



brain matching srls

via foscolo 60, 39012 merano (bz)

p.iva/cod.fisc: 02950980215

www.brainmatching.com

brainmatching@pec.it

**Definizione dei flussi e delle responsabilità di gestione e sviluppo dell'applicazione
(COD. FESR 2016)**

PROGETTO MERIDIA

REPORT ATTIVITA' 3 e 4 | v.02 | 25.09.2020

Responsabile:

Roberto Colonello

roberto.colonello@brainmatching.com

Indice

1. Introduzione
2. Piattaforme ed infrastrutture
3. Struttura dei dati
4. Struttura della skill
5. Flussi dei dati
6. Pubblicazione della skill
7. Conclusioni

1. Introduzione

Google, Amazon ed Apple hanno creato un nuovo modo di usufruire delle informazioni, non più tramite mouse o schermo, ma bensì tramite voce: gli assistenti vocali.

Un assistente vocale è un software presente sul cellulare o in dispositivi appositi forniti di microfono ed altoparlante, che permette una interazione tramite il parlato. E' quindi possibile chiedere informazioni all'assistente parlando come se fosse una persona. Questo è stato possibile grazie alla forte evoluzione che c'è stata negli ultimi anni della potenza di calcolo dei computer e dal forte sviluppo del Cloud.

Quanto detto all'assistente vocale viene mandato ad un servizio cloud dove la voce viene trasformata in testo, il testo viene analizzato e viene generata la risposta, che poi viene letta tramite voce dall'assistente vocale, il tutto in pochi attimi.

Il presente progetto prevede l'uso dell'assistente vocale di Google per l'interazione con i dati del comune di Merano. La scelta è ricaduta su Google in quanto fornisce l'ambiente di sviluppo più completo e permette di gestire simultaneamente più lingue.

I software sviluppati da parte di terzi che arricchiscono le conoscenze dell'assistente vocale ricadono sotto il nome di "Actions" o "Skills".

Le attività 3 e 4 di questo progetto prevedono la pianificazione, strutturazione e sviluppo dell'action per Google Home che permette di consultare i dati del Comune di Merano.

I dati messi a disposizione sono:

- Struttura del Comune di Merano
- RSS delle notizie del Comune di Merano
- API REST JSON delle farmacie di turno fornite dalla Provincia Autonoma di Bolzano
- API REST JSON dei dati dei musei forniti dalla Provincia Autonoma di Bolzano
- Pagine web degli eventi a Merano catturate tramite un crawler

2. Piattaforme ed infrastrutture

Il linguaggio di programmazione utilizzato è PHP.

PHP viene usato dal 78% dei siti web su internet come linguaggio server side, quindi è stata la scelta naturale per permettere di creare una infrastruttura fruibile anche da altri comuni italiani.

Inoltre PHP può funzionare su ogni tipo di piattaforma: linux, windows, osx.

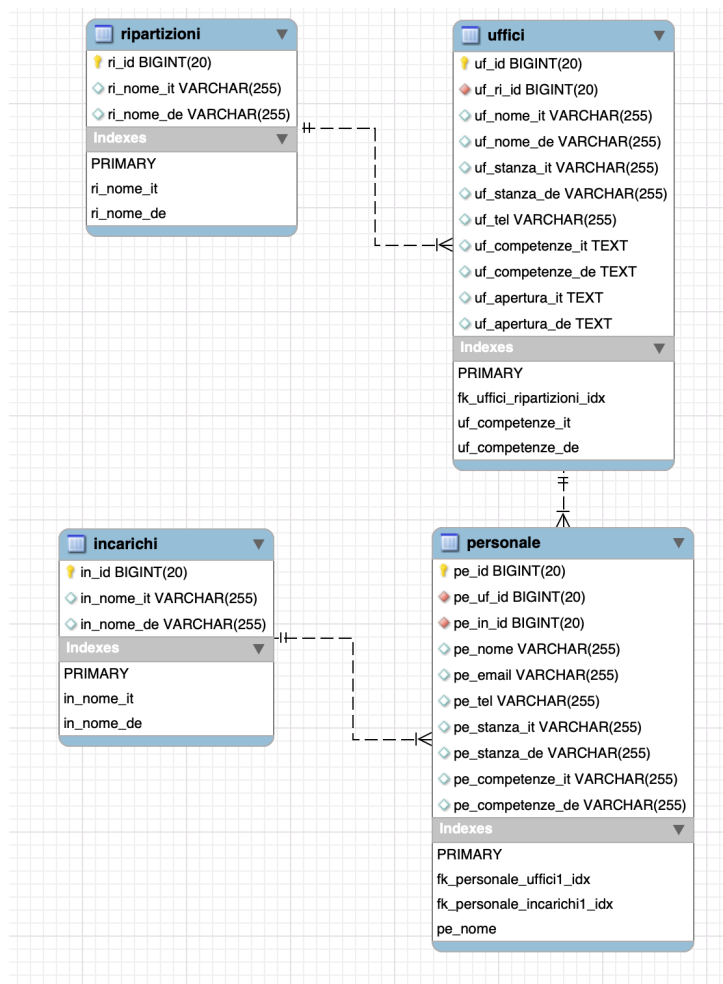
Il sistema di riconoscimento del linguaggio naturale invece viene gestito dalla piattaforma di Google Dialogflow, che rende accessibile in maniera intuitiva algoritmi di intelligenza artificiale.

Come database per l'archiviazione delle informazioni è stato scelto MySQL in quanto Opensource, di semplice gestione e viene usato dalla maggior parte dei siti online.

Il sistema quindi richiede: registrazione (gratuita) sulla piattaforma Actions Google e Dialogflow Google, un server su cui far girare il modulo PHP Webhook ed un database MySQL.

3 – Struttura dei dati

I dati dell'organizzazione comunale vengono memorizzati in questa struttura nel database MySql:



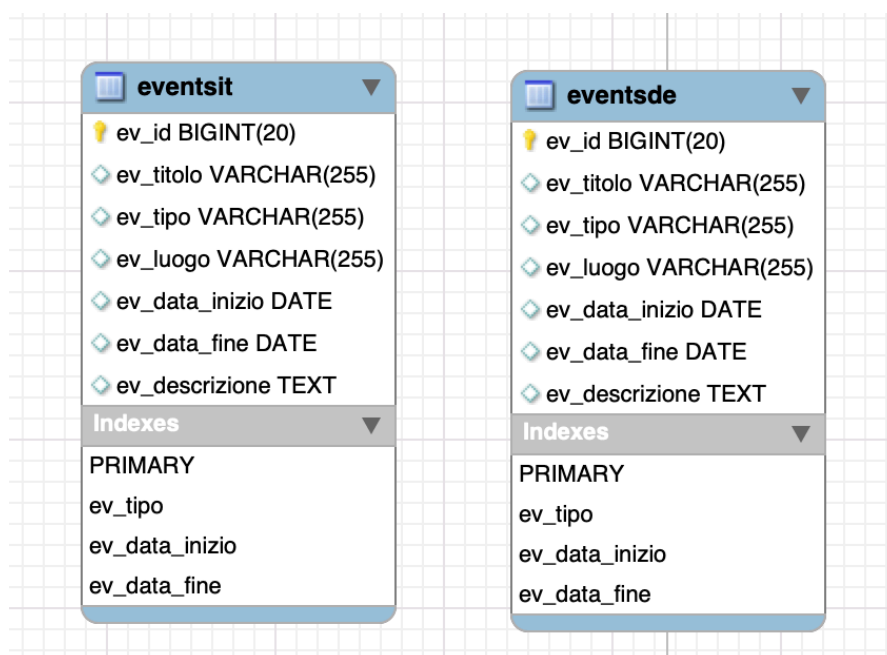
Ogni persona fa parte di un ufficio ed ha un incarico. Ogni ufficio è parte di una ripartizione.

Gli indici sono stati creati per i campi che vengono usati nelle query per l'ordinamento o il filtro.

La ricerca del nome di una persona può creare problemi in quanto i nomi in tedesco non sempre vengono riconosciuti correttamente dalla action italiana, e viceversa, i nomi in italiano non sempre vengono riconosciuti correttamente dalla action in tedesco. Per ovviare a questo problema abbiamo usato la funzione SQL Soundex che permette di effettuare una ricerca sul suono di una parola invece che sulla scrittura di una parola.

Gli eventi vengono mantenuti in 2 tabelle, una per gli eventi in italiano, una per quelli in tedesco.

Ogni evento è formato da un titolo, una tipologia (musica, arte, cucina,...), da un luogo, dalle date di inizio e fine e descrizione.

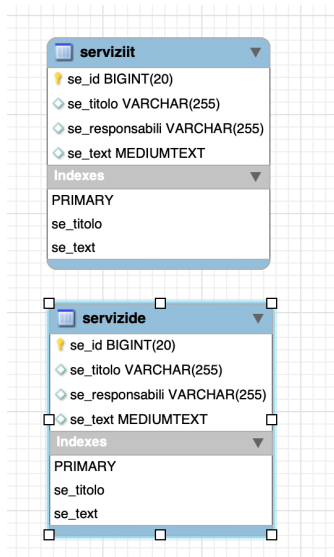


I dati vengono catturati, trasformati e caricati in queste tabelle tramite un software crawler realizzato ad hoc, che prende le informazioni dal sito web in formato HTML, ne riconosce le parti e le memorizza in queste tabelle.

Le informazioni presenti sul sito del Comune di Merano, sotto la voce “il comune dalla A alla Z” contengono documenti su come accedere ai servizi offerti. Questi dati sono stati caricati in 2 tabelle (una in italiano ed una in tedesco) e possono essere accessibili chiedendo “dammi informazioni su”. Esempi di dati disponibili sono “mense”, “taxi rosa”, “cittadinanza”, “carta d’identità”...

Essendo i possibili argomenti i più variegati, abbiamo dovuto trovare una soluzione per creare un ranking di importanza delle risposte possibili, in quanto più testi possono corrispondere alla parola/ frase richiesta.

I testi sono quindi stati memorizzati in un campo di tipo MediumText e sono stati indicizzati con un indice di tipo FullText. Questo ha permesso di usare una query con un campo di risultato “score” che si basa sulla frequenza e presenza di un certo tag nel testo: `SELECT se_titolo, se_text, MATCH(se_text) AGAINST (“__FRASE DA CERCARE__”) as score FROM serviziit ORDER BY score desc`



Le notizie del comune vengono invece elaborate in tempo reale. Quando avviene una richiesta, viene preso il file RSS contenente le notizie direttamente dal sito del comune e le informazioni vengono riorganizzate e

presentate sotto forma vocale. Il formato è RSS XML e ogni record contiene un titolo ("title"), un link ("link"), una descrizione in formato HTML ("description") e la data di riferimento ("pubDate"). Il contenuto del titolo e della descrizione vengono letti da Google Home.

Anche la richiesta della farmacia di turno viene elaborata in tempo reale prendendo l'informazione dal servizio Opendata REST API JSON della Provincia di Bolzano: <http://daten.buergernetz.bz.it/services/pharmacy/v1/json?>

I dati sono in formato JSON e contengono le seguenti informazioni:

- DATE (data)
- PHAR_DESC_D (nome in tedesco della farmacia)
- PHAR_DESC_I (nome in italiano della farmacia)
- PHAR_ADRESS_D (indirizzo in tedesco)
- PHAR_ADRESS_I (indirizzo in italiano)
- PHAR_PHONE (numero di telefono)
- GEME_DESC_D (città in tedesco)
- GEME_DESC_I (città in italiano)
- GEME_ZIP (cap)
- PHAR_TIMETABLE (orario di apertura)
- TURN_TIMETABLE (orario di turno)
- IS_TURN (0=non è di turno, 1=è di turno)
- TURN_STATUS_D (descrizione del turno in tedesco)
- TURN_STATUS_I (descrizione del turno in italiano)
- TURN_NOTES_D (note sulla modalità di funzionamento del turno in tedesco)
- TURN_NOTES_I (note sulla modalità di funzionamento del turno in italiano)

Vengono elaborate solo le farmacie con CAP = 39012 e di turno (IS_TURN=1) e viene restituito il nome, l'indirizzo, il numero di telefono e gli orari. Essendo gli orari solo numerici vengono convertiti in testo da poter essere letto, quindi invece, ad esempio, di 00:00 viene detto "mezzanotte".

I dati dei musei vengono presi in tempo reale dal servizio provinciale in Opendata REST API JSON: <http://daten.buergernetz.bz.it/services/musport/v1/geojson>

I dati sono informato JSON e contengono:

- BEZEICHNUNG_D (nome del museo in tedesco)
- BEZEICHNUNG_I (nome del museo in italiano)
- BEZEICHNUNG_E (nome del museo in inglese)
- ADRESSE_D (indirizzo in tedesco)
- ADRESSE_I (indirizzo in italiano)
- PLZ (cap)
- ORTSCHAFT_D (nome della città in tedesco)
- ORTSCHAFT_I (nome della città in italiano)
- HOMEPAGE_D (url del sito in tedesco)
- HOMEPAGE_I (url del sito in italiano)
- TELEFON (nr di telefono)
- TELEFON2 (eventuale secondo numero di telefono)
- FAX (numero di fax)
- EMAIL_I (email italiano)
- EMAIL_D (email tedesco)
- EINTRITT_D (costo dell'ingresso e descrizione in tedesco)
- EINTRITT_I (costo dell'ingresso e descrizione in italiano)
- EINTRITT_E (costo dell'ingresso e descrizione in inglese)

I dati vengono filtrati sul PLZ=39012 e vengono presi il nome del museo, l'indirizzo, il numero di telefono ed il costo dell'ingresso

4. Struttura della skill

La creazione di un'action per Google Assistant passa attraverso 2 siti di Google:

- Google Actions per la pubblicazione dello skill (<https://developers.google.com/assistant>)
- DialogFlow per la creazione del modello di intelligenza artificiale per il riconoscimento nelle frasi

Una skill è formata da “Intenti”. Gli “Intenti” sono gli argomenti che la skill tratta e contengono i modelli di frasi. Nella skill del comune abbiamo i seguenti intenti:

- Welcome Intent = Il benvenuto (“Benvenuto nel comune di merano”)
- Default Fallback Intent = le risposte da dare nel caso lo skill non capisca cosa l’utente ha detto
- endConversationIntent = il saluto finale quando l’utente lascia lo skill
- OfficeIntent = se l’argomento è la struttura del Comune e gli eventi (quindi riferito ai dati presenti nel db MySQL)
- NewsIntent = se si chiedono le ultime notizie del Comune (quindi riferito all’RSS delle notizie)
- MuseumIntent = se riguarda i musei (dati presi dagli Opendata dei musei)
- PharmacyIntent = se riguarda la farmacia di turno (dati presi dagli Opendata delle farmacie)

Ogni “Intento” contiene delle frasi di esempio ed ogni frase può contenere dei parametri variabili.

Ad esempio se richiedo quali eventi di musica ci sono domani posso farlo in diverse maniere:

“quali eventi di musica ci sono domani”? oppure “che eventi di musica ci saranno domani”?

in questo caso “quali eventi...ci sono...” e “che eventi... ci saranno...” sono i modelli di frasi da insegnare, mentre “musica” e “domani” sono due parametri, il primo è una categoria di eventi, mentre il secondo è un parametro temporale. Insegnando questo al sistema, potrà quindi riconoscere anche frasi come:

“quali eventi di arte ci sono oggi” oppure “quali eventi di arte ci saranno il 18 ottobre”?

Aumentando il numero di esempi di frasi che vengono date all’intelligenza artificiale legata all’intento, migliora la comprensione linguistica dell’intento stesso. Tutte le richieste fatte le memorizziamo in maniera anonima in fase di test, in modo da identificare frasi utili per migliorare il modello di intelligenza artificiale.

L’intento NewsIntent contiene 2 livelli di interazione, prima elenca le ultime 5 notizie e poi chiede se si vogliono maggiori dettagli su una di esse. Al termine dell’elenco dirà: “Dimmi il numero della notizia della quale vuoi maggiori dettagli o stop per terminare”. Dicendo ad esempio “due”, leggerà tutta la seconda notizia.

Le variabili individuate dall’intento vengono inviate ad un modulo scritto in PHP che supporta gli standard Google Webhook. Questo modulo si occupa di generare la risposta e raccogliere i dati dal database o dagli Open Data. Webhook inoltre filtra ed interpreta in maniera “intelligente” alcune informazioni come ad

esempio nella lingua tedesca quando si usano parole che sono composte unendo due vocaboli (ad esempio ...büro, ...amt).

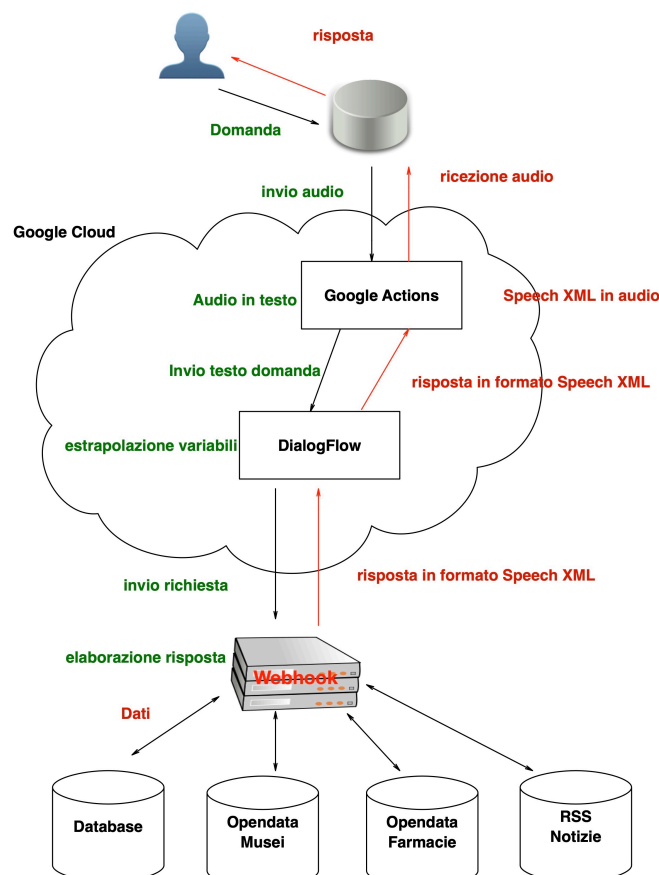
5. Flusso dei dati:

L'utente parla con l'assistente vocale e richiama l'action del comune di Merano.

L'action si attiva e dice "Benvenuto presso il comune di Merano"

L'utente formula la domanda, Google Assistant manda l'audio della domanda ai server di Google che la convertono in testo e lo inviano a DialogFlow.

DialogFlow elabora il testo per capire la domanda, estrapola i parametri ed invia il tutto al modulo Webhook che elabora la risposta prendendo i dati dal database o dalle API e restituisce il risultato in formato Speech



XML di Google al device che provvederà a dirlo a voce all'utente.

6. Pubblicazione della skill

Per essere disponibile su tutti i device che supportano Google Assistant, una skill deve essere pubblicata.

E' possibile mantenere una skill in 3 stadi:

- versione alpha (per un collaudo da parte di utenti selezionati)
- versione beta (per un collaudo da utenti generici, ma limitati di numero)
- pubblicata (per uso da parte di chiunque)

Google per autorizzare la pubblicazione di una skill richiede una verifica che chi pubblica sia veramente il comune interessato, per evitare la pubblicazione di false informazioni

7. Conclusioni

Gli assistenti virtuali come Google Assistant, Amazon Alexa e Apple Siri, faranno sempre più parte della nostra vita quotidiana e permetteranno un accesso più rapido ed intuitivo alle informazioni.

Il livello di intelligenza e capacità di interazione di questi sistemi aumenta di giorno in giorno e diventeranno sempre di più strumenti indispensabili.

L'adozione di queste soluzioni dai parte dei Comuni permetterà di fornire informazioni al cittadino in modo semplice e veloce, alleggerendo l'infrastruttura da compiti ripetitivi e banali.