
This is the **published version** of the bachelor thesis:

Arauzo González, Victor; Prim i Sabrià, Marta, dir. Graubot : Xat bot per resoldre dubtes del grau. 2022. (958 Enginyeria Informàtica)

This version is available at <https://ddd.uab.cat/record/264205>

under the terms of the  license

Graubot:

Xat bot per resoldre dubtes del grau

Víctor Arauzo González

Resum— El principal repte del processament del llenguatge natural és que una màquina entengui un idioma i sigui capaç de dur a terme un diàleg escrit o parlat coherent. En aquest treball final de grau es planteja una interfície de llenguatge natural --anomenada *Graubot*-- per a l'automatització de la resolució de dubtes dels estudiants de l'Escola d'Enginyeria de la Universitat Autònoma de Barcelona a temps real en forma de xat bot. La resolució del problema s'implementa mitjançant paraules clau i amb la gestió del context de la conversa amb l'usuari. El treball ha estat dividit en 3 parts: la interfície web, mitjançant la qual l'usuari es comunica amb el bot a través d'un xat, el servidor, que emmagatzema i intercanvia entre els agents implicats les preguntes o respostes, i finalment, el propi algoritme de *Graubot*, que s'encarrega de processar la petició de l'usuari i proporcionar-li una resposta precisa mitjançant una sèrie de funcions. En aquest treball es demostra la capacitat del xat bot de mantenir i entendre converses en català i castellà per a resoldre peticions d'estudiants sobre horaris, exàmens, treballs final de grau, guies docents i d'altres de manera eficient.

Paraules clau— assistent virtual, grau universitari, llenguatge natural, paraules clau, robot, xatbot

Abstract— The main challenge of natural language processing is for a machine to understand a language and be able to conduct coherent written or spoken dialogue. In this final degree project a natural language interface is proposed --called *Graubot*-- for the automation of the resolution of doubts by students of the Faculty of Computer Engineering of the Autonomous University of Barcelona in real time in the form of chat bot. Problem solving is implemented through keywords and with the management of the context of the conversation with the user. The work has been divided into 3 parts; The web interface, where the user communicates with the bot through a chat, the server, which stores and exchanges questions or answers between the agents involved, and finally *Graubot*'s algorithm, which is responsible for processing the user's request and provide an accurate answer through a number of functions. Multiple experiments have demonstrated the ability of the chat bot to maintain and understand conversations in Catalan and Spanish to resolve student requests about schedules, exams, final degree assignments, teaching guides, among others, in an efficient way.

Índex Terms—virtual assistant, university degree, natural language, key words, robot, chatbot



1 INTRODUCCIÓ

UN xat bot és una aplicació informàtica basada que es comunica amb un usuari a través de missatges de text o veu. Permet simular una conversa amb un usuari, donant respostes automàtiques als seus dubtes o preguntes més comunes sense que hi hagi una persona física responnent aquestes preguntes.

La intel·ligència artificial s'utilitza quan aquest programa necessita reconèixer les intencions de les frases que introdueixen els usuaris en el diàleg per tal de classificar-les. També s'usa en el pas posterior de seleccionar quina ha de ser la resposta (o entitat) a donar a l'usuari.

Els principals avantatges d'un xat bot solen ser l'atenció personalitzada i immediata, una bona accessibilitat per a

l'usuari mitjançant una interfície coneguda i intuïtiva i facilitat d'ús, ja que permeten parlar amb el llenguatge que l'usuari fa servir en la seva vida quotidiana.

1.1 Organització del document

Aquest document està compost per 4 seccions distingides. Primerament es desenvolupa la introducció, amb els objectius del treball, l'estat de l'art i les motivacions que han portat a començar-lo.

Després ens trobem el bloc de la metodologia, amb la informació de com es durà el projecte de creació del prototip, les eines per fer-ho i una planificació del procés que seguirem. El document continua amb el desenvolupament, secció on es detalla el treball realitzat amb cada un de les seves etapes i l'explicació de cada mòdul del software i el seu funcionament.

Finalment, es mostren els resultats obtinguts de manera satisfactòria i el que és capaç de fer el robot arribats al final de tot el procés. S'han extret un seguit de conclusions relacionades amb els objectius inicials tant complerts com no assolits i els problemes trobats durant el projecte i com s'han solucionat.

-
- E-mail de contacte: victor.arauzo@autonoma.cat
 - Menció realitzada: *Computació*
 - Treball tutoritzat per: Marta Prim Sabrià (Departament de Microelectrònica i Sistemes Electrònics)
 - Curs 2021/22

S'acompanya d'aquestes conclusions unes possibles línies futures de desenvolupament a mode de suggeriment. En l'última secció d'aquest article s'adjunten uns agraïments seguit de les referències bibliogràfiques utilitzades per la consulta d'informació.

1.2 Objectius del treball

L'objectiu principal proposat a l'hora de desenvolupar aquest projecte és:

Contestar els dubtes dels estudiants de l'Escola d'Enginyeria que tenen sobre els diferents graus que si cursen, de manera senzilla i autònoma, mitjançant un xat bot basat en intel·ligència artificial, per ajudar a reduir el temps que dediquen els coordinadors a la seva resolució. Aquest objectiu s'ha de complir en un termini de 5 mesos.

1.3 Estat de l'art

Existeixen moltes alternatives de xat bots per tal d'automatitzar les necessitats que un usuari pot tenir sobre un sistema o tòpic en concret, depenent en l'àmbit en què es vulgui implementar. Actualment, podem classificar els xat bots segons la categoria conversacional en tres tipus de funcionament que s'expliquen a continuació:

Xatbots de ITR (Resposta d'Interacció de Text) o "dumb chatbots" són aquells que no requereixen l'aplicació d'intel·ligència artificial, ja que funcionen basant-se en comandes. Utilitzen botons predefinits i segueixen una lògica seqüencial, emulant una conversa però sempre a partir d'un menú d'opcions prèviament establert. És el xatbot més bàsic, però la selecció d'opcions és ràpida i intuïtiva, cosa que el converteix en un xatbot efectiu i eficient.

Els xatbots basats en dades i predictius (conversacionals) o "smart chatbots" es denominen amb freqüència assistents virtuals o assistents digitals i són molt més avançats, interactius i personalitzats que els xatbots orientats a tasques (més endavant). Aquests xatbots són conscients del context i aprofiten la comprensió del llenguatge natural (NLU), el NLP i el ML per a aprendre sobre la marxa. Els assistents digitals poden aprendre les preferències de l'usuari amb el temps, oferir recomanacions i fins i tot anticipar-se a les necessitats.

Finalment els xatbots orientats a tasques o de "keywords" són un bot intermedi entre els de ITR i els conversacionals. Fan servir el processament de llenguatge natural (NLP) pel reconeixement de paraules clau, en funció de les quals donen una resposta predefinida. Són molt més conversacional, sense necessitat d'integrar una tecnologia de NLU, i per això és un dels bots més utilitzats avui dia. Les interaccions són aplicables a les funcions de suport i servei: preguntes freqüents interactives de pensament sòlid. La seva limitació, justament, és que només interpreta paraules clau (*keywords*), sense tenir en compte el context o la intenció.

El funcionament de qualsevol xat bot es defineix amb tres grans fases en el procés per donar resposta a la pregunta introduïda per un usuari:

1. Processament del llenguatge natural (NLP)

S'utilitza per dividir l'entrada de text de l'usuari en paraules. S'encarrega de tractar computacionalment el llenguatge humà de l'usuari, com separar les paraules importants i la resta de l'entrada

2. Comprensió del llenguatge natural (NLU)

Fase en què el programa entén el que l'usuari diu o vol dir a través d'una classificació mitjançant vocabulari, sinònims o temes dels quals pot voler parlar. S'utilitzen unes "normes" per indicar al bot que ha de respondre a la petició.

3. Generació de llenguatge natural (NLG)

Part de la generació de missatges de text per a respondre a l'usuari. Acostumen a ser respostes amb part d'informació "prefabricada" que després es completa amb la generació de llenguatge natural (oracions) per tal que l'usuari rebi una bona experiència simulant el comportament humà.

1.4 Motivació

La motivació i interès pels assistents virtuals va començar des de l'aparició al mercat d'aquest productes. Al poc temps a casa ja en fèiem servir un. El que més m'interessava d'aquest aparell, era el software basat en intel·ligència artificial amb el que funcionava i com era capaç de donar una resposta a les nostres preguntes. Tenia ganes d'aprendre el funcionament i de saber tot el que podia fer, també el que no podia fer.

A més a més durant el meu primer període de pràctiques curriculars de l'any passat vaig descobrir Bookline. Una empresa on vaig treballar en el seu projecte de negoci el qual consistia en un assistent de veu telefònic que automatitzava les reserves dels clients a restaurant. Allà vaig aprendre el funcionament bàsic dels xat bots basats en "keywords" i context, i com dotar-los d'un nivell més d'intel·ligència.

Aquest interès i aquest aprenentatge que havia adquirit tant en el període de pràctiques com en els anys de carrera, em van animar en aprofundir més sobre el tema dels assistents i a voler desenvolupar una versió del meu primer xat bot.

2 METODOLOGIA

Durant el termini dels 5 mesos i per tal de desenvolupar tant la documentació com l'apartat de codi del projecte a desenvolupar s'ha fet ús de la metodologia del model de cascada i posteriorment el model incremental.

Aquesta primer model és un procés lineal per tal de crear el model del prototip funcional, en què les diferents fases

del projecte (Requisits, Anàlisi, disseny, programació, test) es desenvolupen en ordre seqüencial. Totes les fases estan relacionades per una dependència, de manera que s'haurà de finalitzar cada fase abans de començar amb la següent.

Dins de cada fase, hi haurà un seguit de diferents tasques a desenvolupar. Aquestes tasques s'organitzaran seguint la metodologia *Kanban*, on es representen les tasques pendents del projecte utilitzant elements visuals com un taulell (*backlog*, pendent i completada). Aquestes tasques seran la unitat més petita del projecte i no necessàriament tindran un ordre marcat dins de cada fase, poden poder-se desenvolupar de manera simultània.

Aconseguit un MVP (*minimum viable product*) partirem d'aquest model simple amb l'arquitectura completa on anirem fent iteracions per tal de millorar i afegir funcionalitats al robot.

En la Fig. 1 s'hi observa la metodologia a seguir per al desenvolupament.

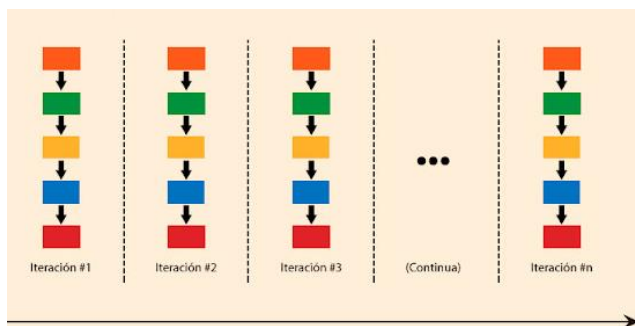


Fig 1 Il·lustració de la metodologia proposada. [10]

2.1 Eines utilitzades

Per al desenvolupament del software d'aquest projecte s'han fet servir diferents eines per tal d'integrar el codi creat i mantenir-lo de manera eficient.

Per a la programació de l'aplicació i el motor d'intel·ligència artificial, s'ha fet servir el llenguatge *Python*, amb l'editor de codi font *Visual Studio Code*. Aquest editor ve amb un seguit d'extensions que el fa molt còmode per treballar amb varis mòduls i llenguatges.

Una segona part poc representativa del codi ha estat desenvolupada en *HTML* i *CSS*, per tal de crear una interfície web per allotjar el client del xat i fer-lo accessible als usuaris.

Per a l'allotjament de dades que el robot ha de subministrar, com el dades sobre de les assignatures, els exàmens o la informació de les diferents temàtiques s'ha utilitzat el servei de base de dades no relacional de *google cloud platform* anomenat *Firebase firestore*. D'aquesta mateixa plataforma s'ha utilitzat *Firebase hosting* per allotjar la nostra pàgina web. El desplegament del servei s'ha fet servir la

plataforma *Heroku* que és una de les millors alternatives per al desplegament d'aplicacions de manera gratuïta i automatitzada.

S'ha fet servir l'eina *Git* per al desenvolupament mitjançant el control de versions del codi i l'emmagatzematge al núvol, concretament el servei gratuït *Git Hub*. Aquest control de versions permetia *rollbacks* en cas de necessitar-ho.

Finalment s'ha fet ús d'una eina que utilitzem com a interfície client per a l'entrada de peticions i dubtes dels usuaris anomenada *Chatwoot*.

2.2 Planificació

La planificació inicial plantejava dues versions del motor per tal de comparar resultats i triar en les etapes posteriors quina de les dos versions utilitzar a l'hora de fer una gestió de les peticions dels usuaris:

1. Familiarització amb les tecnologies i eines a utilitzar
Durant el mes de febrer el focus de feina principal serà l'aprenentatge.
2. Primera versió del motor del xat bot utilitzant keywords

Aquesta primera versió d'agent intel·ligent basarà el seu reconeixement de les intencions de l'usuari a través de informació predefinida.

3. Segona versió del motor del xat bot utilitzant models d'aprenentatge autònom

Aquesta segona versió d'agent intel·ligent basarà el seu reconeixement de les intencions de l'usuari a través de models d'aprenentatge computacional que permetran una interacció més fluïda i una millor experiència d'usuari. En aquest cas es necessiten gran quantitat de dades que s'hauran de recopilar, per tant que el motor aprengui iteració rere iteració.

4. Disseny i refinament de la interfície d'usuari
Disseny d'una interfície d'usuari agradable i intuïtiva per tal que l'usuari tingui una experiència immersiva i semblant el màxim a la interacció humana.
5. Implementació del xat bot en aplicació web accessible als usuaris

Últim pas que servirà per tal que l'usuari pugui fer ús del servei de manera ininterrompuda allotjant el servei en un servidor públic. En la Fig. 2 es pot veure un esquema de la planificació inicial.

	Febrer	Març	Abril	Maig	Juny
1. Familiarització					
2. Vers. Keywords					
3. Vers. models ML					
4. UI					
5. Deployment					

Fig 3 Planificació inicial del projecte.

Com es pot observar a les Fig. 3 la planificació ha variat en el transcurs del projecte. Una fase com la implementació de la versió de paraules claus i una arquitectura i disseny del software més complexa de lo esperat fa que s'hagin endarrerit les tasques i no hagi sigut viable desenvolupar dues versions del motor. La Fig. 3 mostra la planificació pràctica del projecte.

	Febrer	Març	Abril	Maig	Juny
1. Familiarització					
2. UI					
2. Disseny i arquitectura del projecte					
3. Motor amb Keywords					
5. Deployment					

Fig 4 Planificació pràctica del projecte.

3 DESENVOLUPAMENT

En aquest apartat s'explica el treball fet durant les diferents fases del projecte, des de la recopilació inicial de dades fins el desplegament del servei passant per la fases de disseny, arquitectura i codificació

2.1 Disseny

Per tal de poder tenir totes les possibles combinacions de preguntes que l'usuari podia fer, i obtenir una resposta precisa a les seves es va realitzar un treball previ de sintetització de la informació. Tota aquesta informació es va agrupar per tal de poder englobar totes les dades en el menor espai possible i la màxima reutilització d'aquestes. La idea abstracta d'aquest treball ha sigut crear un arbre de combinacions, on les primeres branques són peticions generals i a mesura que pugem de nivell es converteixen en peticions més concretes fins arribar a les fulles, que serien les respostes a donar. Cada fulla és una combinació de peticions de l'usuari i el robot a de tenir-ho el més fàcil possible per tal de crear combinacions de peticions i donar la millor resposta.

Una segona tasca, ha sigut idear l'estructura òptima del codi. De manera que s'intentin seguir les directrius bàsiques de disseny de software i de *clean code*, alhora que es creava una arquitectura fàcil d'entendre i que fos intuïtiva, amb mòduls diferenciats que executessin una sola funció imprescindible cada un.

A continuació, la Fig. 4 hi ha una il·lustració de l'agrupació de temes amb format arbre.



Fig 2 Esquema del disseny arbre de respostes.

2.2. Recopilació de dades

Les dades del que s'han hagut de guardar per les consultes dels usuaris han estat extretes dels llocs webs de l'escola i dels documents de preguntes freqüents disponibles també als webs. S'ha fet un *scrapping* de manera manual per tal d'obtenir les informacions més rellevants i que més podien demanar els estudiants.

A part d'això s'ha intercanviat informació amb el Coordinador del grau, encarregat de l'actual recepció i resolució d'aquests dubtes dels usuaris de l'Escola que li fan arribar via correu electrònic. D'aquesta manera s'ha pogut tenir una primera idea dels dubtes que més es repetien entre els alumnes amb casos reals i emmagatzemar aquesta informació de manera més òptima per al fàcil accés del codi.

2.3 Test Drive Developing (TDD)

Una part molt important del desenvolupament i que més temps comporta a l'hora de dur a terme un projecte de software és la detecció i resolució dels anomenats *bugs* o errors. És per això que durant tot el procés de desenvolupament del codi font del projecte, s'ha fet servir constantment la metodologia guiada per test.

Aquest paradigma de desenvolupament de software té com a idea principal no crear cap funció o sentència del codi que no estigui recolzada de test. Aquest test serà el principi del nostre desenvolupament i ens permetrà tenir la seguretat que aquella funció funciona en tot moment de la manera que s'espera.

TDD consisteix en decidir quina és la resposta esperada (i no esperada) de cada funció atòmica del nostre programa. De manera que abans de programar la funcionalitat en sí, es crea un test amb les possibles respostes que ens pot donar i una manera de comparar-les que sigui veritat. Després es crea la funcionalitat en sí del programa i s'executen els tests per tal de verificar que la funció ens dona el resultat correcte i precis. Això ens ajuda a crear codi més eficientment ja que saber en tot moment que cada sentència ens dona el resultat esperat.

A la Fig. 5 es mostra l'esquema del paradigma de desenvolupament TDD.

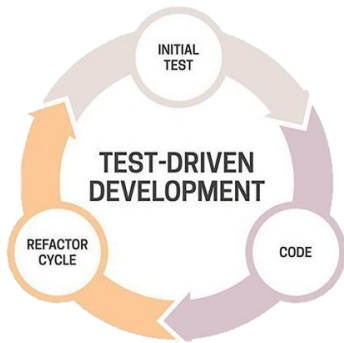


Fig 5 Esquema del cicle TDD, [10]

Aquesta metodologia serveix també per un futur, desenvolupar nou software sense “espatllar” l'anterior funcionament.

2.4 Arquitectura del software

L'arquitectura principal del programa està composta per 4 mòduls ben diferenciats:

El primer engloba la part del *frontend* que és l'encarregada d'interaccionar amb l'usuari i captar les seves preguntes. Aquesta està formada pels fitxers de contingut i estils (*HTML*, *CSS*) que contenen dins seu el *widget* que farà d'interfície xat.

Per altra banda ens trobem el servidor de *Chatwoot*, on la plataforma recull els missatges que s'envien a través del *widget* desplegat. S'emmagatzema la informació a l'espera que la demani el següent mòdul.

Aquest tercer mòdul és el codi que s'ha desenvolupat per tal d'identificar les preguntes, processar-les i donar la millor resposta. És on es troba la principal feina del servei i l'anomenat *backend*. És comunica amb el servidor gràcies a una API.

Finalment ens trobem l'últim mòdul de tots, la base de dades que emmagatzema de manera persistent totes les dades que necessitem per respondre a l'usuari.

En la Fig. 6 s'il·lustra l'arquitectura del programa.

2.5 Flux del programa

El flux del programa comença quan un usuari obre la pestanya del xat de la nostra web.

El primer que fa l'aplicatiu és donar-li la benvinguda i suggerir-li que faci una pregunta. Quan l'usuari fa aquesta pregunta al *widget* de la web el servidor de *chatwoot* la rep. El nostre codi *backend* que té un procés en constant escolta de les noves informacions al servidor identifica que hi ha

una nova pregunta.

El primer que fa amb aquesta pregunta és processar-la i identificar el que se li està demanant. Un cop fet això busca a la base de dades la resposta a donar i altre cop a través de la API (interfície de programació d'aplicacions) envia aquesta resposta al servidor. Aquest mateix servidor és el que posteriorment s'encarrega d'enviar-la a l'usuari a través novament de la interfície del xat.

2.6 Interfície d'usuari

Com a interfície d'interacció dels estudiants, s'ha fet servir un xat dels que es coneixem comunament per entrar missatges per text.

Per tal d'obtenir una experiència d'usuari agradable i intuïtiva s'ha integrat una d'interfície a través d'una plataforma *opensource* anomenada *Chatwoot*.

Aquesta plataforma consisteix en un xat tradicional que es desplega a la web en forma de *widget*. És personalitzat pel projecte i demana a l'usuari que faci preguntes per tal que pugui assessorar-lo.

L'objectiu d'aquesta part és que sigui la part que actua de *frontend* per tal que les preguntes dels usuaris arribin a codi desplegat al servidor.

Per tal d'integrar aquesta plataforma amb el nostre codi que conté el motor d'intel·ligència artificial, fem servir una API que fa utilitza mètodes del protocol HTTP per tal d'intercanviar els missatges entre el servidor i el nostre codi.

2.7 API

API significa “interfície de programació d'aplicacions” i és un sistema de connexió entre dos serveis de dos aplicacions diferents. En altres paraules, aquest mecanisme permet comunicació entre dos softwares. Fa servir peticions *http* per tal de fer intercanvi d'informació entre client i servidor. Per tal de recollir la informació del servidor que emmagatzemava les diferents preguntes de l'usuari i alhora, enviar les respostes que generava el robot, s'ha fet us de la api de xat bot.

Dins el projecte, la API s'ha fet servir entre la part del codi de *Graubot* i el servidor de *chatwoot*.

En el context del nostre hi ha 3 funcions principals que s'utilitzen:

La primera *get_all_conversations()* s'ha utilitzat per obtenir les diferents converses dels usuaris que van creant a mesura que passa el temps. Amb l'identificador que ens retorna podem tenir la referència de la conversa per enviar i rebre missatges amb les dues següents funcions.

La segona *get_messages_from_conversation()* s'utilitza per recuperar l'últim missatge que ha enviat l'usuari per tal de processar-lo.

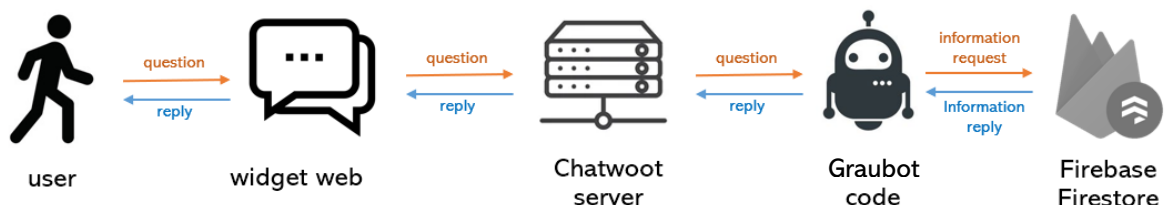


Fig 6 Arquitectura software del xat bot *Graubot*.

La tercera funció `post_messages_to_conversation()` es fa servir per enviar la resposta a l'usuari un cop generada.

2.8 Mòduls del codi principal

S'ha definit una estructura de directoris on cada mòdul té una funció concreta a realitzar. Aquesta part que rep la pregunta i genera la resposta conté el següents mòduls de codi que s'expliquen a continuació:

intents.py

Mòdul NLU que s'encarrega d'identificar quina és la intenció de l'usuari quan fa una pregunta, ja que durant el manteniment d'una conversa poden haver-hi multitud d'intencions diferents de l'usuari, entre les quals ens podem trobar: saludar, acomiadar-se, preguntar, demanar informació sobre un tema en concret (entitats) o també conté les

funcions per tal d'identificar les intencions més freqüents que els usuaris pregunten als xat bot: Com estàs?, Qui ets?, Com funcionen?, etc.

entities.py

Mòdul del NLU que parteix d'un missatge de l'usuari pre-processat i n'obté la llista d'entitats (informació clau). Aquesta llista d'entitats s'obté en forma de llista, on hi apareix cada entitat amb el seu valor.

Per tal de trobar cada entitat es fan servir llistes de paraules clau per tal d'identificar a quina informació s'està referint l'usuari.

Un exemple seria si l'usuari demana "Vull els horaris de 2n del grau d'Informàtica", aquest codi ens donaria les entitats

```
horaris = True,
curs = 2n,
grau = Enginyeria informàtica
```

Aquest mòdul també és l'encarregat de detectar les subentitats dels missatges. Aquestes entitats més concretes serveixen per identificar els subtemes dels tòpics que l'usuari es refereix a primera instància. Així doncs, una subentitat podria ser la duració o la data d'entrega que l'usuari ens podria demanar d'una entitat mare com podria ser el tfg o les pràctiques curriculars. "Vull saber la duració del tfg";

```
entity = tfg,
sub_entity = tfg_duration
```

tools.py

Mòdul multi ús que conté funcions tant de tractament del missatge (*pre process*) com *parsejador* de formats i altres, per tal d'adequar el missatge al bot.

dialogue_manager.py

Mòdul *core* del codi que s'encarrega de decidir la pròxima acció del robot (cap a on anirà encaminada la resposta) a partir de la intenció de la pregunta, la llista d'entitats i el context de la conversa. Consta bàsicament de

combinacions de condicions entre entitats, de manera que, a mesura que l'usuari ens dona informació del que necessita, podem assignar una acció com per exemple "*guiado-cent_tfg_informàtica*" i es buscarà la resposta que compleixi amb aquest patró.

És on hi ha el flux del programa principal depenent del que l'usuari ens hagi transmès. En aquest mòdul hi intervenen tant les intencions, entitats i subentitats com el context i l'acció que durem a terme gràcies a tota la informació rebuda. És el pas previ a donar una resposta a l'usuari.

A la Fig. 7 mostrem l'estructura de directoris dels mòduls anteriorment comentats.

```
├─ chatbot-dev
│  ├─ api
│  │  └─ main.py
│  ├─ bot
│  │  └─ data
│  │     └─ responses.json
│  │     └─ src
│  │        └─ nlg
│  │           └─ reply.py
│  │           └─ nlu
│  │              └─ entities.py
│  │              └─ intents.py
│  │              └─ utils
│  │                 └─ tools.py
│  │                 └─ context.py
│  │                 └─ dialogue_manager.py
│  │                 └─ flowbot.py
│  │                 └─ main.py
│  └─ test
├─ db
│  └─ main.py
├─ public
│  └─ index.html
│  └─ javascript.js
```

Fig 7 Estructura directoris.

reply.py

Mòdul que s'encarrega de generar una resposta per a l'usuari en forma de text. La crea en funció d'una acció i un context donat, de manera que se li donarà a l'usuari una resposta simple (predefinida del fitxer) triada de manera aleatòria de la llista, o dinàmica, utilitzant les dades guardades a la base de dades de la temàtica que s'estigui resolent.

flowbot.py

Mòdul encarregat del flux general del programa, des de

que obtenim un missatge de l'usuari, fins que generem una resposta. En termes del codi del robot, és una funció principal que se li dona un missatge d'usuari per tal d'extreure la intenció, la llista d'entitats i el context. Aquestes es passen al gestor de diàleg que juntament amb un context anterior, ens generarà una acció. Aquesta acció ens generarà una resposta a partir del mòdul mitjançant una part de la resposta fixa i una variable.

main.py

Mòdul principal del programa on integra el funcionament del bot amb la connexió a la API per tal d'obtenir els missatges del servidor de *Chatwoot*. Consta d'un bucle infinit per tal que sempre s'esperin missatges de l'usuari i respostes del robot. Fa la crida de la funció missatge del bot i obtindrà d'aquesta una resposta que passarà al servidor de *Chatwoot* altre cop.

En la Fig. 8 es mostra el diagrama de flux del codi principal.

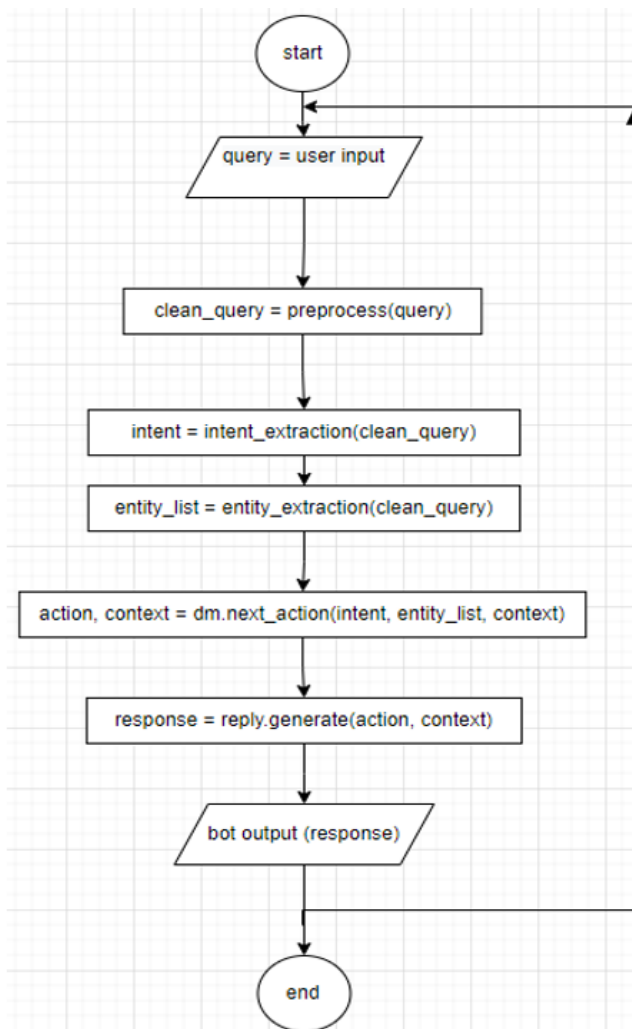


Fig 8 Diagrama de flux del backend de Graubot.

processador que ens permet paral·lelitzar el codi. D'aquesta manera podem tenir una execució del robot per cada una de les noves converses que ens vagin sortint i no tenir un problema d'embotellament quan dos usuaris vulguin accedir al servei.

2.9 Allotjament i desplegament

La idea del projecte és que aquest xat bot estigui integrat en una de les pàgines web de l'escola. Per tal de poder desenvolupar el projecte amb independència, s'ha fet servir en una primera instància els serveis d'allotjament i desplegament de *Google Firebase Hosting* i *Heroku* respectivament, les dos en el mode gratuït.

Per una banda *Firebase hosting* ens proporciona espai per allotjar la nostra web. En un servidor públic, accessible mitjançant la url: <https://chatbot-dev-a51e0.web.app/>. Per a integrar aquest servei a la nostra aplicació, s'han configurat alguns fitxers en el nostre repositori amb les claus d'accés i *Google* s'encarrega de la resta.

Per altra banda el servei de desplegament consisteix en pujar el repositori en un servidor públic de codi obert anomenat *Heroku*.

Aquesta eina permet que el codi s'executi de manera ininterrompuda en una màquina al núvol, de manera que les funcionalitats estiguin sempre preparades per respondre als usuaris que vulguin fer servir l'aplicació web. En les primeres fases del desenvolupament i proves això es feia des de la pròpia màquina en local. Quan ja hi ha una versió final funcional, *Heroku* accedeix al repositori del núvol (*GitHub*) i desplega el codi en una de les seves màquines.

2.10 Proves

Per tal de comprovar que els requeriments del projecte es compleixen i el robot era capaç d'entendre i seguir una conversa, s'han realitzat un seguit de proves per validar el bon funcionament del robot que s'expliquen a continuació. Aquestes proves s'han realitzat a mesura que es desenvolupaven funcionalitats noves per comprovar que aquella funcionalitat complia amb el seu correcte funcionament. La majoria d'aquests tests, s'han fet mitjançant l'entrada de preguntes i la comparació de respostes esperades mitjançant tests en l'entorn de desenvolupament a nivell tant local (execució ordinador personal) com en remot (servei desplegat al núvol).

La idea principal d'aquestes proves és saber si el robot dona la resposta precisa al que se li està demanant de diverses maneres. També validar que els diferents mòduls funcionen i interaccionen tots junts entre sí, sense que hi hagi errors de tipus o mala comunicació.

També s'ha inclòs una gestió dels errors amb la sentència *try, except* per tal de gestionar les excepcions que ens llença el nostre codi amb facilitat.

Per tal de poder gestionar més d'una conversa alhora, s'han fet ús dels anomenats *threads*, fils d'una mateixa

A la Fig. 9 de continuació, mostrem la metodologia amb que s'han dut a terme les proves i l'entorn de desenvolupament.

```

Bot: Hola, encantat de coneixet. Em podries dir el teu nom?
User: m'agradaria saber si tens els horaris de primer curs d'enginyeria informàtica de la universitat autònoma. El 1r semestre vull!

Starting...
Starting...
Query => m'agradaria saber si tens els horaris de primer curs d'enginyeria informàtica de la universitat autònoma. El 1r semestre vull!
Intent captured is => ['info', 75]
List of entity captured is => [['schedule', True], ['course', '1'], ['degree', 'informàtica'], ['semester', '1']]
List of sub_entity captured is => []
Status => schedule
Context until now=>
Context(course='1', degree='informàtica', department=None, language='cat', mention=None, professor=None, semester='1', status='schedule', subject=None, term=None, username='', year=None, adeu=False, start=False)
Next action will be => schedule
Bot reply is => Aquí tens l'horari que em demanes : https://www.uab.cat/doc/Horari_GEI_Curs1_Sem1
Response send!

Bot: Aquí tens l'horari que em demanes : https://www.uab.cat/doc/Horari_GEI_Curs1_Sem1

```

Fig 9 Entorn de desenvolupament i prova de frase llarga amb totes les entitats definides.

3 RESULTATS

Els resultats aconseguits al final d'aquest projecte són molt satisfactoris en quant a compliment dels objectius. Tenim un xat bot funcional capaç de respondre dubtes i mantenir una conversa de manera prou fluida.

Actualment, el robot té la capacitat d'interactuar amb 2 idiomes diferents, identificar la petició dels diferents intencions de l'usuari, i donar resposta a preguntes sobre els horaris del grau (**vull els horaris de 1r curs de química**), oferir les guies docents de les assignatures (**tens siusplau la guia docent d'àlgebra d'enginyeria informàtica?**), els calendaris d'exàmens (**necessito que m'aconsegueixis les dates dels exàmens finals del 2n semestre d'enginyeria de dades!**). En aquests casos el robot pregunta tot el que necessita saber (grau, curs, semestre, menció, assignatura) encara que l'usuari no ho expressi explícitament de primeres.

La informació sobre el tfg i les pràctiques també la pot donar el bot (**saps quantes setmanes dura el tfg? o quantes hores he de fer de pràctiques curriculars?**). També identifica i dona resposta sobre peticions emmarcades als programes de mobilitat, (**on puc trobar les destinacions d'Erasmus?**), les condicions de matriculació, (**com saps puc fer el pagament de la matrícula del grau d'intel·ligència artificial?**), el règim de permanència i condicions per al reconeixement de crèdits i finalment, enllaç a la pàgina de l'expedient i de tràmits de la universitat.

S'ha procurat fer el desenvolupament robust a les diferents maneres d'expressar una mateixa pregunta i s'han integrat correctament les diferents eines del projecte com el *backend* (motor xat bot) i el *frontend* (interfície d'usuari).

Pel que fa aquesta interfície d'usuari el resultat és molt satisfactori, ja que és agradable per a l'usuari, intuïtiva i fàcil d'utilitzar. Fins i tot gràcies a la Plataforma *Chatwoot* s'ha pogut afegir un missatge per al primer cop que els usuaris entren a la plataforma on els ajuda amb un breu *tutorial* i

s'ha activat també una breu enquesta de satisfacció per tal d'obtenir feedback dels usuaris que la utilitzen.

El primer que ha de fer el robot és identificar pel que se li està preguntant. Gràcies a les diferents llistes de captació d'informació aconseguim captar de manera satisfactòria la

intenció de l'usuari (vull informació, necessito ajuda, et saludo, em despedeixo, etc) les entitats que acompanyen aquesta intenció en el cas que sigui petició d'informació (necessito informació sobre exàmens, horaris, tfg, pràctiques, etc) i finalment, les subentitats, temes que contemplen cada una de les entitats estàtiques (duració del tfg, propostes del tfg, professors del tfg, entregues del tfg) que només se'ls hi farà cas en el moment que l'entitat mare sigui el tfg, en l'exemple anterior.

En les següents il·lustracions es mostren converses reals amb el robot. La Fig. 10 se li pregunta pel tfg i les diferents preguntes que se'n poden derivar (subentitats).

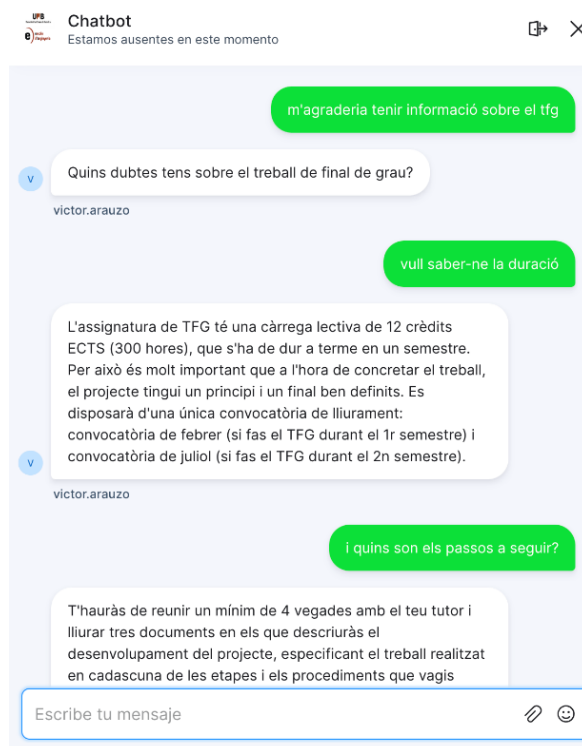


Fig 10 Exemple 1, petició d'informació sobre el treball final de grau.

En la Fig. 11 s'emmarca en l'àmbit d'un inici de conversa i una petició d'uns horaris per part de l'usuari.

Finalment, l'usuari demana les guies docents de diferents assignatures en la Fig. 12.

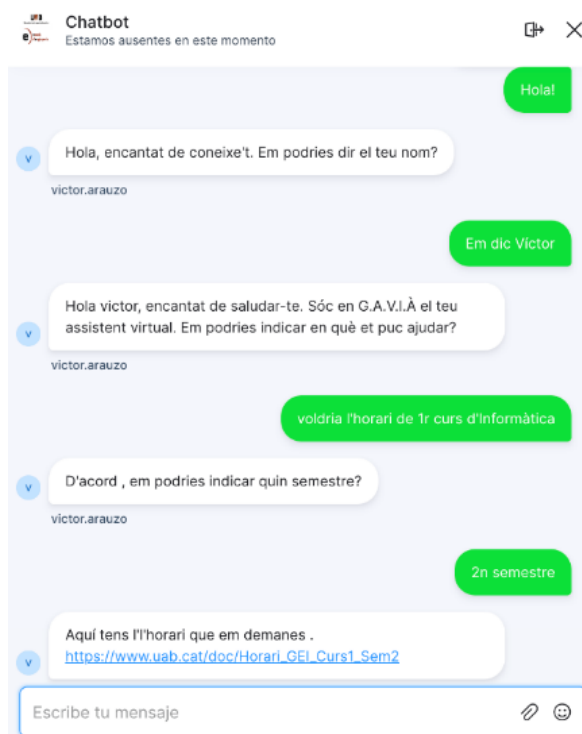


Fig 11 Exemple 2, petició d'horaris.



Fig 12 Exemple 3, petició de l'usuari de guies docents.

El robot és capaç de parlar i entendre 2 llengües diferents; català i castellà. No són auto detectades, sinó que l'usuari és l'encarregat de què en qualsevol moment de la conversa, pot canviar aquesta llengua de comunicació amb una

paraula clau. Això s'ha aconseguit gràcies a què el programa té fitxers de text tant per identificar el tema de les qüestions de l'usuari, com per donar una resposta a aquestes peticions. D'aquesta manera seguint la mateixa estructura per a tots els llenguatges, s'ha pogut adaptar la interacció amb els diferents llenguatges dotant al programa d'una internacionalització característica i una fàcil localització per a futures llengües com anglès.

Finalment, les respostes que dona el robot són acurades i no repetitives, i és capaç de demanar informació que li falta si no pot donar una resposta prou precisa.

En el moment de la generació es dota a l'elecció de la frase d'una aleatorietat que fa que a una mateixa pregunta no es respongui en tots els casos de la mateixa manera, sinó que es tria entre una llista de respostes per tal que l'usuari tingui una experiència més realista.

4 CONCLUSIONS

Entrant a l'apartat de conclusions, es valoraran si els objectius s'han assolit satisfactòriament, quin valor afegit aporta el robot en l'entorn de la universitat, els diferents problemes que han sortit durant el desenvolupament i les solucions proposades i finalment, el benefici que en treuen els estudiants amb l'ús del mateix.

L'objectiu principal del treball era crear un xat bot funcional que resolgués els dubtes i peticions dels estudiants mitjançant una interfície de text emmarcat dins d'una pàgina web. Com hem comentat en l'apartat anterior, aquest objectiu s'ha assolit satisfactòriament i el robot és capaç de mantenir una conversa i donar resposta precisa als dubtes que se li plantegen.

Un altre dels objectius que s'havien proposat era fer dues versions del robot, una mitjançant l'ús de paraules claus i l'altra mitjançant models d'aprenentatge autònom. Amb aquestes dues tècniques funcionant, es pretenia fer una comparació dels resultats. Aquest objectiu no s'ha pogut assolir, principalment per dues raons: Una planificació massa optimista de la primera versió del robot (versió utilitzant paraules clau) ha fet que les tasques s'allarguin més del previst i no haver tingut les suficients hores per tal de desenvolupar aquesta segona versió (versió amb models de predicció).

L'altre factor clau en el no assoliment d'aquest segon objectiu ha sigut la necessitat d'una gran quantitat de dades de preguntes i respostes per tal d'entrenar el model d'aprenentatge autònom i que ha estat inexistente. Es va pensar en generar aquesta base de dades de manera sintètica, però comportava una gran quantitat d'hores de treball que no es tenien.

Els principals contratemps sorgits en el projecte han sigut respecte al desconeixement i la poca fluïdesa del principi en utilitzar eines que mai s'havien utilitzat, com la integració entre serveis (codi amb el servei de *Chatwoot* i *Heroku*). Així ha esdevingut un desenvolupament menys fluid al principi del procés que s'ha solucionat amb les hores

invertides per tal de guanyar experiència.

També ha tingut un punt de complexitat el com plasmar tota la informació que havia de transmetre el robot mitjançant el sistema de preguntes i respostes i que la conversa esdevingués fluida i amb sentit com podria ser la interacció amb una persona. Aquesta tasca s'ha pogut duu a terme consultat diverses fonts expertes en el tema, per tal d'aconseguir la millor experiència en quant a interacció màquina-humà, sovint consultat també material de diferents llocs web especialitzats en xat bots comercials.

Com a principals beneficis del projecte de cara a la comunitat d'estudiants i professors de l'Escola d'Enginyeria de la universitat, el robot aporta un servei extra que pretén alliberar de la tasca de respondre a les preguntes dels alumnes que duen a terme als coordinadors. Aquest bot contesta automàticament als dubtes més freqüents que puguin sorgir i de manera simultània, de manera que si estigués implementat a la web de l'Escola seria una eina útil per a tothom.

4.1 Línies de continuació

Com a línies de continuació del projecte, la primera que es proposa seria dotar a l'aplicatiu d'una robustesa i una conversacionalitat en un grau més, de manera que el bot millori en precisió i exactitud per a qualsevol tipus d'entrada de l'usuari. Això és podria desenvolupar mitjançant un període de proves on s'omple un registre de les peticions d'un grup d'usuaris per posteriorment analitzar-lo i poder millorar l'efectivitat del bot a través d'una fase de *testeig* d'entrades d'usuaris reals.

Una altra possible línia de continuació, seria dotar a l'aplicatiu d'un sistema on els coordinadors i professors de l'Escola poguessin actualitzar algunes informacions que varien any rere any, de manera que el bot tingués un manteniment més fàcil i fos més útil a mesura que passen els cursos acadèmics. En aquesta plataforma hi entrarien directament el personal acadèmic de manera que el manteniment del desenvolupador podria centrar-se en millorar el robot i no en actualitzar la informació.

Finalment, una última línia de continuació proposada, seria la implementació del robot a nivell de tota la universitat, el que requeriria d'una escalabilitat i d'una nova injecció de nous mòduls per encapsular tota la informació i requeriments dels altres graus de la UAB.

AGRAÏMENTS

Aquest treball de desenvolupament del *software* ha sigut realitzat sota la supervisió de la Dra. Marta Prim Sabrià a qui m'agradaria expressar el meu agraïment pel seu tracte, comprensió i paciència durant totes les etapes d'aquest projecte.

També volia fer especial menció al meu company de feina i amic Sr. Joan Salvatella, que ha estat durant tot el procés pendent i m'ha mentoritzat en els moments més complicats del treball.

Finalment, agrair el suport rebut en tot moment per la

meva família, tant anímicament com econòmicament, en especial a la meva mare, per la paciència i els valors que sempre m'ha ensenyat, podent dur a terme aquest projecte en les etapes finals d'aquest grau universitari.

BIBLIOGRAFIA

[1] Zendesk & Salesforce Service Cloud, *How to create website channel?* update: April. 19, 2021. Accessed on: March, 2022.[Online]Available: <https://www.chatwoot.com/docs/product/channels/live-chat/create-website-channel>

[2] Zendesk & Salesforce Service Cloud, *API documentation* update: Nov. 19, 2021. Accessed on: March, 2022. [Online]. Available: <https://www.chatwoot.com/developers/api/>

[3] L. Fischer, *A Beginner's Guide to HTTP and REST*, en-vato-tuts+ ,update: Dec. 4, 2016. Accessed on: Feb, 2022. [Online]. Available: <https://code.tutsplus.com/tutorials/a-beginners-guide-to-http-and-rest-net-16340>

[4] *chatwoot-API*, Postman, update: Nov. 7, 2017. Accessed on: Feb, 2022. [Online]. Available: <https://documenter.getpostman.com/view/3378293/SztD5n9K#intro>

[5] Sojan-official, *Chatwoot Agent Bot Implementation using DialogFlow*, Chatwoot Developers ,update: Apr. 1, 2022. Accessed on: Feb, 2022. [Online]. Available: <https://github.com/chatwoot/dialogflow-agent-bot-demo>

[6] Joseph Weizenbaum. ELIZA – A Computer Program For the Study of Natural Language Communication Between Man And Machine. [Internet]. Communications of the ACM. 1 Gener 1966. [Consultat 20 Abril 2022]; Disponible en: <http://web.stanford.edu/class/cs124/p36-weizenbaum.pdf>

[7] Victoria Pintos. ¿Qué tipos de chatbots existen y cuál es mejor para tu negocio?. inConcert. [Internet]. 6 Ago 2019. [Consultat 15 Set 2021].

[8] Morgan Newman – Medium - How to Keep Your Free Heroku App Online Forever [Internet]. 3 Nov 2018. Accessed on 1 Juny 2022 Available <https://medium.com/@morgannewman/how-to-keep-your-free-heroku-app-online-forever-4093ef69d7f5>

[9] Guzman Gomez – Planeta Chatbot -Chatbots: algoritmos, intenciones, entidades e historias [Internet]. 11 Oct 2019. Accessed on 30 Març 2022 Available <https://planetachatbot.com/chatbots-algoritmos-intenciones-entidades-e-historias/#NLU>

[10] Oracle. ¿Qué es un chatbot?. [Internet]. Consultat 18 Maig 2021. Available: <https://www.oracle.com/es/chatbots/what-is-achatbot/>

A1. ARBRE D'INTENCIONS, ENTITATS I SUBENTITATS DE LA WEB DE L'ESCOLA

Arbre creat durant les primeres fases del projecte per implantar estructura de les possibles preguntes amb les seves respostes de la web de l'escola.

[illegible]

Fig 12 Esquema arbre temàtiques Escola Enginyeria

