni la giusta lista di tutte le immagini o gli eventi o altro combacianti i parametri di input.

- Se invece si tratta della comunicazione con i client, altre chiamate vanno a leggere l'indice per restituire un contesto. Questo argomento è descritto in maniera specifica alla sezione D.2.5.

Per questioni più tecniche, nell'appendice ?? è disponibile la documentazione delle api.

### C.2.3 Simple CRUD

Affiancato ai moduli ETL e alle api è presente un piccolo e semplice sistema con interfaccia minima che effettua CRUD, per aggiungere e modificare manualmente il materiale del db.

### C.2.4 Booklet

Un'ultima parte riguarda il lavoro per comporre un'interfaccia che potesse essere utile a visualizzare i dati restituiti dalle API, indipendente dai client di Reminiscens: chiedendo all'utente una decade e un luogo, quello che viene mostrato è un libricino sfogliabile tramite browser, e contenente un contesto completo in una forma che possa essere familiare a chi lo guarda, soprattutto nel caso dei meno giovani, che mancano di dimestichezza con i mezzi più tecnologici. Il libro restituito è però un insieme di dati impersonali, in quanto per avere un risultato personalizzato la scelta giusta è il vero e proprio client di Reminiscens. Il caso d'uso del Booklet come applicazione per il testing è descritto nell'appendice ??, mentre per una prova pratica è disponibile online una versione della web app<sup>3</sup>.

### C.2.5 Algoritmo LifeContext

Come definito sopra, il contesto è un insieme di immagini, eventi, personaggi e musica che hanno a che fare con il tempo e il luogo di cui l'utente potrebbe voler scoprire qualcosa, e che potrebbe aiutarlo a raccontare della sua vita. La nostra scelta è stata quella di comporre questo contesto utilizzando cinque entità per tipo, selezionate secondo alcuni criteri all'interno dell'indice. Partendo dalla knowledge base, la Lifeincontext API crea un contesto intorno ad una lista di eventi. L'elaborazione si divide due casi principali, all'interno dei quali si snodano diversi sottocasi:

- se le coppie lat-lon possono essere racchiuse all'interno di un cerchio virtuale di raggio 50 km:
  - se la maggior parte delle decadi è uguale, viene creato un unico contesto a partire da una delle coppie lat-lon e da quella decade. questo si può ottenere con una singola query che, ordinando i risultati per pertinenza, seleziona direttamente i primi cinque;
  - se il numero degli eventi di input è minore di tre, viene creato un contesto componendo dei “sotto-contesti”, uno per ogni coppia formata dalle decadi di input e da una delle coppie lat-lon;
  - l'ultimo caso è un'estensione del secondo, infatti, a partire da una lunga lista di coppie, viene creato un contesto componendo tre sotto-contesti

tra quelli possibili; questo viene fatto per non inquinare troppo il risultato, che senza il filtraggio appena indicato sarebbe una semplice accozzaglia di elementi che in ultima analisi sarebbero poco correlati tra di loro;

- se le coppie non soddisfano la condizione, vengono proposti il secondo e il terzo sotto-caso indicati sopra, con la differenza che ogni contesto viene computato a partire dalle coppie così come indicate in input, senza elaborazioni aggiuntive.

Oltre al calcolo a partire dalle sole indicazioni spazio-temporali, un'ulteriore versione dell'algoritmo è stata pensata per cercare, filtrare e personalizzare i risultati ottenuti anche in base a delle parole chiave, siano esse argomenti, generi musicali, interessi dell'utente e oggetto della risorsa: ad esempio, una richiesta riguardante Trento nel decennio 1980, con keyword “calcio”, dovrebbe restituire tra le altre una voce sul campionato di calcio serie C2 1984-85, quando la squadra locale venne promossa in Serie C1 dopo uno spareggio vinto contro l'Ospitaletto. Grazie a queste poche informazioni, fornite al sistema dall'utente in fase di configurazione del profilo, l'API dovrebbe consegnare del materiale interessante ed evocativo di momenti del suo passato, stimolando così il racconto di episodi della sua vita.

## C.3 La Knowledge Base

Certamente buona parte della riuscita di questo lavoro è dipesa dalla scelta di modellazione per la Knowledge Base di Reminiscens, per questo abbiamo puntato su uno schema semplice e facilmente modificabile: ogni tipologia di entità (eventi, immagini, lavori artistici come musica e libri, personaggi famosi) è mappata su una singola tabella, e possiede una relazione con due tabelle che contengono i dati riguardanti luogo e date associati a ogni elemento e una con la tabella che funge da indice. Figura ?? mostra in breve l'architettura del nostro database.

Durante la raccolta delle risorse con i moduli ETL, ci siamo resi conto che le coordinate spazio-temporali erano in forme molto diverse, e contenevano dati con precisione diversa: per questo abbiamo introdotto abbiamo costruito la Knowledge Base con i concetti di *Fuzzy Date* e *Fuzzy Location*, cioè date e luoghi che possono essere più o meno definiti (e.g. una data contiene l'anno ma non il giorno, mentre un'altra contiene solo la decade), classificandole tramite due scale di valutazione. Non ci siamo preoccupati troppo di ottenere solo entità con informazioni precise per due semplici motivi: (1) avremmo ovviamente dovuto scartarne molte, e (2) passando tramite una fase preliminare di pulizia automatica - durante la raccolta - e un'altra di pulizia manuale, riusciremo a rendere consistenti e semplici da cercare gran parte dei contenuti.

## C.4 Framework e librerie

Per realizzare queste componenti, mi sono avvalso di diverse tecnologie, elencate qui sotto per sezione.

- Moduli ETL e Indice: Tutti i moduli sono realizzati in Java utilizzando Apache HttpComponents per

<sup>3</sup><http://test.reminiscens.me/lifecontext/booklet/ui/>

gli scambi di messaggi HTTP, Hibernate, Apache Jena per poter interrogare tramite SPARQL, le API di Google Maps per risolvere tutti i problemi relativi alla risoluzione sia di coordinate in luoghi che il contrario, Google Gson per il parsing di messaggi JSON e le API di Youtube per associare un video musicale a ogni canzone nella KB. Riguardo SPARQL, in figura ?? sono mostrate due semplici query utilizzate dal sistema per interrogare DBpedia e ottenere rispettivamente canzoni ed eventi sportivi con dati in lingua italiana.

- Api REST: la scrittura di queste librerie mi è stata facilitata dall'utilizzo di Ruby unito al framework Sinatra, con l'ausilio di ActiveRecord e del linguaggio di templating RABL. Al fine di eseguire alcuni calcoli complessi, ho eseguito il porting di uno script per la conversione di coordinate geografiche in UTM<sup>4</sup> e ho adattato un'implementazione Ruby dell'algoritmo con complessità lineare elaborato da Megiddo per il calcolo del Minimum Enclosing Circle di un set di punti<sup>5</sup>.
- CRUD: Questa è un'applicazione Ruby on Rails che si avvale di una semplice UI per la modifica manuale dei dati, ma che permette eseguire le stesse operazioni programmaticamente, inviando e ricevendo JSON.
- Booklet: L'interfaccia del booklet si basa su "20 things I learned about browsers and the web"<sup>6</sup>, applicazione web HTML5 (sviluppata dal team di Google Chrome e resa open source) riadattata per le nostre necessità.

## C.5 Stato della Knowledge Base

Per questa prima parte di sviluppo, anche pensando a possibili fasi di testing, abbiamo voluto concentrarci sulla raccolta di materiale relativo alla zona del Trentino, con maggiore attenzione alle zone di Trento, Rovereto e Pergine Valsugana; nella realtà dell'implementazione, questo significa che, dove possibile, i dati sono stati filtrati per luogo, con alcune eccezioni: per quanto riguarda la musica ad esempio, abbiamo fatto la scelta di cercare anche musica straniera, data la facilità con cui un abitante locale può venire a contatto con canzoni che non provengono dall'Italia. Al tempo della scrittura di questo documento, la Knowledge Base di Reminiscens contiene 3244 entità, tra immagini, canzoni, eventi e personaggi famosi. Tabella ?? mostra quante di queste risorse appartengono a ogni tipo, mentre tabella ?? mostra la provenienza delle stesse. Entrambe le tabelle mostrano che, nonostante il lavoro da noi compiuto sia ancora in uno stato di sviluppo, (1) è stato molto più semplice raccogliere elementi di tipo musicale (un piccolo progetto che ha lavorato con la musica come terapia è illustrato brevemente nella sezione ??), (2) Il web semantico e il crowd si dimostrano due fonti di valore inestimabile per gli scopi che ci siamo posti, soprattutto considerando il fatto di averle sfruttate in maniera non intensiva; i dati qui mostrati sono solo uno degli input che è necessario analizzare approfonditamente per scegliere quale strada tracciare per il futuro di Reminiscens.

---

<sup>4</sup><http://home.hiwaay.net/~taylorc/toolbox/geography/geoutm.html>

<sup>5</sup><http://www.dseifert.net/code/mec/index.html>

<sup>6</sup><http://www.20thingsilearned.com/>