
Estudi i creació d' un assistent virtual (xatbot).

Anesti Álvarez Ruiz.
Enginyeria Informàtica (2n. cicle).
Intel·ligència artificial.

Consultor: Jordi Delgado Pin.

Gener 2018.

*Per a la Mire, l' Ona i
l' Anes.*



Aquesta obra està subjecta a una llicència de [Reconeixement-NoComercial-SenseObraDerivada 3.0 Espanya de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

FITXA DEL TREBALL FINAL

Títol del treball:	<i>Estudi i creació d' un assistent virtual (xatbot)</i>
Nom de l'autor:	<i>Anesti Álvarez Ruiz</i>
Nom del consultor/a:	<i>Jordi Delgado Pin</i>
Data de lliurament (mm/aaaa):	<i>01/2018</i>
Titulació o programa:	<i>Enginyeria Informàtica (2n. cicle)</i>
Àrea del Treball Final:	<i>Intel·ligència artificial</i>
Idioma del treball:	<i>Català</i>
Paraules clau	<i>AIML, bot</i>

Resum del Treball

Com a projecte de final de carrera he realitzat l'anàlisi, disseny e implementació d'un bot conversacional que interpreta comandes AIML.

L'aplicació web desenvolupada pretén donar recolçament als pares que s'enfronten al procés de realitzar la preinscripció escolar dels seus fills. El problema és que la primera vegada que han de realitzar aquest procés, les famílies tenen molts dubtes i desconeixen els processos o passes a seguir, què és l'àrea d'influència, quins centres et donen punts i quins no, si tens cap avantatge, les característiques de les diferents escoles o centres, etc. En el nostre cas, el bot estarà especialitzat en respondre a consultes sobre el procés de preinscripció escolar a l'àrea de Barcelona metropolitana.

A més a més del bot, l'aplicació inclou un mòdul específic per a gestionar-lo i des d' on es podrà consultar la seva activitat, es podran resoldre preguntes que no és capaç de respondre i incloure-les en la seva base de dades de coneixement (afegint patrons AIML)

Per a la seva elaboració s'ha aplicat el model de cicle de vida en cascada, realitzant l'anàlisi, disseny, implementació i proves del sistema desenvolupat.

Abstract

I have done the analysis, design and implementation of a conversational bot that interprets AIML commands in this project.

The bot gives support to parents who have problems during registering children's school process. The parents don't have knowledge to perform this task successfully and they have many questions: what is the area of influence?, How many points do we need?, What is my influence area? What is the characteristic of each school?, etc. The bot is specialized in solving doubts around the metropolitan area of Barcelona.

In addition to the bot, the application includes a specific module to manage it. You can check the activity's bot. You can know how many questions the robot has been able to answer. Also it can work with questions not solved and insert them into a data base (adding AIML patterns)

For its elaboration, the cascade life cycle model has been applied, carrying out the analysis, design, implementation and testing of the developed system.

Índex

1. Introducció.....	1
1.1 Context i justificació del Treball.	1
1.2 Objectius del Treball.....	1
1.3 Enfocament i mètode seguit.	2
1.4 Planificació del Treball.....	3
1.5 Productes obtinguts.....	7
1.6 Estructura de la memòria.	8
2. Anàlisi i disseny.....	9
2.1 Anàlisi de requeriments.	9
2.2 Estructura del AIML.....	10
2.3 Casos d'ús.....	12
2.4 Diagrames UML.	16
2.5 Model de dades.....	19
3. Implementació.....	20
4. Conclusions.....	27
5. Glossari	29
6. Bibliografia.....	34
7. Annexos	35

Llistat de figures

Il·lustració 1 – Tasques diagrama Gantt projecte.	4
Il·lustració 2 – Diagrama Gantt projecte.	5
Il·lustració 3 – Diagrama Gantt replanificat.	7
Il·lustració 4 – Casos d'ús.	13
Il·lustració 5 – Diagrama conversa bot.	17
Il·lustració 6 – Diagrama accés administració.	18
Il·lustració 7 – Model de dades.	19
Il·lustració 8 – Pantalla inici xat index.php.	20
Il·lustració 9 – Pantalla inici administració admin/web/index.php.	24
Il·lustració 10 – Tecnologies utilitzades.	35
Il·lustració 11 – Instal·lació : setup XAPP.	36
Il·lustració 12 – Instal·lació : components XAPP.	36
Il·lustració 13 – Instal·lació : inici instal·lació XAPP.	37
Il·lustració 14 – Instal·lació : panel control XAPP.	38
Il·lustració 15 – Instal·lació : panel administració base de dades.	38
Il·lustració 16 – Instal·lació : prova bot.	42
Il·lustració 17 – Mòdul del bot.	43
Il·lustració 18 – Exemple xat.	43
Il·lustració 19 – Consulta d'estadístiques.	44
Il·lustració 20 – Anàlisi de preguntes no resoltes.	45
Il·lustració 21 – Afegir resposta.	45

1. Introducció.

1.1 Context i justificació del Treball.

Els assistents virtuals o bots automàtics són actualment una realitat. Aquests bots són capaços de respondre dubtes i preguntes que se'ls hi plantegin. Inclús aquest tipus de programes estan sovint dissenyats per a simular un comportament humà, com podria ser una conversa.

D'altra banda, el procés d'escolarització dels infants és un moment molt important on apareixen molts dubtes ja que per a poder realitzar la preinscripció cal tenir molta informació nova pels pares, com per exemple la zona escolar a la qual pertany, saber quines escoles et toquen per proximitat, si són públiques, concertades o privades (i què significa això), com determinar la selecció de les preferències de les escoles, quina puntuació inicial tens, quines qualitats donen puntuació extra, quins requisits has de complir per a poder demanar una beca de menjador o de transport, etc.

Aquest projecte pretén oferir una eina automàtica pels pares que es troben en aquesta situació i que puguin resoldre els seus dubtes a mida que recerquen la informació per a poder realitzar la preinscripció escolar amb les millors garanties evitant el neguit que suposa no saber què et demanen, no saber on trobar respostes o trobant respostes en xats que són contradictòries.

1.2 Objectius del Treball.

L'objectiu principal d'aquest projecte consisteix en desenvolupar un assistent virtual o xatbot d'ajuda que doni suport i respongui els dubtes dels pares que estan cercant informació sobre com fer la preinscripció escolar del seu fill o filla. En el nostre cas, el bot estarà especialitzat en respondre a consultes sobre el procés de preinscripció escolar a l'àrea de Barcelona metropolitana.

El bot es basarà en el paradigma estímul-resposta escrit mitjançant patrons que fan servir el llenguatge AIML (Artificial Intelligence Mark-up Language). Aquest llenguatge està especialment indicat per a la creació d'agents amb llenguatge natural. Per això, la part més important d'aquest projecte serà implementar un intèrpret capaç d'interpretar el llenguatge AIML.

A més, perquè el bot sigui capaç de funcionar s'hauran de crear els paràmetres AIML amb totes les preguntes i respostes amb les que haurà de treballar el bot.

Un altre objectiu d'aquest projecte es posar a prova els coneixements i habilitats adquirides en el transcurs de la carrera d'una forma pràctica i ser capaç de plasmar-ho en una aplicació. L'elaboració d'aquest projecte em donarà l'oportunitat d'aprendre coses noves, no fetes durant la carrera.

1.3 Enfocament i mètode seguit.

Avui en dia els bots conversacionals gaudeixen d'una àmplia popularitat. Totes les companyies d'Internet han desenvolupat els seus propis framework per a implementar bots de forma senzilla; IBM ofereix Watson, Google té dialogflow, Amazon per la seva banda ofereix Lex, etc. Tot i que de disseny i aspecte molt revolucionari, totes es basen en la mateixa idea: la implementació de patrons pregunta-resposta.

Malgrat que han definit el seu propi llenguatge per a integrar els patrons pregunta-resposta, el funcionament principal és el mateix: interpretar el llenguatge natural de l'usuari per a posteriorment, buscar en una base de coneixement de patrons pregunta-resposta la informació que més s'ajusti a l'entrada de l'usuari i entregar la resposta més coherent. A més a més, aquesta base de coneixement ha de ser creada i integrada per l'administrador del bot. I el mateix passa amb l'entrenament i

l'aprenentatge del bot, que en tot moment ha de ser assistit i supervisat per l'administrador del bot.

Una vegada analitzats els frameworks que he comentat anteriorment, decideixo implementar un bot utilitzant AIML ja que aquest llenguatge està especialment indicat per a la creació d'agents amb llenguatge natural. Per això, la part més important d'aquest projecte serà implementar un intèrpret capaç d'interpretar el llenguatge AIML

El llenguatge AIML (Artificial Intelligence Mark-up Language) és un llenguatge de programació basat en XML desenvolupat pel Dr. Richard Wallace entre l'any 1995 i l'any 2000. Tot i que descrit molt àmpliament, aquest llenguatge AIML està especialitzat en la creació d'agents software amb llenguatge natural. La definició d'aquest procés es porta a terme a través d'una sèrie de categories AIML que són les encarregades de definir una reacció de part del bot. Per a això l'AIML conté elements que defineixen com el bot ha de buscar un patró de coincidència amb una frase o paraula.

1.4 Planificació del Treball.

Per a implementar aquest projecte s'ha decidit seguir la metodologia en cascada ja que és la que considero que millor s'ajusta a les necessitats del projecte.

En aquesta metodologia s'ordena rigorosament les etapes del procés per al desenvolupament de software, de tal forma que l'inici de cada etapa ha d'esperar a la finalització de l'etapa anterior.

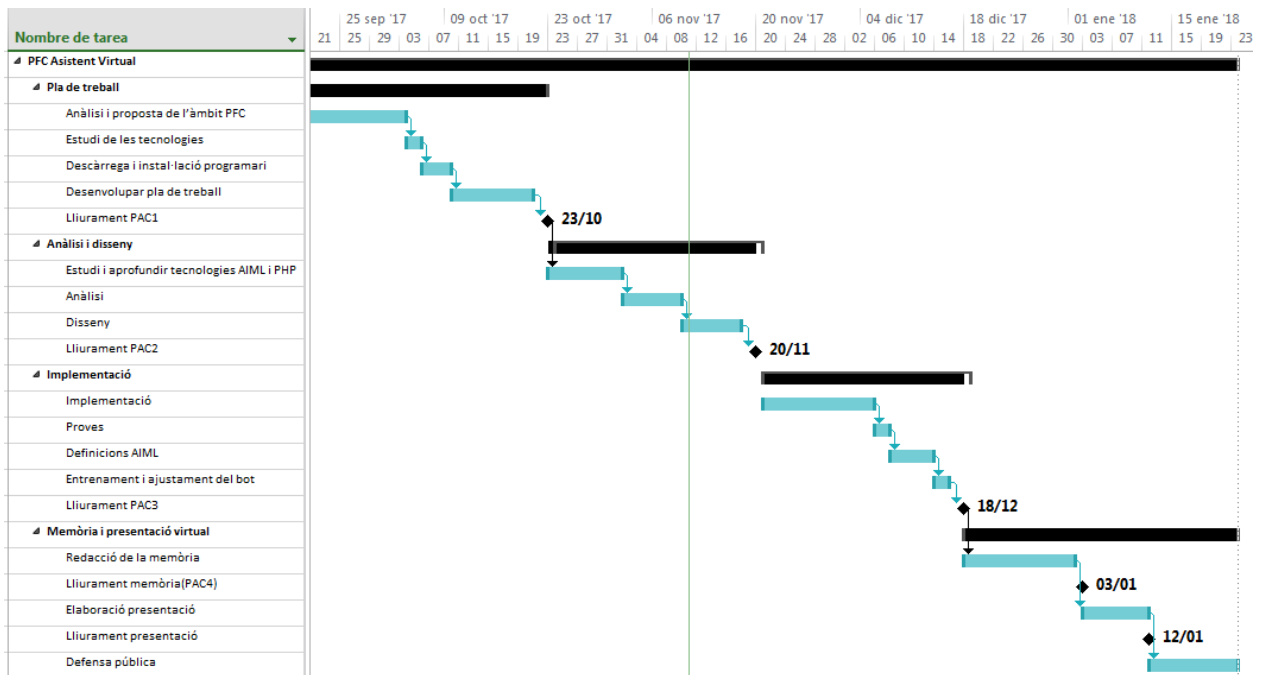
El projecte es distribueix en les següents quatre fases o feines:

FASE	INICI	FINAL
Pla de treball	20/9/2017	23/10/2017
Anàlisi i Disseny	24/10/2017	20/11/2017
Implementació	21/11/2017	18/12/2017
Memòria i presentació virtual	19/12/2017	23/01/2018

El diagrama de Gantt associat al projecte és el següent:

Nombre de tarea	Duració	Comienzo	Fin	Predecesoras
▲ PFC Asistent Virtual	90 días	mié 20/09/17	mar 23/01/18	
▲ Pla de treball	23 días	mié 20/09/17	lun 23/10/17	
Anàlisi i proposta de l'àmbit PFC	10 días	mié 20/09/17	mar 03/10/17	
Estudi de les tecnologies	2 días	mié 04/10/17	jue 05/10/17	3
Descàrrega i instal·lació programari	2 días	vie 06/10/17	lun 09/10/17	4
Desenvolupar pla de treball	9 días	mar 10/10/17	vie 20/10/17	5
Lliurament PAC1	0 días	lun 23/10/17	lun 23/10/17	6
▲ Anàlisi i disseny	20 días	mar 24/10/17	lun 20/11/17	
Estudi i aprofundir tecnologies AIML i PHP	8 días	lun 23/10/17	mié 01/11/17	7
Anàlisi	6 días	jue 02/11/17	jue 09/11/17	9
Disseny	6 días	vie 10/11/17	vie 17/11/17	10
Lliurament PAC2	0 días	lun 20/11/17	lun 20/11/17	11
▲ Implementació	20 días	mar 21/11/17	lun 18/12/17	
Implementació	11 días	mar 21/11/17	mar 05/12/17	
Proves	2 días	mié 06/12/17	jue 07/12/17	14
Definicions AIML	4 días	vie 08/12/17	mié 13/12/17	15
Entrenament i ajustament del bot	2 días	jue 14/12/17	vie 15/12/17	16
Lliurament PAC3	0 días	lun 18/12/17	lun 18/12/17	17
▲ Memòria i presentació virtual	27 días	lun 18/12/17	mar 23/01/18	
Redacció de la memòria	11 días	lun 18/12/17	lun 01/01/18	18
Lliurament memòria(PAC4)	0 días	mié 03/01/18	mié 03/01/18	
Elaboració presentació	7 días	mié 03/01/18	jue 11/01/18	20
Lliurament presentació	0 días	vie 12/01/18	vie 12/01/18	
Defensa pública	8 días	vie 12/01/18	mar 23/01/18	22

Il·lustració 1 – Tasques diagrama Gantt projecte.



Il·lustració 2 – Diagrama Gantt projecte.

A continuació es detalla cadascuna de las fases del projecte:

1.4.1 Fase 1 (PAC1): Pla de treball.

Aquesta fase correspon al període comprès entre el 20 de setembre de 2017 fins al 23 d'octubre de 2017. En aquesta fase es realitzaran les següents activitats:

- Estudi, anàlisi i proposta de l'àmbit PFC i estudi de les possibles opcions.
- Començament de l'estudi de les tecnologies que es faran servir en el desenvolupament del projecte..
- Generació del document "Pla de Treball".

1.4.2 Fase 2 (PAC2): Anàlisi i disseny.

Aquesta fase correspon al període comprès entre el 24 d'octubre de 2017 fins al 20 de novembre de 2017. En aquesta fase es realitzaran les següents activitats:

- Anàlisi del projecte seleccionat.

- Disseny del projecte seleccionat.
- Estudi de les tecnologies que es faran servir en el desenvolupament del projecte (MariaDB, PHP, Apache, AIML)
- Generació del document “Anàlisi i Disseny”.

1.4.3 Fase 3 (PAC3): Implementació.

Aquesta fase correspon al període comprès entre el 21 de novembre de 2017 fins al 18 de desembre de 2017. En esta fase es realitzaran les següents activitats:

- Implementació del projecte a partir dels documents d’anàlisis i dissenys.
- Es continua amb l’estudi de les tecnologies que es faran en la implementació, sobre tot AIML.
- Fase de proves de l’interpret AIML.
- Definició dels fitxers AIML.
- Entrenament i ajustament del bot.
- Realització de documentació. Aquesta activitat inclou la realització de diferents documentacions a entregar: documentació d’instal·lació, manual de usuari, etc

1.4.4 Fase 4 (PAC4): Memòria i presentació virtual.

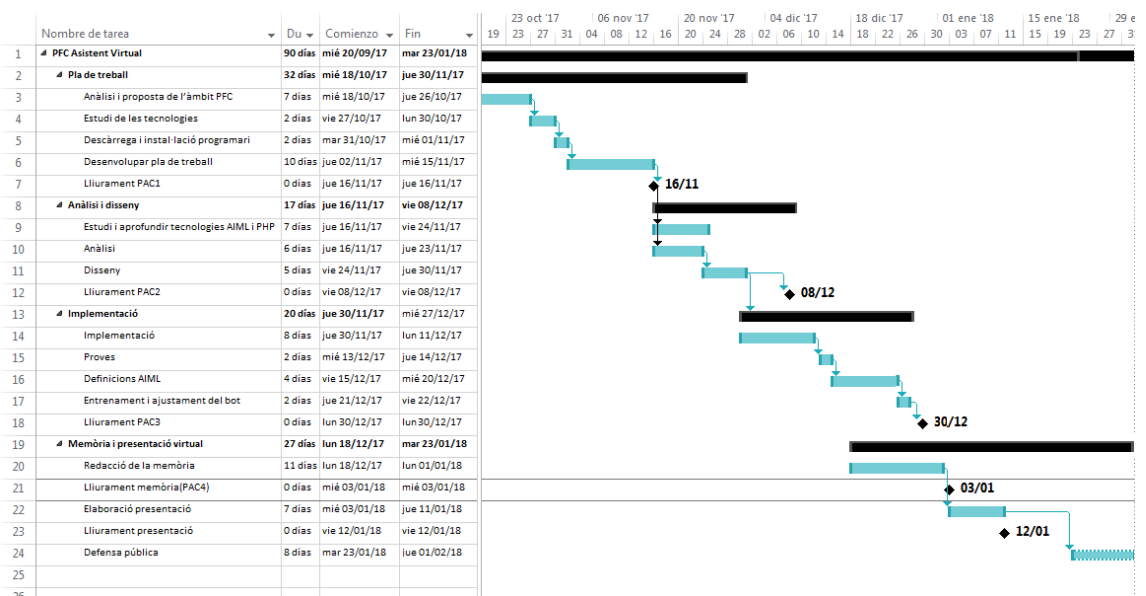
Aquesta fase correspon al període comprès entre el 18 de desembre de 2017 fins al 23 de gener de 2018. En aquesta fase es realitzaran les següents activitats:

- Desenvolupar la memòria del projecte. S’ha de realitzar una síntesi de totes les entregues anteriors i del projecte.
- Generació de la presentació virtual mitjançant un vídeo explicatiu, que inclourà una demostració de la seva utilització.
- Defensa del projecte.

Replanificació del projecte.

El desenvolupament del projecte ha anat amb un cert retard respecte el pla inicial anteriorment plantejat. Aquest retard té el seu origen en la fase de selecció de la temàtica del PFC. A partir d'aquí la resta de fases es van veure impactades i vaig haver de replantejar el pla de projecte. Per això vaig haver de fer un esforç per assolir les fites proposades al pla de treball. A més a més, per poder arribar correctament a les dates previstes, he hagut de fer diverses feines de forma paral·lela.

El diagrama de Gantt replanificat que he portat a terme és el següent:



Il·lustració 3 – Diagrama Gantt replanificat.

1.5 Productes obtinguts.

Al finalitzar el desenvolupament d'aquest projecte obtindrem els següents productes :

- Codi font amb les classes php.
- Scripts de creació dels objectes de base de dades necessaris.
- Entregables de les PACs mencionades anteriorment.

- Manual de l'usuari.
- Manual d'instal·lació.
- Memòria del projecte.
- Presentació del projecte.

1.6 Estructura de la memòria.

La present memòria està dividida en 7 capítols, dividits a la vegada en diferents apartats.

- Capítol 1. Introducció: aquest capítol estarà dedicat a la introducció del projecte, la motivació per a realitzar-lo i els objectius que es pretenen assolir.
- Capítol 2. Anàlisi i disseny: en aquest capítol es detalla l'anàlisi i disseny realitzats per a implementar l'aplicació.
- Capítol 3. Implementació: s'abordaran els temes més importants sobre l'estructura i desenvolupament de l'aplicació i els aspectes clau de cadascun d'ells.
- Capítol 4. Conclusions: estarà dedicat a exposar les conclusions finals del projecte.
- Capítol 5. Glossari : es mostra la definició dels termes i acrònims més rellevants utilitzats en aquesta Memòria.
- Capítol 6. Bibliografia: es disposarà la informació consultada per a la realització del projecte i de la memòria.
- Capítol 7. Annexos: finalment, en aquest capítol es mostren els apartats que són massa extensos per a incloure en la memòria.

2. Anàlisi i disseny.

2.1 Anàlisi de requeriments.

Com ja s'ha comentat anteriorment, l'assistent virtual/bot que es va a implementar en aquest PCF té com a objectiu respondre a preguntes escrites en llenguatge natural i realitzades pels usuaris. Aquestes preguntes arribaran a l'assistent i aquest les processarà enviant una resposta adequada en funció de la seva base de coneixement. Les preguntes que els bots són capaços de respondre han de ser concretes i curtes.

Dins de l'àmbit d'aquest PFC s'implementarà un intèrpret d'AIML. Aquesta tasca és complexa, costosa i ambiciosa perquè es desenvolupa des de zero. Aquest intèrpret d'AIML serà l'encarregat de buscar i parsejar els fitxers AIML on es troba el coneixement del Bot i també de processar la informació introduïda per l'usuari i entregar una resposta coherent a la demanda feta.

El llenguatge AIML (Artificial Intelligence Mark-up Language) és un llenguatge de programació basat en XML desenvolupat pel Dr. Richard Wallace entre l'any 1995 i l'any 2000. Tot i que descrit molt àmpliament, aquest llenguatge AIML està especialitzat en la creació d'agents software amb llenguatge natural. La definició d'aquest procés es duu a terme a través d'una sèrie de categories AIML que són les encarregades de definir una reacció de part del bot. Per a això AIML conté elements que defineixen com el bot ha de buscar un patró de coincidència amb una frase o paraula.

Per a poder aconseguir un funcionament complet d'aquesta iniciativa, és necessari que l'aplicació implementi les següents tasques:

- Ser capaç de rebre un text d'un usuari i intentar interpretar-ho.
- Buscar la millor resposta a la pregunta de l'usuari seguint patrons pregunta-resposta d'AIML que contingui el bot. Aquesta acció la realitzarà l'intèrpret d'AIML.
- I finalment mostrar l'activitat del bot i les preguntes que no és capaç de respondre per si sol perquè en una fase posterior es puguin incorporar aquests patrons desconeguts i enriquir el coneixement del bot.

2.2 Estructura del AIML.

Com que el llenguatge AIML està basat en XML, és format per etiquetes. Dins de la definició de AIML hi ha disponible nombroses etiquetes que són les encarregades de donar la intel·ligència al bot.

Les etiquetes bàsiques i més importants que té el llenguatge són:

- **<aiml>**: és el tag d'inici i fi del document AIML.
- **<category>**: tag que marca una unitat de coneixement dins del bot.
- **<pattern>**: conté el patró que ha de coincidir amb allò que l'usuari introdueix al bot.
- **<template>**: conté la resposta a l'entrada d'un usuari.

A nivell d'estructura l'etiqueta **<aiml>** obre i tanca qualsevol document AIML. Cadascú d'aquests elements **<aiml>** ha de contenir com a mínim un element **<category>**.

Aquestes etiquetes **<category>** contenen la unitat del coneixement del bot i estan compostes per etiquetes **<pattern>** i **<template>**. La primera conté el patró introduït per l'usuari i la segona, la resposta que proporciona el bot.

Un exemple senzill de fitxer AIML és el següent:

```
<aiml>
  <category>
    <pattern>Hola</pattern>
    <template>Hola com et dius?</template>
  </category>
</aiml>
```

Unes altres etiquetes interessants d'AIML són les següents :

- **<star>** : serveix per introduir en el template els caràcters comodins, amb els quals s'extreuen dades de l'entrada de l'usuari per capturar-les i fer-les servir en la resposta del bot.

Per exemple `<star index= "n">` on s'indica que l'asterisc és el que es fa servir.

- **<srail>** : proporciona concurrència als documents AIML. És una etiqueta molt potent.
- **<random>**: permet donar respostes aleatòries, en un mateix pattern, el bot pot donar diverses respostes contingudes dins dels tags `<random>`.
- **<set>**, **<get>** y **<think>**: es fan servir per a poder treballar amb variables en AIML. En primer lloc, `<set>` es fa servir per assignar un valor a una variable que l'obté del pattern introduït per l'usuari o que també pot ser creada i definida per l'interpret AIML.

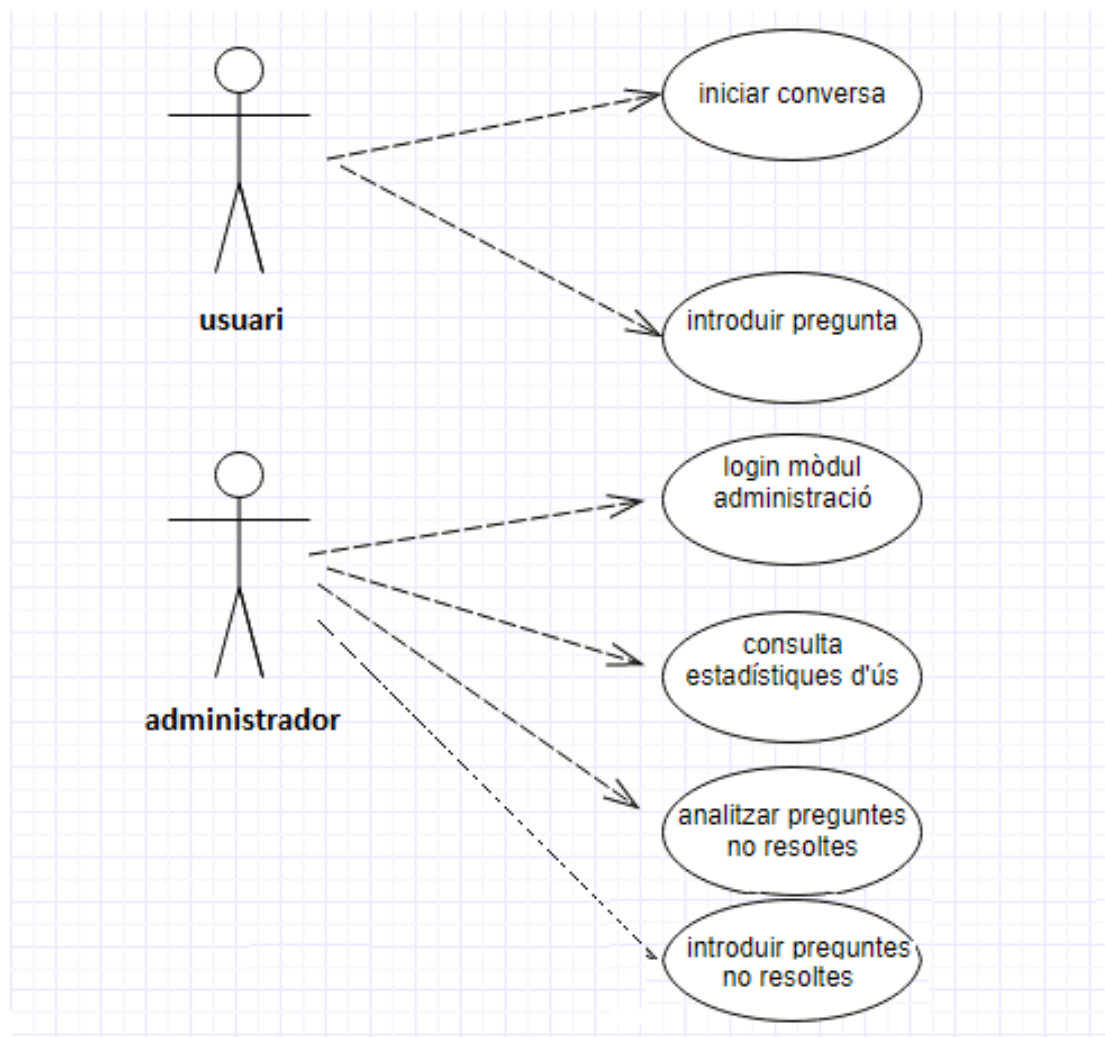
D'altra banda, `<get>` obté el valor d'una variable i la fa servir en el missatge mostrat a l'usuari. La variable que fa servir `<get>` pot haver estat capturada per un set d'un pattern diferent al que genera la variable amb `get`.

Finalment, <think> és com get però quan captura una variable no la mostra al missatge de l'usuari.

- **<that>**: ho fa servir el bot per contestar a una resposta que dona l'usuari. Captura la resposta que l'usuari introdueix a una pregunta del bot i en funció d'aquesta resposta el bot contesta l'usuari.
- **<topic>**: es fa servir per emmagatzemar un tema/contexte que només ha estat pensat si es tracta d'aquell tema, per això acostuma a ser molt específic.
- **<condition>**: serveix per comparar el valor d'una variable amb un valor donat. Si aquest coincideix es dona la resposta de la condició. És com les sentències "if" d'altres llenguatges de programació.

2.3 Casos d'ús.

A partir de la identificació dels actors que intervenen en aquesta aplicació, els següents casos d'ús descriuen totes les funcionalitats de l'aplicació.



Il·lustració 4 – Casos d'ús.

Així l'aplicació comptarà amb dos perfils d'accés :

- Usuaris: tots aquells que fan preguntes al bot.
- Administradors: qui accedeix a l'àrea d'administració del bot per consultar l'activitat del bot i sobre tot per analitzar les preguntes que el bot no ha pogut respondre basant-se en la seva base de dades de coneixement i que per tant, són susceptibles de ser introduïdes en el bot per a millorar les seves capacitats.

Descripcions Casos d'ús:

Codi:	C001
Nom:	Iniciar conversa.
Actors:	Usuari.
Funció:	Permetre l'accés al bot.
Descripció:	L'usuari navegant des del web accedeix a l'àrea de conversa del bot i activa la conversa.
Referències:	

Codi:	C002
Nom:	Introduir pregunta.
Actors:	Usuari.
Funció:	Realitzar conversa amb el bot.
Descripció:	L'usuari introdueix una pregunta en l'àrea de text i prem "enter". El bot busca en els patrons AIML la resposta més adequada i la publica a l'àrea del bot perquè la pugui visualitzar l'usuari.
Referències:	

Codi:	C003
Nom:	Login mòdul administració.
Actors:	Administrador.
Funció:	Limitar l'accés només a usuaris administradors.
Descripció:	L'usuari accedeix a l'àrea d'administració i introdueix un codi d'usuari contrasenya. Si és correcte permetrà l'accés a les pantalles consulta estadístiques d'ús i preguntes no resoltes.
Referències:	C004, C005

Codi:	C004
Nom:	Consulta estadístiques d'ús.
Actors:	Administrador.
Funció:	Analitzar els KPIs d'ús del bot.
Descripció:	L'administrador en accedir a aquesta pantalla podrà consultar diversos indicadors d'ús del bot com ara quants usuaris han accedit, quantes preguntes ha contestat, etc.
Referències:	C003, C005

Codi:	C005
Nom:	Analitzar preguntes no resoltes.
Actors:	Administrador.
Funció:	Millorar les capacitats del bot.
Descripció:	En accedir a la pantalla es mostrarà un llistat de les preguntes que el bot no ha pogut respondre. Per això hi haurà un selector de temps amb el qual es seleccionarà el període que es vol consultar.
Referències:	C003, C004

Codi:	C006
Nom:	Introduir noves respostes.
Actors:	Administrador.
Funció:	Millorar les capacitats del bot.
Descripció:	Des de la pantalla d'anàlisi de preguntes no resoltes es pot seleccionar una pregunta i prement el botó d'afegir, s'obre un desplegable on l'administrador indicarà els components AIML pattern i template perquè es donin d'alta a la base de dades. A partir d'aquest moment el bot ja podrà respondre a aquesta pregunta.
Referències:	C005

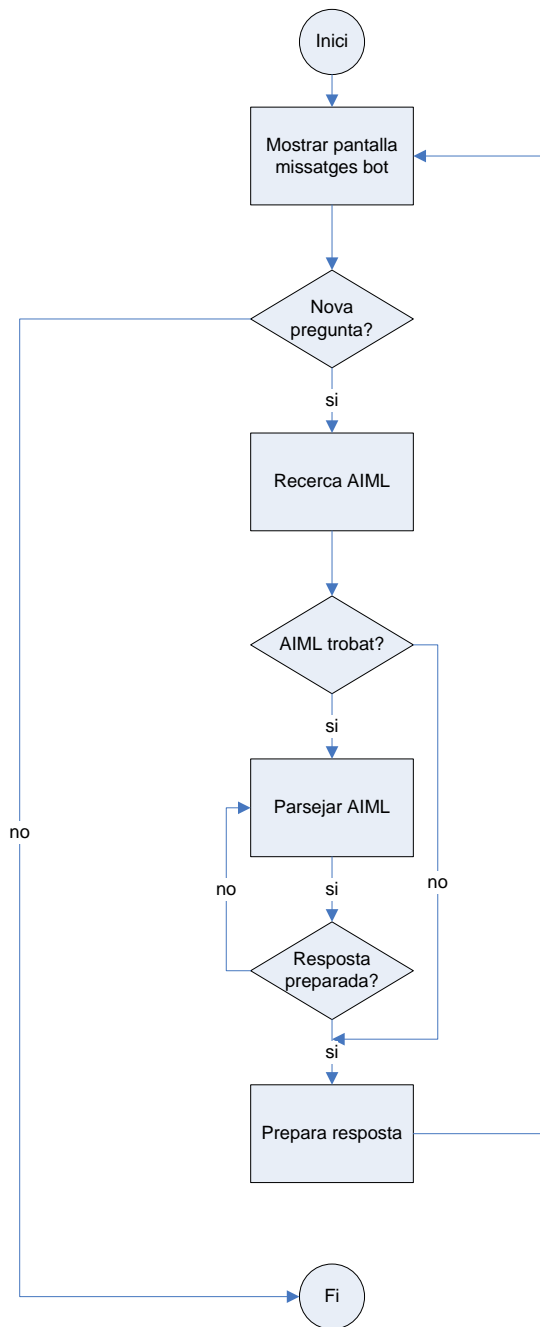
2.4 Diagrames UML.

En aquest apartat es descriuran els diagrames d'activitats on s'especifica el funcionament del bot i de l'aplicació que es desenvolupa.

2.4.1 Conversa del bot.

Aquest diagrama representa com es realitza la conversa entre l'usuari i el bot. L'usuari introdueix a través del teclat la pregunta. Quan pressiona el botó d'enviar, el client fa una trucada a la classe que refresca el text que veu l'usuari, afegint el text introduït. El servidor rep l'input de l'usuari, el processa i comença a buscar en la base de dades on estan els continguts dels patrons AIML aquell que s'ajusta a la frase introduïda per l'usuari. Aquesta tasca la fa l'interpret d'AIML.

Una vegada que troba la resposta, aplica les funcions AIML i envia un missatge http de resposta al client, que rep la resposta i la mostra a l'usuari. En el cas que no s'hagi trobat una resposta adequada, la pregunta es guarda a la base de dades per a la seva anàlisi posterior. I el client donarà la resposta que es mostrarà a l'usuari i que serà de l'estil "no he entès la pregunta", "en aquests moments no sé com resoldre el teu dubte", o similar. Una vegada que l'usuari veu la resposta pot seguir introduint més preguntes i seguir amb la conversa.



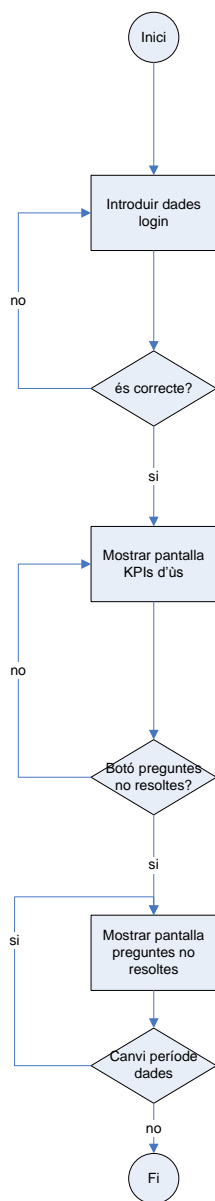
Il·lustració 5 – Diagrama conversa bot.

2.4.2 Accés administració.

En el següent diagrama d'activitat es mostra el procés d'administració del bot. Com a primer pas l'administrador accedeix a la url d'administració admin.php. En aquest moment s'executa una funció de login que bàsicament consisteix en introduir un codi d'usuari i contrasenya. El servidor rep aquest usuari/contrasenya i el compara amb

aquells que té a la base de dades i si existeix un client, s'enviarà un missatge d'OK i se li mostrarà la pantalla d'administració on podrà consultar les estadístiques d'ús del bot.

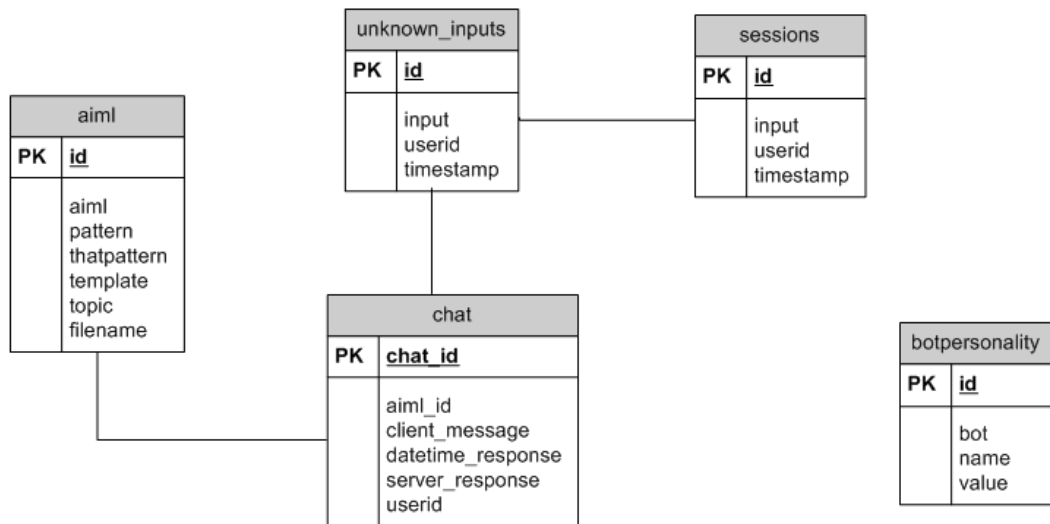
Si l'usuari prem el botó de preguntes no resoltes, el client enviarà aquest missatge al servidor i aquest accedirà a la base de dades recuperant les preguntes que durant l'últim període que el bot no ha pogut resoldre i les enviarà al client perquè aquest se les mostri a l'administrador.



II-lustració 6 – Diagrama accés administració.

2.5 Model de dades.

El model de base de dades que s'ha definit per al projecte és el següent:



Il·lustració 7 – Model de dades.

Taula aiml: conté els paràmetres AIML que el bot pot respondre.

Taula unknow_inputs: aquí es van guardant les preguntes que el bot no pot respondre. Més endavant, en el mòdul d'administració l'usuari les podrà analitzar i carregar-les a la taula aiml perquè les pugui respondre.

Taula sessions: aquí és on s'emmagatzemen totes les connexions que ha rebut el bot. És una taula d'auditoria de l'activitat per a poder consultar-la després al mòdul d'administració.

Taula chat: emmagatzema tota l'activitat del bot, les preguntes i respostes.

Taula botpersonality: conté els paràmetres de personalitat que li volem donar al bot.

3. Implementació.

3.1 Aplicació web.

L'aplicació web està dividida en 2 grans blocs o mòduls :

- El xat de conversa del bot.
- El mòdul d'administració.

A continuació explicarem el funcionament i els components que s'han desenvolupat per a cadascun d'aquests mòduls.

3.1.1 Mòdul xat.

Aquest mòdul consisteix en una pantalla tipus xat en la qual l'usuari introdueix una pregunta, el bot captura aquesta pregunta i la processa comparant-la amb la base de dades d'AIML, interpretant aquest llenguatge i buscant una resposta adequada a la pregunta de l'usuari.



Il·lustració 8 – Pantalla inici xat index.php.

La classe principal des de la qual s'accedeix a aquest mòdul és index.php que es troba en el directori arrel del servidor d'aplicacions. És l'encarregada de pintar la pantalla inicial del xat, assignant els javascrpts que s'encarregaran de la gestió dels objectes de la pantalla. També assigna els fitxers CSS, que són els responsables de les diferents visualitzacions de pantalla i dels ajustos per a ser compatible amb tots els navegadors i modes d'accés (ordenador, tauleta i mòbils).

```
<?php session_start(); ?>
<!DOCTYPE html>
<html lang="ca">
  <head>
    <meta name="theme-color" content="#809FBC" />
    <meta http-equiv="Cache-Control" content="max-age=3600">
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1, user-scalable=no" />
    <meta name="robots" content="noindex">

    <link rel="icon" type="image/ico" href="/img/iconos/favicon.png" />
    <link rel="icon" type="image/x-icon" sizes="30x30" href="/img/iconos/favicon.png" />

    <title>Uocbot</title>

    <script src="/bower_components/webcomponentsjs/webcomponents-personal.js"> </script>
    <script src="/bower_components/webcomponentsjs/webcomponents-loader.js"></script>
    <link rel="import" href="/bower_components/iron-icons/iron-icons.html">

    <link rel="stylesheet" href="/fonts/roboto/fonts.min.css" type="text/css" />
    <link rel="stylesheet" href="/css/general.min.css">
  </head>
  <body>
    <div class="container_bg">
      <div class="cover"></div>
    </div>
    <div class="container_cv">
```

```

<div class="box_cv">
  <header>
    
  </header>
  <main>
    <div class="cont_chat"></div>
    <div class="cont_form">
      <div>
        <textarea rows="1"></textarea>
      </div>
      <div>
        <iron-icon icon="send"></iron-icon>
      </div>
    </div>
  </main>
</div>
</div>
<script src="/js/general.min.js"></script>
</body>
</html>

```

La resta de classes que es fan servir en aquest mòdul són:

- **Conexion.php:** la seva funció és establir la connexió amb la base de dades.
- **Response.php:** crea una classe per a contenir els paràmetres de les pantalles que després farà servir ajax.
- **Autoload.php :** conté una funció genèrica encarregada de carregar les classes php perquè després siguin utilitzades directament.
- **Data.php:** és la que s'encarrega de contenir les funcions i classes de l'interpret AIML. Processa el missatge de l'usuari, busca en els AIML i retorna la resposta a l'usuari.
- **Send-question.php:** és la classe principal que crida a la resta de classes. Rep la pregunta de l'usuari, carrega totes les classes, fa

la connexió a la base de dades, inicia la sessió, passa el text per ser processat, rep la resposta i la passa a la funció javascript encarregada de pintar-la en pantalla.

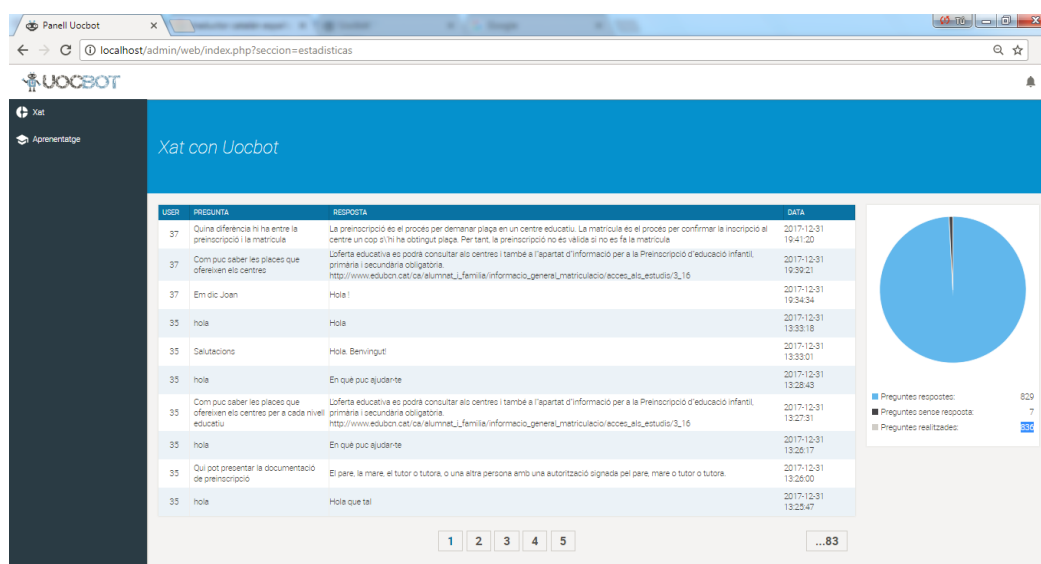
La gestió dels events de pantalla i el pintat de les preguntes i respostes, les realitza el javascript **general.min**.

Les funcions que el componen són:

- **constructor()** : funció encarregada de definir la pantalla de xat.
- **_initConver()** : funció encarregada d' iniciar la conversa i de mostrar el missatge de benvinguda.
- **_createLoading()**: funció encarregada de l'assignació de la icona de carregant.
- **_adjustHeights()**: funció encarregada d'ajustar reajustaments en l'alçada de la pantalla.
- **_scrollBottom()**: funció encarregada del control del scroll.
- **_setLoading()**: funció encarregada de pintar el missatge d'espera de càrrega.
- **_remLoading()**: funció encarregada d'esborrar el missatge d'espera de càrrega.
- **_autosize()**: funció encarregada de gestionar el reajustament de mida del navegador.
- **_sendQuestion()**: funció encarregada de pintar l'àrea de xat del missatge de l'usuari, envia les dades per ajax per a no refrescar la pantalla i truca a la classe send-question.php
- **_printQuestion(question)**: funció encarregada d'imprimir la pregunta de l'usuari.
- **_printResponse(response)**: funció encarregada d'imprimir la resposta del bot.
- **_listeners()**: funció encarregada de gestionar l'àrea d'escriptura perquè no es puguin injectar dades no controlades, ni salts de línia.

3.1.2 Mòdul administrador.

Amb el mòdul d'administració assolit, l'usuari accedirà a unes pantalles de consulta d'estadístiques d'ús del bot. Així per exemple podrà veure quantes preguntes se li han fet, quantes ha contestat bé, quantes no han robat resposta, etc. I des d'aquest mòdul d'administració també es podrà arribar a una altra pantalla on es poden consultar les preguntes que el bot no ha pogut contestar. I un cop han estat analitzades des d'una altra pantalla, l'usuari les podrà donar d'alta a la base de dades de coneixement del bot, perquè des d'aquests moment les pugui contestar. Per fer aquesta tasca, l'administrador haurà de tenir coneixements d'AIML.



II-lustració 9 – Pantalla inici administració admin/web/index.php.

Aquest mòdul està dividit en dos apartats:

- Estadístiques: en aquest apartat es pot consultar l'activitat d'ús del bot.
- Aprentatge: des d'aquest apartat es poden consultar les preguntes que el bot no ha pogut respondre. També podem donar d'alta a aquestes preguntes no respostes per a integrar-les a la base de coneixement del bot.

La classe principal des de la qual s'accedeix és index.php que es troba en el directori admin/web. Aquesta classe s'encarrega d'organitzar aquest mòdul d'administració i poder navegar entre la consulta d'ús del bot i l'anàlisi de les preguntes no respostes.

A continuació es descriuen les classes que componen aquests dos apartats.

Estadístiques :

- **Estadisticas.php**: la seva funció és establir la connexió amb la base de dades i crear la pàgina completa d'estadístiques, realitzar la paginació, etc.
- **Estats-preguntas.php**: la seva funció és la de pintar la gràfica de dades.

Aprentatge:

- **Aprendizaje.php**: la seva funció és establir la connexió amb la base de dades per a obtenir les preguntes que no s'han contestat per a poder pintar-les.
- **New-amil.php** : conté el formulari per a crear un nou AIML.
- **Save-aiml** : conté el formulari per a afegir un patró AIML d'una pregunta no resposta.

Els javascript que gestionen les trucades ajax i les pantalles d'aquests dos apartats contenen les següents funcions:

- **sendResponse ()**: envia el formulari de pregunta sense respondre.
- **newPattern ()**: envia i obre el formulari per a crear un nou AIML.
- **_openResponse()**: obre el formulari per a afegir un AIML d'una resposta no resolta.
- **setSlider()**: permet que la gràfica d'estadístiques sigui lliscant.
- **loadGrafica()**: captura les dades per a pintar-les a la gràfica i mostra les dades estadístiques afegides.
- **_printPieQuestions()**: la seva funció és pintar la gràfica.

- **_listeners():** la seva funció és esperar acció sobre els botons de la pantalla per a obrir el formulari que es necessita.

4. Conclusions.

Aquest projecte ha estat una bona prova per saber si havia assolit satisfactòriament els coneixements adquirits durant la carrera.

A més m'ha servit per aprendre a desenvolupar una aplicació web completa i per aprofundir en els meus coneixements d'HTML, PHP, javascript i SQL. D'altra banda, també he hagut d'aprendre HTML5 i Ajax, conceptes que desconeixia fins ara, però que actualment en programació web són totalment necessaris.

L'estudi de l' AIML i les anàlisis dels bots conversacionals m'han servit per fer-me una idea d'aquesta part de la Intel·ligència Artificial que actualment és un camí on encara queda molt a fer.

Pel que fa a l'aplicació que he desenvolupat, compleix tots els objectius definits a l'inici del projecte i és una aplicació totalment funcional. Visualment, és agradable i compatible amb tots els exploradors i formats de visualització que es fan servir actualment com a ordenadors, tauletes i mòbils.

La falta de temps per a desenvolupar aquest bot no m'ha permès crear una base de dades totalment completa. Així com per a altres idiomes si que hi ha a internet molts exemples i fitxers AIML per al català no n'he trobat. Per això m'he focalitzat en el fet que el bot fos capaç de respondre a preguntes relatives al procés de matriculació escolar dels infants i deixant una mica de banda que el bot sigui capaç de contestar a preguntes generals.

La part d'entrenament del bot i el seu enriquiment és una fase molt més llarga i complexa de l'esperada inicialment. El llenguatge humà és molt complex i una mateixa pregunta es pot escriure de formes molt diverses pel que requereix de moltes sentències AIML.

El fet que l'elecció de l'àmbit del projecte s'enrederís va afectar a la planificació del projecte i ha impactat durant tota la realització. La sensació d'anar sempre per darrere de les dades estimades afegeix pressió a la feina. També penso que en la planificació el temps especificat per al desenvolupament del projecte o estava ben dimensionat i ha estat necessari més temps de l' inicialment previst.

Per finalitzar comentar que el fet de fer la documentació en català ha estat també un gran esforç per la meua part, ja que no és la meua llengua materna i ha estat tot un repte per a mi, fent-me sentir més orgullós encara del resultat obtingut.

5. Glossari

Tomcat.

Tomcat és un servidor web amb suport de servlets i JSPs. Inclou el compilador Jasper, que compila JSPs convertint-los en servlets.

El motor de servlets de Tomcat sovint es presenta en combinació amb el Servidor HTTP Apache o altres servidors web. Tomcat pot funcionar com a servidor web per si mateix. En els seus inicis, va existir la percepció que l'ús de Tomcat de forma autònoma era només recomanable per a entorns de desenvolupament i entorns amb requisits mínims de velocitat i gestió de transaccions. Avui dia ja no existeix aquesta percepció, i Tomcat és usat com a servidor web autònom en entorns amb alt nivell de tràfic i alta disponibilitat.

Ja que Tomcat va ser escrit en Java, això fa que funcioni en qualsevol sistema operatiu que disposi de la JVM.

PHP.

PHP és un llenguatge de programació interpretat que s'utilitza per a generar pàgines web de forma dinàmica. S'executa al cantó del servidor, per aquest motiu al navegador web ja l'hi arriba la pàgina en format HTML, no podent visualitzar-ne el codi php. Tot i que és menys popular, també pot utilitzar-se per generar interfícies de comandes i aplicacions executables amb interfície gràfica. Disposa de versions tant per a Windows com per a GNU/Linux, així com d'altres.

PHP es distribueix sota la llicència PHP, que la Free Software Foundation qualifica com a programari lliure. Les sigles són un acrònim recursiu de PHP: Hipertext Preprocessor, tot i que originalment eren les sigles de Personal Home Page Tools.

El codi font escrit en PHP és invisible al navegador i el client, ja que és el servidor el que s'encarrega d'executar el codi i enviar el seu resultat

HTML al navegador. Això fa que la programació en PHP sigui segura i de confiança.

MariaDB.

MariaDB és un sistema de gestió de bases de dades relacional multi-fil (multithread) i multi usuari, que usa el llenguatge SQL (Structured Query Language). Va ser impulsada per la comunitat, per tal de mantenir el seu estat lliure sota la GNU GPL, degut a la incertesa de l'estat de la llicència de MySQL ara que està sota la propietat d'Oracle. Així mateix, hi ha la intenció de mantenir una alta fidelitat amb MySQL, que garanteixi una capacitat de substitució "drop-in" per l'equivalència de la biblioteca binària i la compatibilitat amb les APIs i les ordres de MySQL

HTML.

Acrònim d'Hyper Text Markup Language, en català, "llenguatge de marcat d'hipertext", és un llenguatge de marcat que deriva de l'SGML dissenyat per estructurar textos i relacionar-los en forma d'hipertext. Gràcies a Internet i als navegadors web, s'ha convertit en un dels formats més populars que existeixen per a la construcció de documents per a la web.

El llenguatge HTML és un estàndard reconegut a tot el món i les seves normes defineixen un organisme sense ànim de lucre anomenat World Wide Web Consortium, més conegut com a W3C.

Com es tracta d'un estàndard reconegut per totes les empreses relacionades amb el món d'Internet, la mateixa pàgina HTML es visualitza de forma molt semblant en qualsevol navegador de qualsevol sistema operatiu. El propi W3C defineix el llenguatge HTML com "un llenguatge reconegut universalment i que permet publicar informació de forma global". Per convenció, els arxius de format HTML fan servir l'extensió .htm o .html.

HTML 5.

Acrònim d'HyperText Markup Language, versió 5, és la cinquena gran revisió del llenguatge bàsic de la World Wide Web, HTML. HTML 5 especifica dues variants de sintaxi per a HTML: un «clàssic» HTML (text / html), la variant coneguda com a HTML5, i una variant XHTML coneguda com a sintaxi XHTML5 que haurà de ser usada com a XML (XHTML) (application/xhtml+xml). Aquesta és la primera vegada que HTML i XHTML s'han desenvolupat en paral·lel. El desenvolupament d'aquest codi és regulat pel Consorci W3C.

HTML 5 estableix una sèrie de nous elements i atributs que reflecteixen l'ús típic dels llocs web moderns.

CSS.

Cascading Style Sheets (en català: Fulls d'Estil en Cascada) és un llenguatge de fulls d'estil utilitzat per descriure la semàntica de presentació (l'aspecte i format) d'un document escrit en un llenguatge de marques. La seva aplicació més comuna és dissenyar pàgines web escrites en HTML i XHTML.

CSS està dissenyat principalment per permetre la separació de contingut del document HTML de la presentació del document, incloent-hi elements com la disposició, colors, i fonts. Aquesta separació pot millorar l'accessibilitat al contingut, proporcionar més flexibilitat i control en l'especificació de característiques de presentació, permetre que múltiples pàgines comparteixin un format comú, i redueix complexitat i repetició en el contingut estructural (com per exemple al permetre disseny web sense taules). CSS també pot deixar la mateixa pàgina de marques ser presentada en estils diferents mitjançant mètodes de render diferents, com a la pantalla, en impressió, per veu i amb mecanismes tàctils amb sistemes Braille.

CSS especifica un esquema de prioritat per determinar quines regles d'estil s'apliquen si més d'una regla està associada amb un element en particular. En aquests fulls anomenats en cascada, es calculen prioritats

o pesos i s'assignen a regles, de manera que els resultats siguin previsibles.

El document HTML en si mateix és més clar d'entendre i s'aconsegueix reduir considerablement la seva grandària (sempre que no s'utilitzi estil en línia).

AJAX.

Són les sigles de Asynchronous Javascript And Xml, (JavaScript asíncron i XML), un conjunt de tecnologies que permeten actualitzar continguts web sense haver de tornar a carregar la pàgina. Això obre la porta a pàgines web interactives.

Ajax és asíncron en tant que les dades addicionals són demanades i carregades en un segon pla, sense interferir en la presentació i el comportament de la pàgina. Habitualment les funcions d'Ajax es criden des del llenguatge JavaScript. Les dades són adquirides usant l'objecte XMLHttpRequest, que està disponible als llenguatges de script que corren en navegadors moderns, o també usant Remote Scripting en navegador que no suportin XMLHttpRequest. En qualsevol cas, no és necessari que el contingut asíncron sigui formatat en XML.

Ajax és multi plataforma i es pot usar en diversos sistemes operatius, arquitectures de computador i navegadors web, ja que es basa en estàndards oberts com JavaScript i DOM. Hi ha implementacions open source de frameworks i llibreries.

Una possible tècnica per programar en AJAX és mitjançant un motor AJAX (escrit en JavaScript). Aquest motor és l'encarregat de mostrar la interfície gràfica i comunicar-se amb el servidor. Mitjançant AJAX, l'usuari no es connecta directament amb el servidor, sinó que primer ho fa amb el motor d'AJAX i aquest és el que fa la sol·licitud al servidor. Un cop això, el servidor retorna la sol·licitud que passa a través del motor i renderitza el resultat, de manera que mai es veurà la pàgina

completament blanca com passa amb els models clàssics d'aplicacions web.

XAMPP.

XAMPP és un paquet de programari lliure que conté el servidor HTTP Apache, base de dades de MariaDB i eines necessàries per utilitzar el PHP i el llenguatge de programació Perl. El programa es llença sota el GNU General Public License i d'un servidor web, d'ús fàcil, capaç de servir pàgines dinàmiques. Actualment, XAMPP està disponible per a Windows, Linux, Solaris i Mac OS X. El nom XAMPP prové de l'acrònim de X (fent referència al fet que és compatible amb diversos sistemes operatius), Apache, MySQL, PHP i Perl.

Javascript.

JavaScript és un llenguatge script basat en el concepte de prototipus (herència per delegació), implementat originàriament per Netscape Communications Corporation, i que va a derivar en l'estàndard ECMAScript. És conegut sobretot pel seu ús en pàgines web però s'utilitza en altres aplicacions.

HTTP.

EL protocol de transferència d'hipertext o HTTP (HyperText Transfer Protocol) estableix el protocol per a l'intercanvi de documents d'hipertext i multimèdia al web. HTTP disposa d'una variant xifrada mitjançant SSL anomenada HTTPS

6. Bibliografia

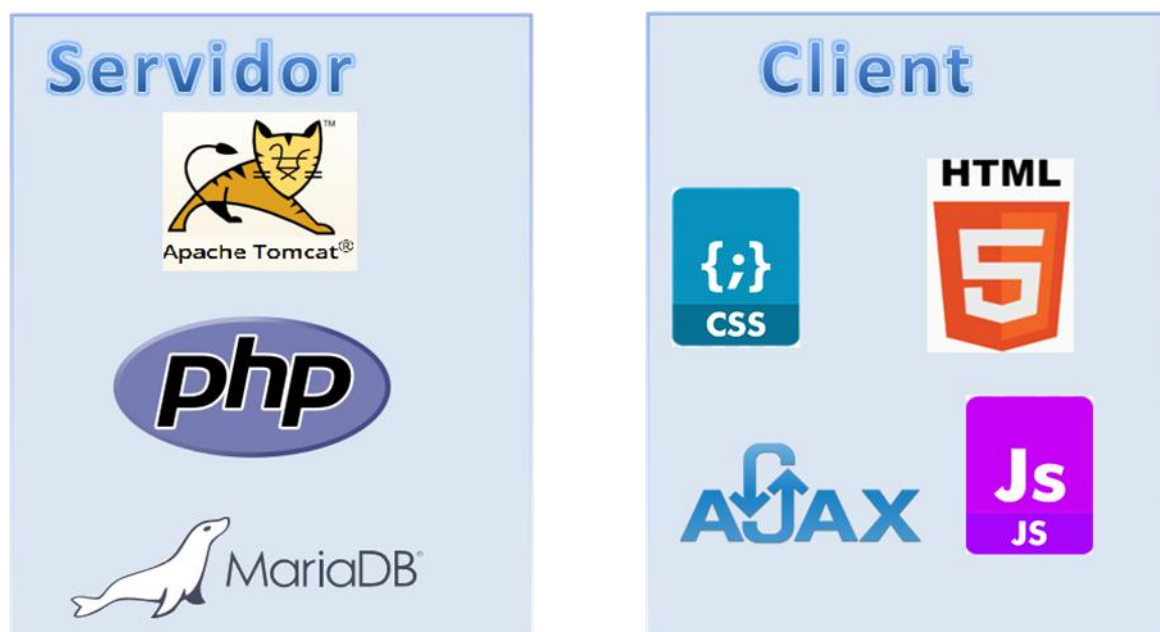
En aquest apartat es mostra el llistat de fons d'informació y recursos webs consultades des del inici del projecte , 20/09/2017, fins la finalització, 10/01/2018.

- [1] Pàgina oficial de PHP, <http://www.php.net> ,data d'accés 20-10-2017.
- [2] Pàgina oficial de MySQL, <https://www.mysql.com>, data d'accés 20-10-2017.
- [3] Pàgina oficial de MariaDB, <https://mariadb.com>, , data d'accés 20-10-2017.
- [4] Stack Overflo, pàgina de suport web, <http://www.stackoverflow.com>, data d'accés 05-11-2017.
- [5] W3Schools, ajuda dubtes HTML/HTML5/AJAX, <http://www.w3schools.com> , data d'accés 15-11-2017.
- [6] CSS-TRICKS, ajuda disseny web. <http://css-tricks.com> , data d'accés 22-11-2017.
- [7] Consorci d'Educació de Barcelona Ensenyament, <http://www.edubcn.cat/ca>, data d'accés 18-11-2017.
- [8] A.L.I.C.E Artificial Intelligence Foundation, <http://www.alicebot.org/aiml.html>, data d'accés 15-10-2017.
- [9] Guia AIML, http://www.w3ii.com/es/aiml/aiml_quick_guide.html, data d'accés 16-10-2017.
- [10] Foros del web. Comunitat per aprenentatge web.<http://www.forsdelweb.com/>, data d'accés 10-11-2017.
- [11] Wikipedia. Enciclopèdia.[en línea] : <http://www.wikipedia.org>, data d'accés 17-11-2017.
- [12] Edició en català de la Wikipedia, <https://ca.wikipedia.org>, data d'accés 17-11-2017.
- [13] Comunitat desenvolupadors de bots, <https://www.chatbots.org>, data d'accés 27-10-2017.

7. Annexos

7.1 Manual d'instal·lació.

En el següent diagrama es mostren les diferents tecnologies utilitzades tant pel servidor com pel client:



Il·lustració 10 – Tecnologies utilitzades.

7.1.1 Instal·lació i configuració de components.

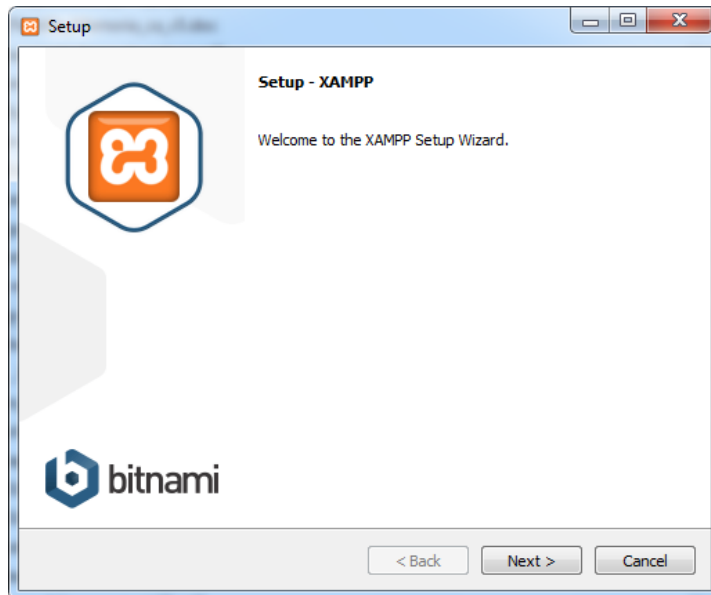
XAMMP.

Com ja he comentat abans, XAMPP és un paquet que integra un servidor web (Tomcat), una base de dades (MariaDB) y PHP. de manera que amb la seva instal·lació aconseguim tenir tots els components que fem servir en aquest projecte.

El primer que haurem de fer ara és accedir al web <https://www.apachefriends.org/index.html> i un cop aquí anem a la pestanya “download descargar la darrera versió”.

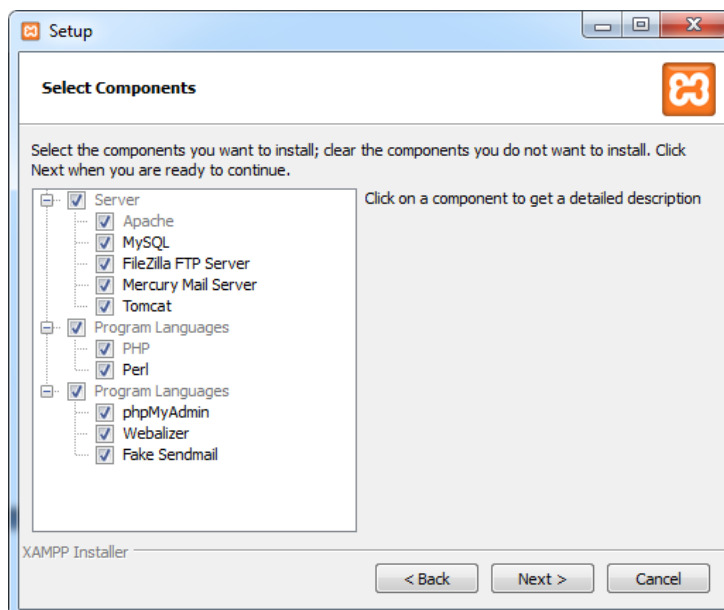
Executem el fitxer excel que ens descarreguem i només hem de seguir les instruccions que ens van apareixent.

La primera pantalla que ens trobem ens dóna la benvinguda, ara premen en next.



Il·lustració 11 – Instal·lació : setup XAPP.

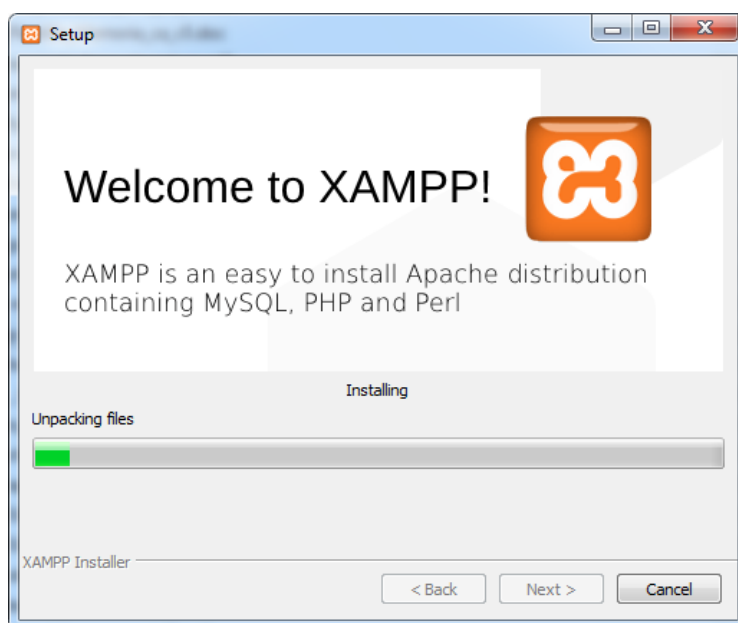
A continuació ens mostra tots els components que s'instal·laran i premem en next.



Il·lustració 12 – Instal·lació : components XAPP.

Seleccionem la carpeta on volem instal·lar-lo i prenem una altra vegada next.

Continuem els passos fins que ens apareix la següent pantalla on s'indica que ha començat la instal·lació i esperem que finalitzi.

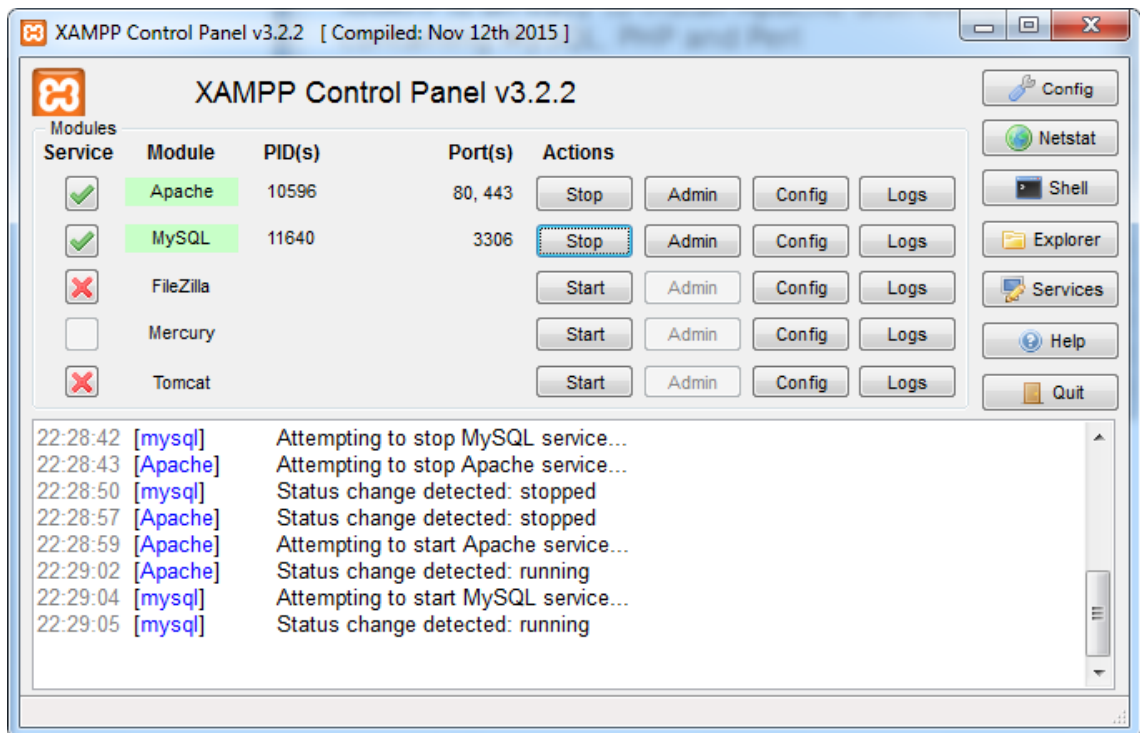


Il·lustració 13 – Instal·lació : inici instal·lació XAPP.

Al finalitzar ja tenim tots els components instal·lats.

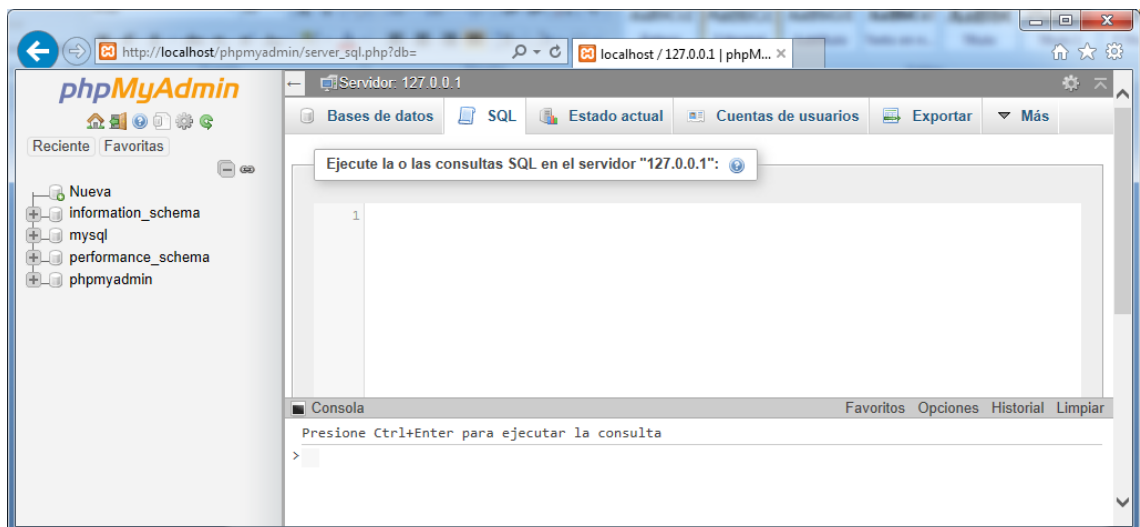
Creació de la base de dades.

Obrim el panel de control de Xampp i arrenquem el servidor web (apache) i la base de dades que es mostra com a MySQL. Ho fem simplement prenem els botons “Start”.



Il·lustració 14 – Instal·lació : panel control XAPP.

Quan ja han arrencat ambdós servidors, prenem el botó “Admin” de MySQL i se’ns obrirà un navegador amb el panel d’administració de la base de dades com el que es veu a continuació.



Il·lustració 15 – Instal·lació : panel administració base de dades.

I des de la consola o des de la finestra SQL executem aquestes sentències per a crear la base de dades i un usuari:

```
SET NAMES utf8;
create database if not exists 'uocbot';
CREATE USER 'uocbot'@'localhost' IDENTIFIED BY 'uocbot';
```

És en aquest moment quan podem crear totes les taules que necessita la nostra aplicació

```
CREATE TABLE 'aiml' (
  'id' int(11) NOT NULL AUTO_INCREMENT,
  'aiml' text NOT NULL,
  'pattern' varchar(255) NOT NULL,
  'thatpattern' varchar(255) NOT NULL,
  'template' text NOT NULL,
  'topic' varchar(255) NOT NULL,
  'filename' varchar(255) NOT NULL,
  PRIMARY KEY ('id'),
  KEY 'topic' ('topic'),
  KEY 'pattern' ('pattern'),
  KEY 'thatpattern' ('thatpattern'),
  KEY 'patternthatpattern' ('pattern','thatpattern')
);

CREATE TABLE 'botpersonality' (
  'id' int(11) NOT NULL AUTO_INCREMENT,
  'bot' tinyint(4) NOT NULL DEFAULT '0',
  'name' varchar(255) NOT NULL DEFAULT "",
  'value' text NOT NULL,
  PRIMARY KEY ('id'),
  KEY 'botname' ('bot','name')
);

CREATE TABLE 'chat' (
  'chat_id' int(5) NOT NULL AUTO_INCREMENT,
```

```

'userid' int(5) DEFAULT NULL,

'client_message' varchar(255) DEFAULT NULL,

'server_response' varchar(500) DEFAULT NULL,

'datetime_response' datetime DEFAULT NULL,

'aiml_id' int(11) default null,

PRIMARY KEY ('chat_id')

);

CREATE TABLE 'sessions' (

'id' int(5) NOT NULL AUTO_INCREMENT,

'session_id' varchar(100) DEFAULT NULL,

PRIMARY KEY ('id'),

UNIQUE KEY 'session_id' ('session_id');

CREATE TABLE 'unknown_inputs' (

'id' int(11) NOT NULL AUTO_INCREMENT,

'input' text NOT NULL,

'userid' int(11) NOT NULL,

'timestamp' timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,

PRIMARY KEY ('id')

)

grant all on uocbot.aiml to uocbot@localhost;

grant all on uocbot.botpersonality to uocbot@localhost;

grant all on uocbot.chat to uocbot@localhost;

grant all on uocbot.sessions to uocbot@localhost;

grant all on uocbot.unknown_inputs to uocbot@localhost;

```

Càrrega de dades AIML.

Per a carregar els patrons de pregunta-resposta d'AIML a la base de dades, repetim el mateix procés que en el punt anterior. Agafem els inserts i l'executem des de la pantalla importar o sql del panel d'administració de la base de dades.

Amb això ja tenim carregades la lògica del bot perquè pugui respondre als usuaris.

Instal·lació components web.

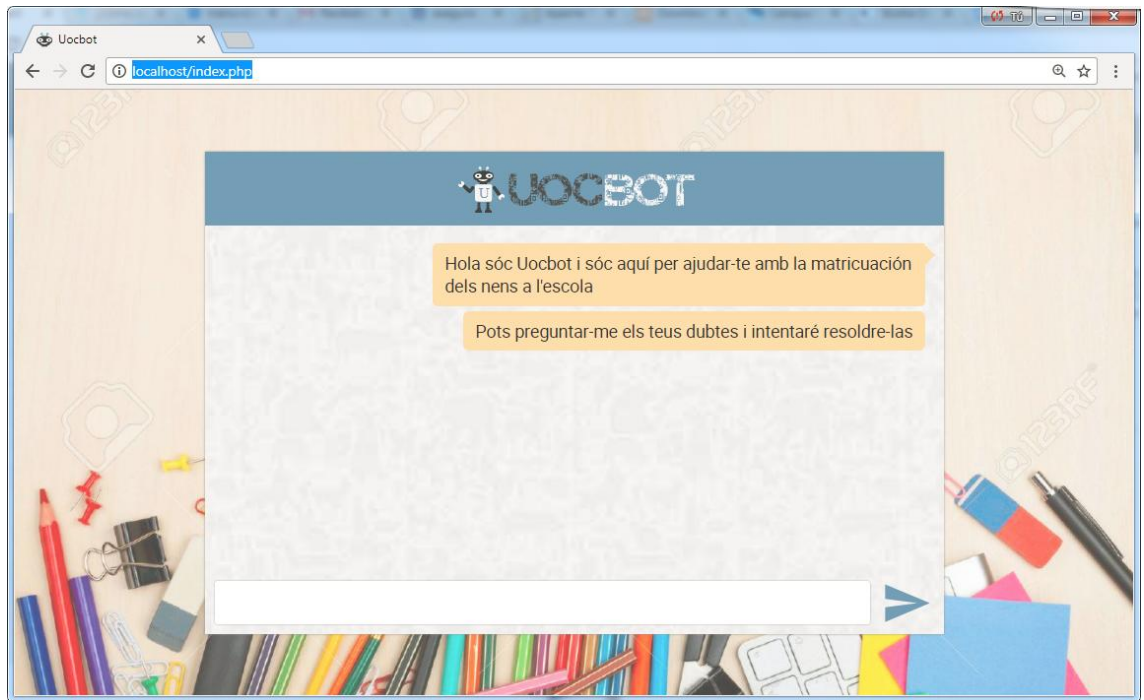
Per a instal·lar tots els components php i javascript que el bot utilitza haurem de posar els fitxers directament a la carpeta htdocs on hem instal·lat Xampp (per defecte està en l'adreça C:\xampp\htdocs

Provar instal·lació.

Una vegada que ja hem fet tots els passos anteriors i tots els components estan instal·lats i configurats, podrem accedir al bot.

I amb aquesta finalitat obrirem un navegador on posarem la següent url <http://localhost/index.php>

Ens apareixerà una pantalla com la que es mostra a continuació i ja podrem entrar les nostres demandes al bot.



Il·lustració 16 – Instal·lació : prova bot.

7.2 Manual d'usuari.

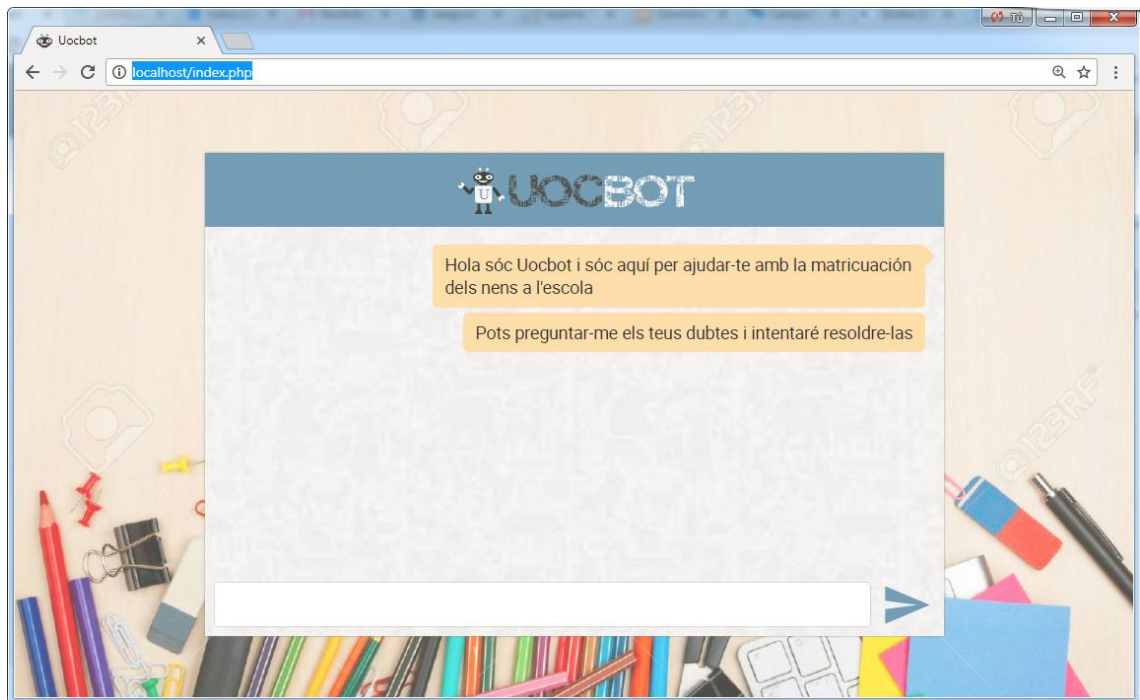
A continuació s'explicarà com interactuar amb el bot desenvolupat i també què és el mòdul d'administració.

7.2.1 Mòdul del bot.

Una vegada que ja hem fet tots els passos indicats al manual d'instal·lació i que tenim tots els components instal·lats i configurats, podrem accedir al bot.

Per això obrirem un navegador on posarem la següent url <http://localhost/index.php>

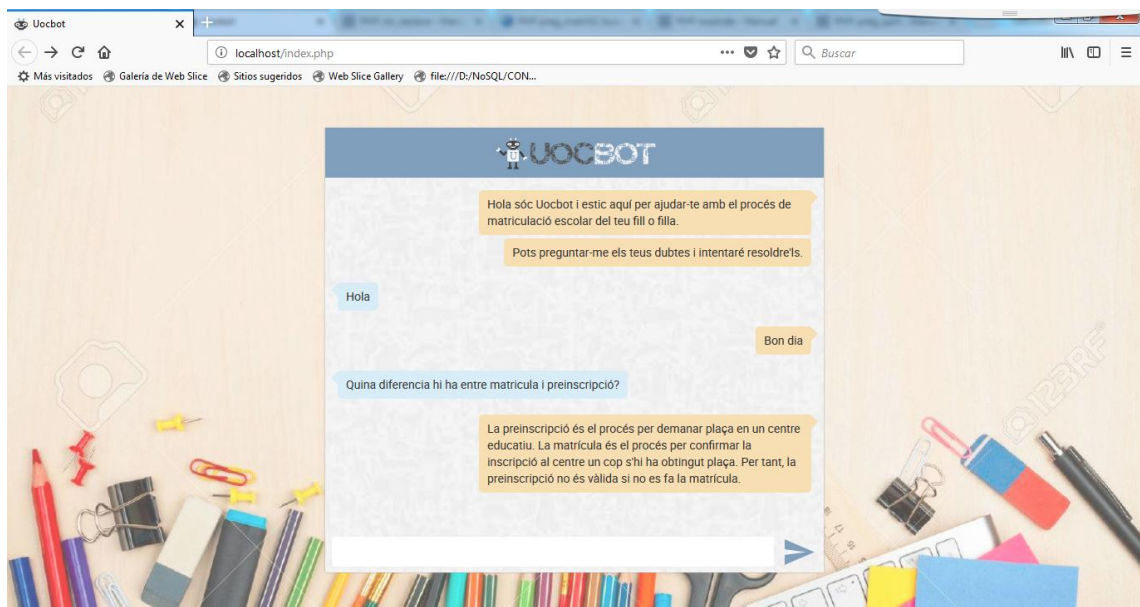
Ens apareixerà una pantalla com la que es mostra a continuació i on ja podrem escriure les nostres demandes al bot.



Il·lustració 17 – Mòdul del bot.

A cada pregunta introduïda, el bot ens respondrà amb la resposta més adequada que hagi trobat. Si no en troba cap resposta adient, ens ho indicarà i guardarà la pregunta en una graella especial que després podrem analitzar amb el mòdul d'administració.

Deixo un parell d'exemples de com s'interactua amb el bot, ho podem veure en la següent pantalla.

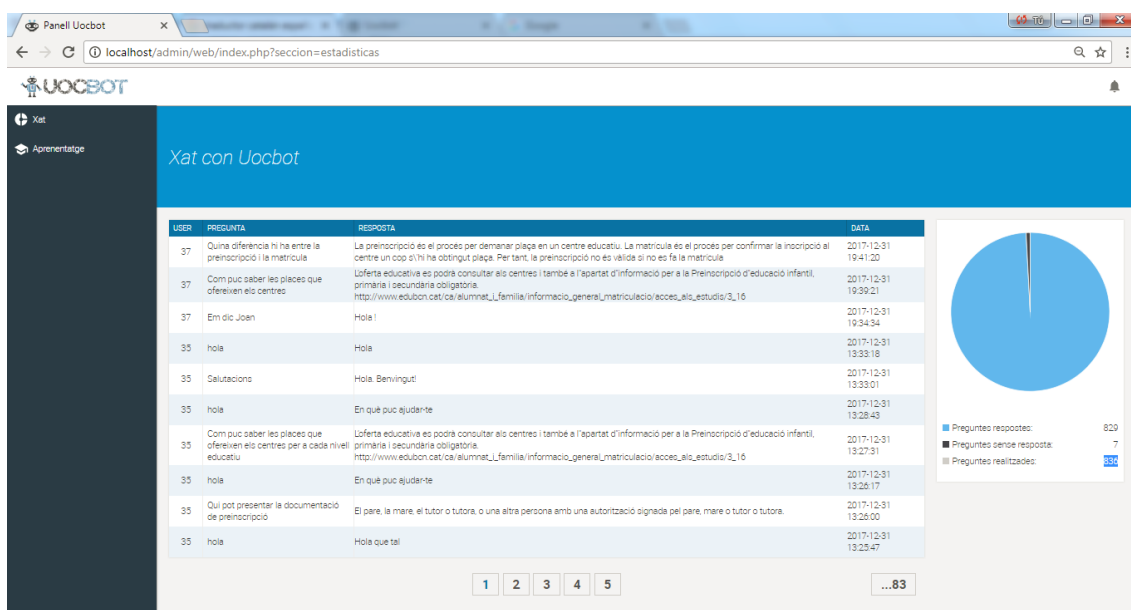


Il·lustració 18 – Exemple xat.

7.2.1 Mòdul d'administració.

Consulta d'estadístiques.

Accedint a la url <http://localhost/admin/web> l'usuari accedirà al dashboard d'estadístiques d'ús del bot. També es podran veure les sessions del bot, les preguntes que li han fet, les respostes que ha pogut donar i les preguntes que no ha pogut respondre satisfactòriament.

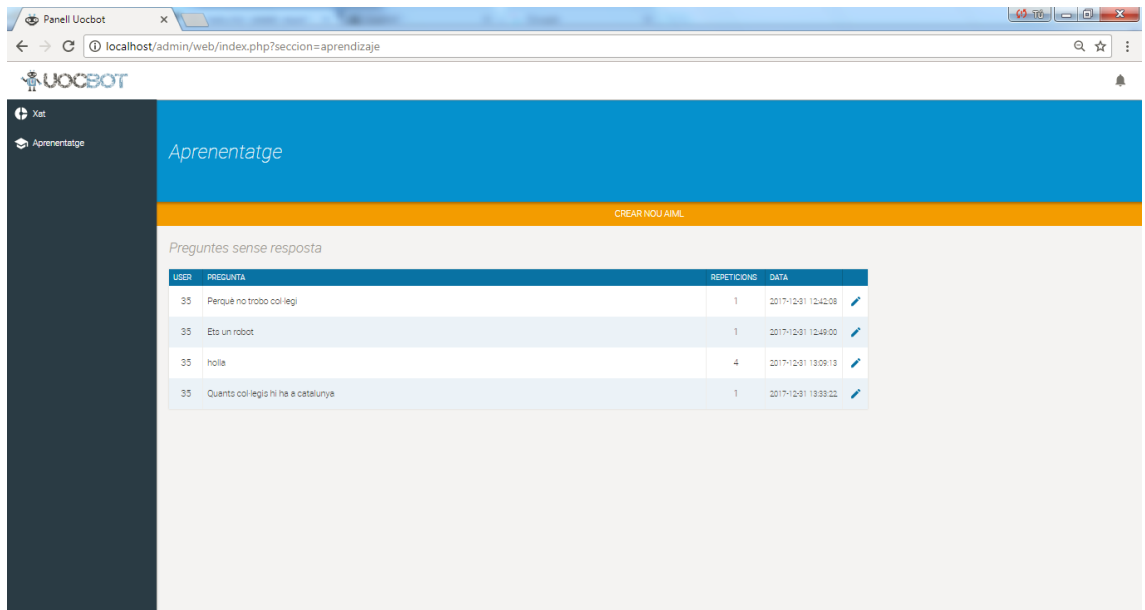


Il·lustració 19 – Consulta d'estadístiques.

Anàlisi de preguntes no resoltes.

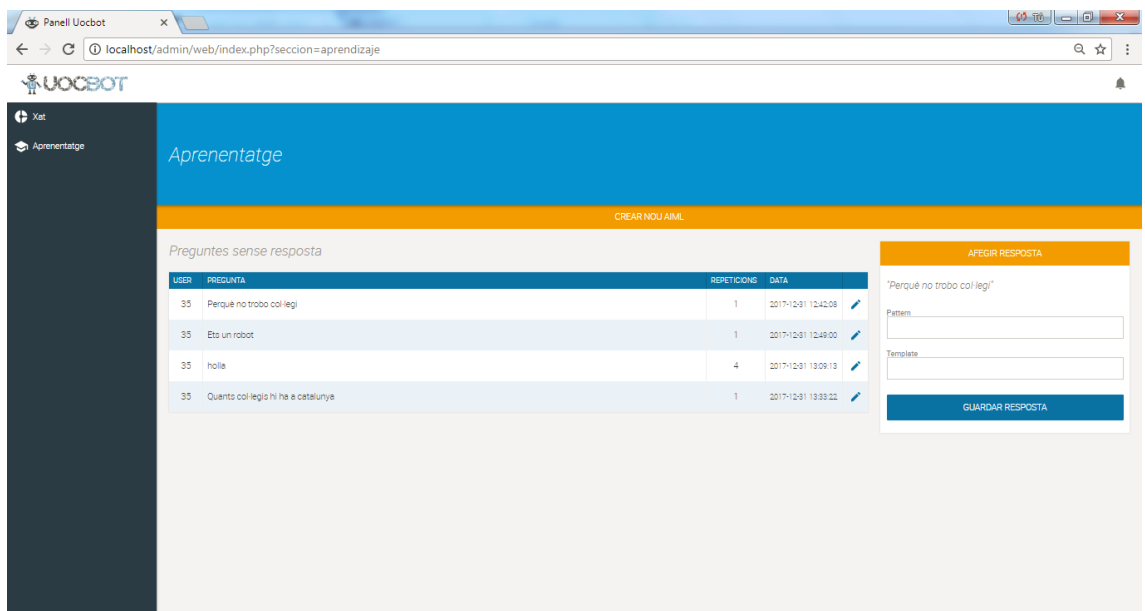
Per a poder consultar les preguntes no resoltes l'usuari haurà de prémer el botó "Aprentatatge" de la pantalla d'administració que hem vist anteriorment.

Aleshores apareixerà la següent pantalla, des de la que podrem veure quines preguntes no ha pogut respondre el bot perquè no ha trobat cap patró AIML similar a la base de dades.



Il·lustració 20 – Anàlisi de preguntes no resoltes.

A la pregunta que volem afegir haurem de pulsar al llapis i s'obrirà una pantalla al costat des de la qual podrem inserir a la base de dades la pregunta perquè el bot sigui capaç de respondre-la. Per aconseguir-lo, l'administrador haurà de tenir coneixements d'AIML ja que haurà de definir els tags pattern i template per a l'interpret d'AIML. Des de l'opció "CREAR NOU AIML" es podrà crear i carregar a la base de dades un patró AIML des de zero.



Il·lustració 21 – Afegir resposta.