

Progetto Ingegneria Informatica - Documento di Specifica

Paolo Roncaglioni Stefano Sanitate

14 febbraio 2018

Sommario

Questo documento di specifica racchiude in modo organizzato tutte le ricerche, informazioni e prove che nel corso di questo progetto abbiamo raccolto, creato, analizzato. Dopo una prima parte in cui viene descritto il funzionamento generale del Bot che abbiamo pensato di progettare, si passa ai dettagli e criteri dei test che abbiamo eseguito per poter fare un deployment di una prima semplice demo del progetto descritto.

Indice

1 Generalità	3
1.1 Scopo	3
1.2 Personalità	4
2 Requisiti non funzionali	4
2.1 Prestazioni e Price strategies	4
2.2 Privacy	5
2.3 Motore di Scraping	5
2.4 Persistenza	5
3 Flow	5
4 Ricerca del FrameWork: Bot di test	5
4.1 Requisiti	5
4.2 Parametri di valutazione	5
4.3 Descrizione bot	6
4.4 Deployment description	6
4.4.1 DialogFlow	6
4.4.2 FlowXo	6
4.4.3 PandoraBots	6
4.4.4 GupShup.io	6
4.4.5 Microsoft Bot Framework	6
4.4.6 Wit.ai	6
4.4.7 ChatFuel	6
4.4.8 kitt.ai	6
4.5 Commenti Finali e decisione	6
5 Costruzione bot di Demo	6
5.1 ServerSide	6
5.2 UX (DialogFlow)	6
5.3 Limitazioni	6
6 Futuri Sviluppi	6

1 Generalità

Il progetto é ovviamente iniziato con la delineazione dello scopo del bot, cioè che cos'avrebbe dovuto fare e a chi potrebbe essere stato utile. In un primo momento son venute fuori idee interessanti, ma magari già presenti (e.g. bot orari Trenitalia) e difficilmente migliorabili. Dopo aver quindi analizzato varie possibilità di scelta delle funzionalità del bot, abbiamo infine pensato di progettare uno a tema Politecnico, che racchiuda un insieme di semplici tool allo studente (o al professore) nella gestione della propria vita universitaria.

1.1 Scopo

La definizione delle caratteristiche funzionali inizia col dividerle in utenti che effettuano la propria registrazione con la coppia <Codice persona, Password> e in utenti che non lo fanno. La registrazione non é ovviamente obbligatoria, ma gli utenti loggati avranno a disposizione più interazioni personalizzate col bot, oltre che avere tutte quelle possibili per gli utenti normali non loggati.

Logged Users

1. *Orario universitario*: richiesta orario delle lezioni, previo inserimento di tutti quei parametri che servono per identificare il corso (nome, corso di studio, sede)
2. *Aule libere*: richiesta elenco aule libere, previa scelta del giorno e orario
3. *Orario aula*: richiesta elenco corsi e orari di una specifica aula (utile per controllare velocemente se l'aula davanti a te é libera)
4. *Posizione aula*: restituisce semplicemente l'edificio in cui é l'aula (utile per le matricole)
5. *Calendario accademico*: richiesta orario accademico, ti manda il pdf presente sul sito del Poli

UnLogged Users

1. *Orario universitario personalizzato*: stessa cosa della funzionalità parte utente loggato, ma con i dati a disposizione é possibile avere maggiore accuratezza e non é richiesto l'inserimento dei dati
2. *Esami*: richiesta elenco degli esami a cui ci si é iscritti
3. *Informazione esami*: informazioni su esami a cui si é iscritti (i.e. aula, orario)
4. *Carriera didattica*: restituzione informazioni generali della propria carriera (e.g. media, crediti)
5. *Avvisi generici*: possibilità di settare avvisi generici con qualsiasi evento di trigger (inizialmente solo uscita voti esami con relativa votazione)

1.2 Personalità

Non sapremmo bene come prendere questa parte, poiché non ci è stato detto nulla in merito. Inizialmente di pensava di supporre che il nostro bot possa rispondere a un set di comandi predefiniti e dati da noi, non lasciando spazio all'utente per eventuali frasi complesse da interpretare. Questo approccio ci avrebbe permesso di tralasciare quasi completamente la parte sul parsing e riconoscimento degli intenti. Man mano che cominciavamo a sviluppare il nostro progetto però ci è sembrato limitante questo approccio, poiché la vera natura del nostro progetto era di "esplorazione" di questo nuovo ambiente. Abbiamo quindi convenuto che la parte interessante sarebbe stata appunto confrontare le varie soluzioni già presenti di NLU (natural language understanding) e rendere la conversazione col bot il più vicino possibile a quella con un umano. Ecco quindi che si delinea l'importanza dell'utilizzo di un Framework che abbia queste possibilità.

2 Requisiti non funzionali

Sotto questa voce vanno i requisiti e caratteristiche che impongono vincoli allo sviluppo del bot, sotto diversi punti di vista. Per ogni problema si cercherà quindi una possibile soluzione di implementazione, che non andremo a sviluppare nella demo in quanto non riteniamo necessario in questo progetto approfondire ulteriormente.

2.1 Prestazioni Price strategies

Secondo il modello posto sopra, un bot in generale si può rappresentare con due parti distinte, poste totalmente o in parte su un server esterno. La scelta del server influenzerà sicuramente il numero di interrogazioni che si possono fare, la velocità di risposta e l'usabilità in generale, a discapito del prezzo che la piattaforma di hosting ci farà pagare per avere questi servizi. Se si utilizza una soluzione "locale" in cui il proprio pc fa da server si avranno sicuramente degli svantaggi rispetto a una soluzione cloud, ma il prezzo sarà nullo e le limitazioni (sulla carta) non esisterebbero. Ad ora, esplorando le varie possibili scelte, per la fase di testing della demo è stato possibile sempre usufruire delle opzioni "free" della piattaforma di turno. Questo include sia UX che Hosting Server.

2.2 Privacy

2.3 Motore di Scraping

2.4 Persistenza

3 Flow

4 Ricerca del Framework: Bot di test

Bot creato per testare i vari framework esistenti, aiuterà la selezione del nostro ambiente di lavoro in base a dei parametri illustrati sotto. È essenziale che sia un bot semplice (per non perdere troppo tempo con lo sviluppo, essendo di test), ma con funzionalità che siano correlate alla costruzione del nostro progetto in specifica. Pertanto ecco le regole che pensiamo debba soddisfare:

4.1 Requisiti

4.2 Parametri di valutazione

Questi rappresentano i parametri che useremo per valutare il framework che testeremo. Essi comprendono sia i componenti di valutazione della costruzione del bot, che quelli più generali della piattaforma che stiamo usando. La valutazione è a nostra discrezione, ovviamente seguita da una minima descrizione delle procedure svolte.

Bot

- *Facilità di progettazione:*
- *Tempo di implementazione:*
- *Profondità:*
- *Persistenza della memoria:*

Framework

- *Pricing strategies:*
- *Piattaforme:*
- *Disponibilità di SDK:*

Framework

4.3 Descrizione bot

4.4 Deployment description

4.4.1 DialogFlow

Implementazione

Valutazione

- *Facilità di progettazione:*
- *Tempo di implementazione:*
- *Profondità:*
- *Persistenza della memoria:*
- *Pricing strategies:*
- *Piattaforme:*
- *Disponibilità di SDK:*

Commento personale

4.4.2 FlowXo

Implementazione

Valutazione

- *Facilità di progettazione:*
- *Tempo di implementazione:*
- *Profondità:*
- *Persistenza della memoria:*
- *Pricing strategies:*
- *Piattaforme:*
- *Disponibilità di SDK:*

Commento personale

4.4.3 PandoraBots

Implementazione

Valutazione

- *Facilità di progettazione:*
- *Tempo di implementazione:*
- *Profondità:*
- *Persistenza della memoria:*
- *Pricing strategies:*
- *Piattaforme:*
- *Disponibilità di SDK:*

Commento personale

4.4.4 GupShup.io

Implementazione

Valutazione

- *Facilità di progettazione:*
- *Tempo di implementazione:*
- *Profondità:*
- *Persistenza della memoria:*
- *Pricing strategies:*
- *Piattaforme:*
- *Disponibilità di SDK:*

Commento personale

4.4.5 Microsoft Bot Framework

Implementazione

Valutazione

- *Facilità di progettazione:*
- *Tempo di implementazione:*
- *Profondità:*
- *Persistenza della memoria:*
- *Pricing strategies:*
- *Piattaforme:*
- *Disponibilità di SDK:*

Commento personale

4.4.6 Wit.ai

Implementazione

Valutazione

- *Facilità di progettazione:*
- *Tempo di implementazione:*
- *Profondità:*
- *Persistenza della memoria:*
- *Pricing strategies:*
- *Piattaforme:*
- *Disponibilità di SDK:*

Commento personale

4.4.7 ChatFuel

Implementazione

Valutazione

- *Facilità di progettazione:*
- *Tempo di implementazione:*
- *Profondità:*
- *Persistenza della memoria:*
- *Pricing strategies:*
- *Piattaforme:*
- *Disponibilità di SDK:*

Commento personale

4.4.8 kitt.ai

Implementazione

Valutazione

- *Facilità di progettazione:*
- *Tempo di implementazione:*
- *Profondità:*
- *Persistenza della memoria:*
- *Pricing strategies:*
- *Piattaforme:*
- *Disponibilità di SDK:*

Commento personale

4.5 Commenti Finali e decisione

5 Costruzione bot di Demo

5.1 ServerSide

5.2 UX (DialogFlow)

5.3 Limitazioni

6 Futuri Sviluppi

Riferimenti bibliografici

bibliografia, software usati, github repo.