



## PROJECTE: FICAS

ESTUDI DE VIABILITAT D'UN PROTOTIP D'INTEGRACIÓ DE SISTEMES D'IA

**Francisco Cortés Sánchez**

Grau Enginyeria Informàtica  
Intel·ligència Artificial

**Nom Consultor/a:** David Isern Alarcón

**Nom Professor/a responsable de l'assignatura:** Carles Ventura Royo

Data Lliurament: 8 gener de 2018



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/)

<b>Títol del treball:</b>	<i>Projecte FICAS. Estudi de viabilitat d'un prototip d'integració de sistemes d'IA.</i>
<b>Nom de l'autor:</b>	<i>Francisco Cortés Sánchez</i>
<b>Nom del consultor/a:</b>	<i>David Isern Alarcón</i>
<b>Nom del PRA:</b>	<i>Carles Ventura Royo</i>
<b>Data de lliurament (mm/aaaa):</b>	<i>octubre de 17</i>
<b>Titulació o programa:</b>	<i>Grau Enginyeria Informàtica</i>
<b>Àrea del Treball Final:</b>	<i>Intel·ligència Artificial</i>
<b>Idioma del treball:</b>	<i>Català</i>
<b>Paraules clau</b>	<ul style="list-style-type: none"> <li>- <i>Reconeixement automàtic de la parla (RAH)</i></li> <li>- <i>Síntesi de veu</i></li> <li>- <i>Reconeixement de veu</i></li> </ul>

#### **Resum del Treball:**

Projecte FICAS. Estudi de viabilitat d'un prototip de integració de sistemes d'Intel·ligència artificial amb capacitat per a aprendre.

Aquest projecte es basa amb la idea que, avui dia tots els sistemes d'intel·ligència artificial fan tot, i sol, allò per al que estan programats. El propòsit és aconseguir que es realitzin reaccions noves, davant de noves accions. El principal problema a resoldre és la integració dels diferents sistemes de: reconeixement facial, de la parla i patrons; perquè el projecte FICAS simuli que aprengui com ho fem els humans, mitjançant la vista, l'oïda i la parla. Les persones aprenem associant una imatge, o successió d'elles, a un nom en format de text i amb un so en freqüència de veu.

Després d'un minuciós estudi dels sistemes anteriors, fiques s'implementarà en Python, per la seva senzillesa i versatilitat (a l'hora de menjar dades), que permeten que el projecte sigui viable, amb una

gran varietat de possibilitats d'integració. I el més important que, pot ser un projecte que tingui una gran comercialització per que millorarà diversos aspectes de la societat.

**Abstract:**

Project FICAS Feasibility study of a prototype integration of artificial intelligence systems with a capacity to learn.

This project is based on the idea that, today, all artificial intelligence systems do everything, and only, what they are programmed for. The purpose is to achieve new reactions, before new actions. The main problem to solve is the integration of the different systems of: face recognition, speech and patterns; because the project simulates learning how we humans do, through sight, hearing and speech. People learn by associating an image, or sequence of them, with a name in text format and with a sound in voice frequency.

After a thorough study of previous systems, goals will be implemented in Python, for its simplicity and versatility (when it comes to data storage), which allow the project to be viable, with a wide variety of integration possibilities. And the most important thing that can be a project that has great commercialization so that it improves diverse aspects of the society.

# Índex

---

Índex .....	iii
Índex d'il·lustracions .....	vi
Índex de taules .....	vii
Agraïments.....	viii
1. Introducció.....	1
1.1. Descripció del PFG.....	1
1.2. Objectius generals.....	2
1.3. Objectius específics .....	2
1.4. Planificació amb fites i temporalització.....	4
1.4.1. Diagrama de Gantt .....	5
1.4.2. Fites durant el desenvolupament del projecte.....	6
1.4.3. Estudi de costos .....	7
1.4.4. Contingut de les PACs i del lliurament final.....	7
2. Definicions de conceptes .....	10
2.1. Què es la Intel·ligència artificial? .....	10
2.2. Com aprenen els humans?.....	11
2.3. Per què Python?.....	13
3. Breu descripció de sistemes de IA.....	18
3.1 Sistema de captura d'imatges .....	20
3.1.1 Definició.....	20
3.1.2 Funcionament.....	24
3.1.3 Cas d'ús i diagrama de seqüència .....	25
3.2 Sistema de reconeixement de veu .....	27
3.2.1 Definició.....	27
3.2.2 Funcionament.....	28
3.2.3 Cas d'ús i diagrama de seqüència .....	29
3.3 Sistema de síntesis de so .....	31
3.3.1 Definició.....	31
3.3.2 Funcionament.....	31
3.3.3 Cas d'ús i diagrama de seqüència .....	33
4. Integració de sistemes .....	35
4.1.Descripció de funcionament .....	38
4.1.1.Diagrama de classes .....	41
4.1.2.Prototip d'Interface gràfica .....	41
4.1.3.Descripció d'integració .....	42
4.2.Casos d'ús .....	42
4.3.Diagrames de seqüència .....	48
5.Implementació.....	51

5.1.Funcionalitat de FICAS.....	53
5.2. Joc de proves.....	59
6.Conclusions .....	63
6.1. Objectius complits .....	63
6.2. Viabilitat .....	64
6.3. Futures millores .....	65
7. Glossari .....	66
8. Bibliografia .....	68
9. Annexos .....	70

# Índex d'il·lustracions

---

Il·lustració 1: Esquema del cicle de vida clàssic de desenvolupament de projectes informàtics. ....	4
Il·lustració 2: Esquema de la escala de temps de desenvolupament del projecte. ....	5
Il·lustració 3: Diagrama de Gantt amb la planificació detallada de la PAC0 i PAC1. ....	6
Il·lustració 4. Del lòbul frontal (3), on es centralitza la memòria. ....	13
Il·lustració 5. Arquitectura i funcionament dels sistemes experts. ....	18
Il·lustració 6. Integració de sistemes ....	19
Il·lustració 7. Comparativa de fotos en format RGB, amb un petit desplaçament en el enquadri. ....	21
Il·lustració 8. Imatge convertida a escala de grisos. ....	22
Il·lustració 9. Imatge convertida a blanc i negre. ....	23
Il·lustració 10. Imatge de simulació de visió artificial. ....	24
Il·lustració 11. Diagrama de blocs que mostra el procés de captura d'una imatge. ....	25
Il·lustració 12. Diagrama de seqüència del subsistema SCI ....	27
Il·lustració 13. Diagrama de blocs que mostra el procés del subsistema de reconeixement de veu. ....	29
Il·lustració 14. Diagrama de seqüència del subsistema SRV ....	30
Il·lustració 15. Diagrama de blocs que mostra el procés del subsistema de síntesi de so. ....	32
Il·lustració 16. Diagrama de seqüència del subsistema SSS. ....	34
Il·lustració 17. Exemple de classificació del catàleg ("animals") ....	36
Il·lustració 18. Representació UML del model-vista-controlador. ....	39
Il·lustració 19. Representació UML de la programació per capes i les respectives funcions. ....	40
Il·lustració 20. Diagrama de classes del projecte FICAS. ....	41
Il·lustració 21. Pantalla de presentació del projecte FICAS. ....	41
Il·lustració 22. Esquema d'integració de sistemes del projecte FICAS. ....	42
Il·lustració 23. Diagrama de casos d'ús de les funcionalitats del projecte. ....	43
Il·lustració 24. Diagrama de seqüència per crear un nou catàleg. ....	48
Il·lustració 25. Diagrama de seqüència per crear un nou diccionari. ....	49
Il·lustració 26. Diagrama de seqüència del cas d'ús conversar (admin i usuari registrat) ....	49
Il·lustració 27. Diagrama de seqüència del cas d'ús usuàries ....	50

# Índex de taules

---

Taula 1: Diagrama de Gantt de tot el projecte de TFG.....	5
Taula 2: Taula de planificació de les diferents fases del desenvolupament del projecte 'FICAS'.....	6
Taula 3: Taula de costos de les diferents fases del desenvolupament del projecte 'FICAS'.....	7
Taula 4. Codi en Python que mostra que fa servir tipat estàtic.....	15
Taula 5. Comparativa de les principals característiques diferencials entre Java i Python.....	16
Taula 6. Comparativa des de el punt de vista dels costos.....	16
Taula 7. Exemple de paràmetres arbitraris en Python.....	17
Taula 8. Exemple de crides amb recursivitat en Python.....	17
Taula 9. Codi amb Java per comparar dues fotos píxel a píxel.....	21
Taula 10. Codi Java per fer la conversió d'una imatge en RGB a escala de grisos.....	22
Taula 11. Codi Java per fer la conversió d'una imatge en RGB a blanc i negre.....	23
Taula 12. Descripció del cas d'ús del subsistema de captura d'imatges (SCI).....	26
Taula 13. Descripció del cas d'ús del subsistema de reconeixement de veu (SRV).....	30
Taula 14. Descripció del cas d'ús del subsistema de síntesis de so (SSS).....	34
Taula 15. Exemple de descripció dels camps arbitraris.....	37
Taula 16. Exemple de definició de paràmetres arbitraris en Python.....	37
Taula 17. Descripció del cas d'ús de crear un nou catàleg.....	45
Taula 18. Descripció del cas d'ús de crear un nou diccionari.....	46
Taula 19. Descripció del cas d'ús de conversar.....	47
Taula 20. Descripció del cas d'ús de registre d'usuaris.....	48
Taula 21. Verificació inicial de funcionament de sistemes.....	54
Taula 22. Menú de verificació d'usuaris per registrar-se en FICAS.....	55
Taula 23. Menú principal de FICAS per accedir al diferents subsistemes gestors.....	56
Taula 24. Les ordes per interactuar amb els diferents subsistemes.....	57
Taula 25. Codi per escriure als fitxers amb codificació UTF-8.....	58
Taula 26. Joc de proves per verificar la funcionalitat del projecte.....	60
Taula 27. Resultat del joc de proves del projecte.....	62
Taula 28. Diferencia del directori després d'aplicar el joc de proves.....	62
Taula 29. Sistema de funcionament de una Macro que te varies ordres.....	65



## Agraïments

---

Abans de donar les gràcies, em permetreu que se sigui una mica objectiu perquè necessito explicar-vos una part de vida. Quan era jove m'agradava molt estudiar i vaig obtenir bons resultats, però per motius de diners no vaig poder estudiar, a molt pesar meu, per falta de diners; bo, coses de la vida! ...

La meva vida va continuar, vaig formar una família, ens venim a viure a Catalunya, i gràcies a la UOC estic complint un somni, que és acabar els meus estudis (; tot i que, el temps transcorregut ha estat força llarg, ja estic veient llum al final del camí, les hores de falta de son que han estat excessives, no em provocat desanimo, perquè he tingut un gran suport de la meva família, de la qual estic molt orgullós, encara que molt poques vegades tens el valor de dir-ho, o simplement la rutina i la manca de temps no et permeten, ni tan sols pensar-, però, no puc deixar de recordar en la il·lusió que tenia la meva mare amb els meus estudis: recordo un dia, quan la vaig visitar a la residència, em va presentar a diversos coneguts, i va dir en veu alta, ¡Aquest és el meu fill i, hi ha on ho veieu, és enginyer! Per la meva mare, fa uns 8 anys, jo ja era enginyer.

En aquesta vida cal buscar alguna cosa al que aferrar-se per aconseguir els teus objectius; en el meu cas, ha estat la meva família i la il·lusió de la meva mare, perquè al final, per molt que costi, 'en la nostra vida, som el que volem ser ".

Per acabar, em permetreu que em reiteri i torni a donar les gràcies a la UOC, per inventar aquest sistema d'estudi; a la meva mare (que al cel sia) per inculcar els valors de lluita i superació; i la meva família, per aguantar-me i no dedicar-los en temps que jo he necessitat per intentar complir el meu somni.

Reflexió: 'el saber' és l'únic que no es pot comprar amb diners, però és lamentable, que avui en dia, compti tants diners 'el saber'.

# 1. Introducció

## 1.1. Descripció del PFG

---

Projecte FICAS. Estudi de viabilitat d'un prototip d'IA amb capacitat per a aprendre.

Avui dia tots els sistemes d'intel·ligència artificial fan tot, i sol, allò per al que estan programats. El propòsit és aconseguir que es realitzin reaccions noves, davant de noves accions. El principal problema a resoldre és la integració dels diferents sistemes de: reconeixement facial, de la parla i patrons; perquè el projecte FICAS simuli que aprengui com ho fem els humans, mitjançant la vista, l'oïda i la parla. Les persones aprenem associant una imatge, o successió d'elles, a un nom en format de text i amb un so en freqüència de veu.

Per tant, un cop seleccionats i estudiats els diferents sistemes, aquest projecte es basarà en realitzar un estudi de viabilitat d'un prototip, d'un cas d'estudi, que generi pautes d'interacció entre sistemes; amb l'únic objectiu que el projecte FICAS sigui capaç de aprendre tot el que ens interessi, interactuant amb l'usuari per mitjà de la parla; amb la gran dificultat de la efectivitat, es a dir, serà més efectiu quant la taxa d'errors sigui la major possible (De vegades les persones no pronunciem bé les nostres paraules i provoquen dubtes al nostre interlocutor).

A més a més, disposarà amb una gran versatilitat de suports d'instal·lació. El suport pot variar depenent de l'ús que necessitem, però és imprescindible una entrada i sortida d'àudio, i entrada de vídeo; exemples: en qualsevol Pc, domòtica, robòtica, automòbils i, per què no, en les naus que ens portaran a mart.

En conclusió, no estaria malament ser un poc optimista i pensar que aquest projecte podria ser la base de una nova interacció persona-ordinador, també podria iniciar una nova perspectiva de com seran els nous sistemes de

intel·ligència artificial, ja que aquest projecte podrà continuar amb unes expectatives de desenvolupament sense límits.

## 1.2. Objectius generals

---

Aquest projecte es basarà en l'estudi de viabilitat per a realitzar un prototip d'IA amb capacitat per a aprendre, basant-se en un cas d'estudi, que generi pautes d'interacció entre sistemes.

El prototip es basarà en la interacció de diversos sistemes, per simular l'aprenentatge humà, quan som petits tenim sentit, vista i podem parlar; per tant, s'ha de partir d'un sistema de reconeixement d'objectes (la vista), d'un sistema de reconeixement de veu (l'oïda), un sistema de síntesi de veu (el parla), que siguin implementats i que estiguin operatius en el sistema FICAS.

Per aquest motiu és necessari, una petita definició de com funcionen per separat cadascun, amb l'objectiu d'iniciar les pautes de com han d'interactuar entre si y amb la finalitat que el projecte FICAS sigui capaç de interactuar amb l'usuari.

## 1.3. Objectius específics

---

- Definir que és la Intel·ligència artificial.
- Explicar com aprenem els humans.
- Buscar el programari més idoni per a la programació, Python?
- Definir els diferents sistemes d'IA: Captura de fotos, Reconeixement de veu i objectes, i Síntesi de veu.
  - Explicar breument com funcionen.
  - Aportar il·lustracions.
  - Definir casos d'ús particulars.
- Definir les pautes d'integració de sistemes.
  - Definir com funcionaran

- Aportar il·lustracions.
- Definir casos d'ús d'unificació.
- Donar exemples reals sobre les funcions que realitzarà FICAS.

### **Exemple de lectura**

Per simular lectura s'haurà de educar el sistema de la mateixa manera que els humans aprenem, aprenem associant fotos a un so i després aprenem a escriure-ho en format text. De manera anàloga, si al sistema li vam mostrar en un paper un símbol egipci o una síl·laba com "PA", capturarà una foto i ens demanarà un so que serà el seu nom; en tornar a ensenyar un paper de nou, si és el mateix el sistema reproduirà el seu nom, en el cas contrari, demanarà un nou so. Si tot això ho repetim diverses vegades aconseguim que el sistema aprengui a llegir el que ens interressi. En aquest procés es basarà en recursivitat i finalitzarà davant d'una ordre com "stop, sortir", o si el temps d'espera de sistema sobrepassa el límit preestablert. Això també pot ser aplicable per reconèixer objectes.

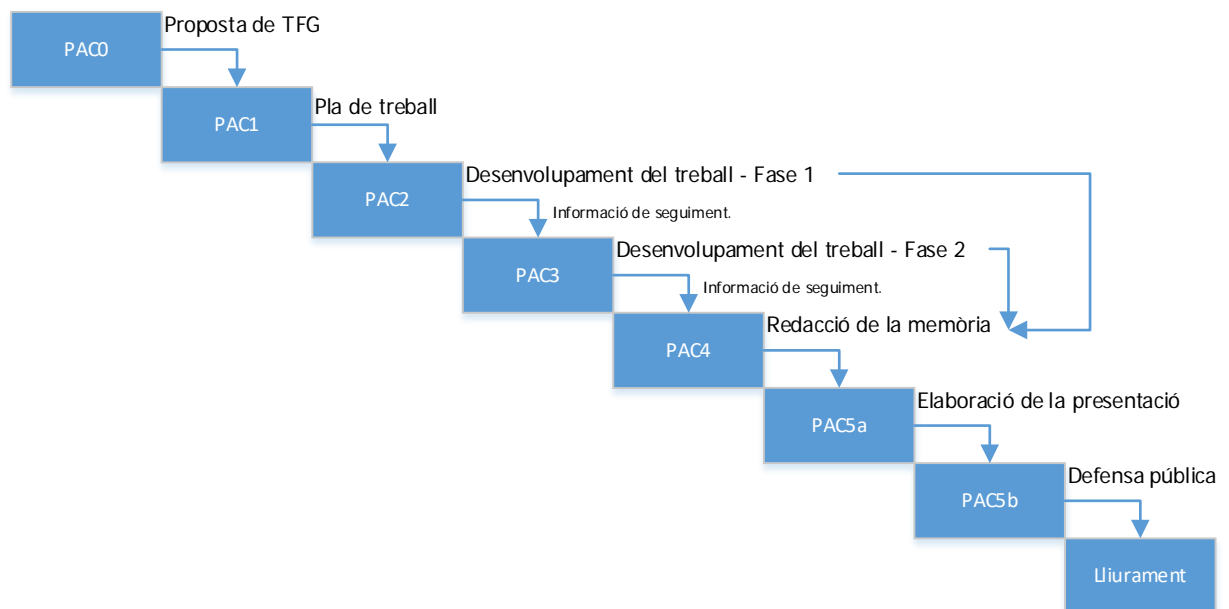
### **Exemple classes abstractes**

Si agafem l'exemple anterior i ho fem el més abstracte possible podem aconseguir que aprengui qualsevol tipus d'associacions que desitgem. Per això de manera senzilla, hem de registrar les accions (funcions, que podem anomenar macros): el sistema ens demanarà que volem fer, pronunciarem una paraula que serà el nom de la funció; si no la reconeix, ens demanarà quants camps tindrà aquesta nova macro i el nom de cada un d'ells; en el cas que la reconegui, ens demanarà les dades de cada un d'ells, reproduint el seu nom.

- Implementar un exemple de captura d'imatges.

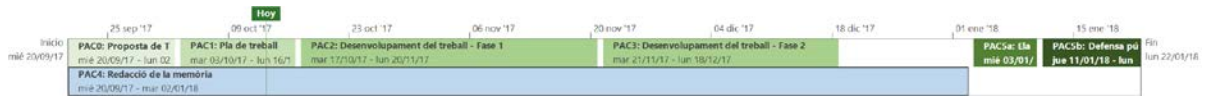
## 1.4. Planificació amb fites i temporalització

El projecte 'FICAS' es planteja amb un cicle de vida clàssic. Es seguiran, per tant, les fases d'aquest cicle que han de portar cap a la finalització del projecte. Una fase no comença fins que no finalitza l'anterior i es consuma amb el lliurament de la documentació específica d'aquella etapa: especificació d'objectius, definició de fites, desenvolupament de la memòria, etc. En el cas del present projecte, els documents propis de cada fase equivalen als documents de cada fita (PACs).



Il·lustració 1: Esquema del cicle de vida clàssic de desenvolupament de projectes informàtics.

El projecte finalitza amb el lliurament de la memòria, una presentació, un vídeo de defensa i una fitxa d'autoavaluació. La planificació general del projecte 'FICAS', desglossada en les diferents fases del seu desenvolupament, es presenta en el següent gràfic:

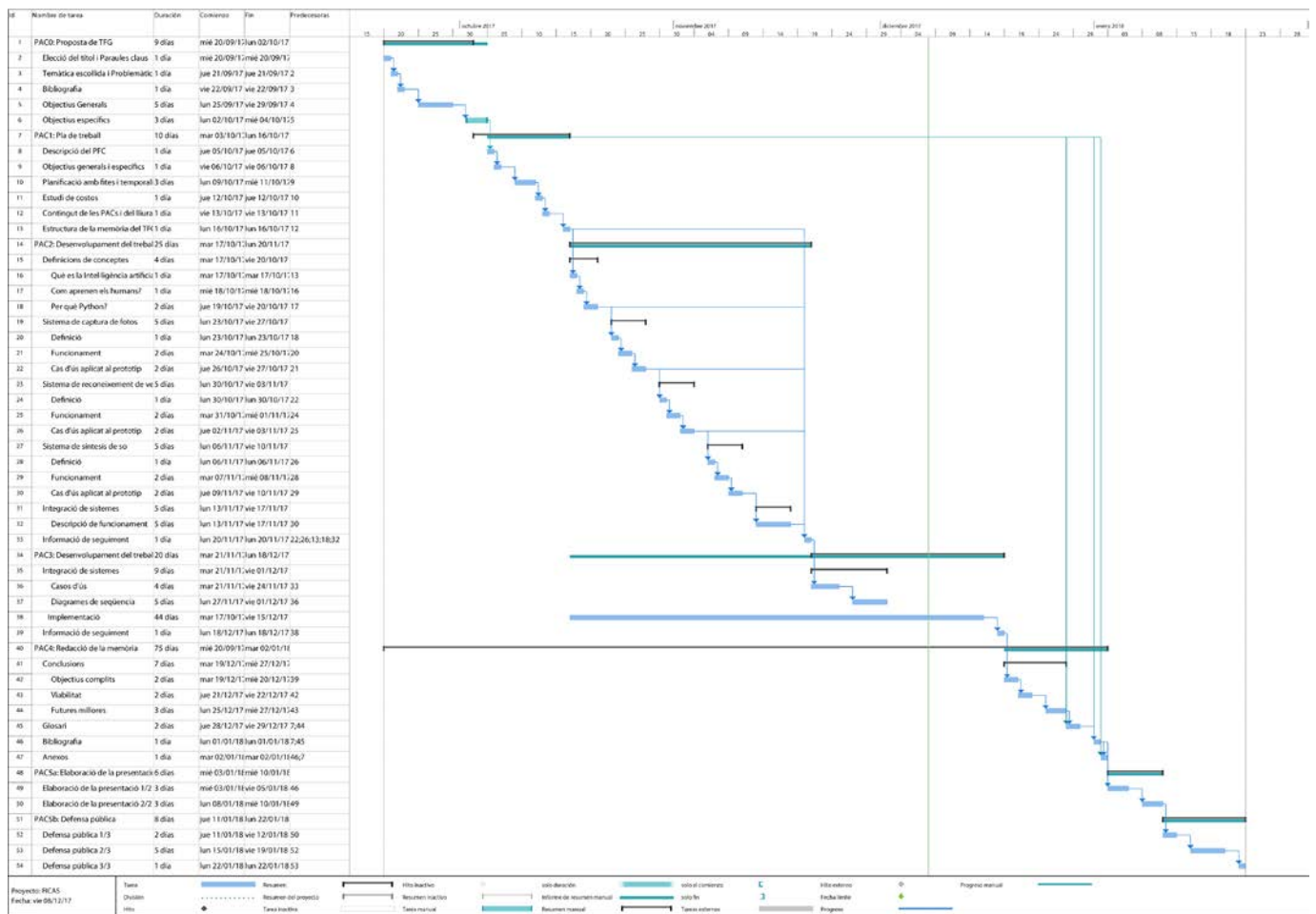


Il·lustració 2: Esquema de la escala de temps de desenvolupament del projecte<sup>1</sup>.

Cal remarcar que, tant el glossari com la bibliografia i els annexos, es poden actualitzar, al llarg del projecte, a mesura que es recopila informació.

La següent il·lustració es la planificació de tot el projecte, en forma de Diagrama de Gantt, amb l'objectiu de fer una visió global del abast de tot el procés.

### 1.4.1. Diagrama de Gantt



Taula 1: Diagrama de Gantt de tot el projecte de TFG

<sup>1</sup> Es pot apreciar que per fer la redacció de la memòria es recopila informació durant tot el procés de les part anteriors.

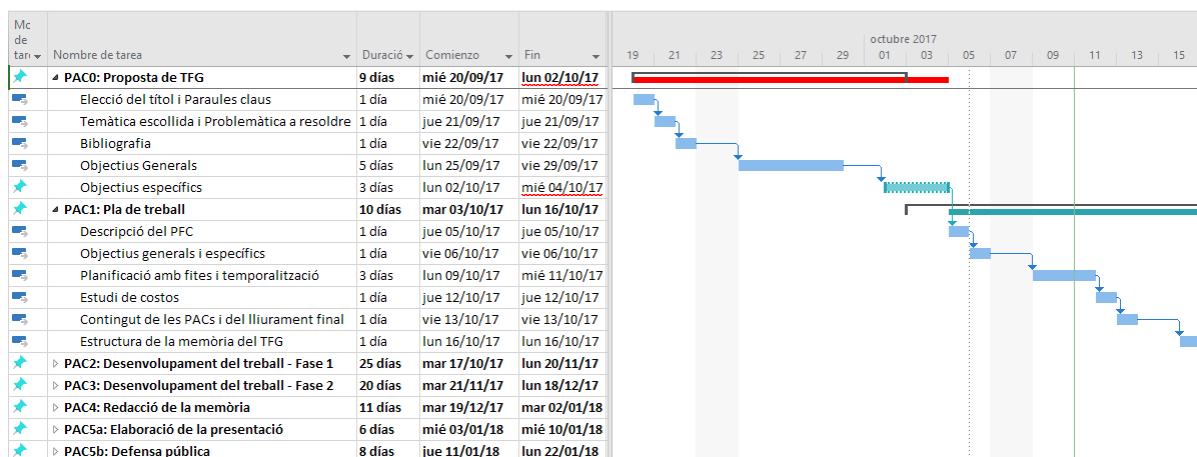
## 1.4.2. Fites durant el desenvolupament del projecte

Les fites principals del projecte corresponen als lliuraments de la documentació pròpia de cada una de les fases que marca la seva finalització. Indiquem a continuació quines són les dates previstes<sup>2</sup>.

Fita	Data
1 Inici projecte	20/09/2017
2 Lliurament document PAC0: Proposta de TFG	02/10/2017
3 Lliurament document PAC1: Pla de treball	16/10/2017
4 Lliurament document PAC2: Desenvolupament del treball - Fase 1	20/11/2017
5 Lliurament document PAC3: Desenvolupament del treball - Fase 2	18/12/2017
6 Lliurament document PAC4: Redacció de la memòria	02/01/2018
7 Lliurament document PAC5a: Elaboració de la presentació	10/01/2018
8 Lliurament document PAC5b: Defensa pública	22/01/2018
9 Fi projecte	22/01/2018

Taula 2: Taula de planificació de les diferents fases del desenvolupament del projecte 'FICAS'.

La següent il·lustració es la planificació d'aquest lliurament.



Il·lustració 3: Diagrama de Gantt amb la planificació detallada de la PAC0 i PAC1.

Es pot apreciar un petit desfas<sup>3</sup> en la Planificació de la PAC0, degut al retard en la definició dels objectius previstos del projecte.

<sup>2</sup> Corresponen a les dates indicades a la planificació de l'assignatura (TFG) per al present semestre.

<sup>3</sup> La línia vermella de la PAC0 es superposa, en el temps, a la línia verda de la PAC1.

### 1.4.3. Estudi de costos

Per fer el estudi de costos es basarem amb el dies hàbils de la durada del TFG.

	Inici	lliurament	Total dies	Dies hàbils	Costos Previstos	Dies reials	Costos reials
<b>PAC0</b>	20/09/2017	02/10/2017	12	8	13,71	12	20,57
<b>PAC1</b>	03/10/2017	16/10/2017	13	9	15,43	9	15,43
<b>PAC2</b>	17/10/2017	20/11/2017	34	25	42,86	25	42,86
<b>PAC3</b>	21/11/2017	18/12/2017	27	18	30,86	18	30,86
<b>PAC4</b>	19/12/2017	02/01/2018	14	8	13,71	8	13,71
<b>PAC5a</b>	03/01/2018	10/01/2018	7	5	8,57	5	8,57
<b>PAC5b</b>	11/01/2018	22/01/2018	11	7	12,00	7	12,00
<b>Totals</b>			<b>118</b>	<b>80</b>	<b>137,14</b>	<b>84,00</b>	<b>144,00</b>

Taula 3: Taula de costos de les diferents fases del desenvolupament del projecte 'FICAS'.

- Els dies hàbils son el total de dies disponibles, menys els cap de setmana i festius.
- Per calcular els costos tindrem en compte els crèdits de la assignatura i farem el següent: per estudiar el TFG fan falta 12 hores a la setmana, per tant, dividirem 12 entre 7 i es multiplicarem per els dies hàbils.
- Els dies en vermell son dies en desfasi sobre els dies previstos.
- Si el preu del cost es de 25€, tenim un preu final de 3.600€.

### 1.4.4. Contingut de les PACs i del lliurament final

#### **Pac0** (20/09/2017 - 02/10/2017)

Proposta de TFG

1. Elecció del títol.
2. Paraules claus.
3. Temàtica escollida.



4. Problemàtica a resoldre.
5. Objectius Generals.
6. Objectius específics.
7. Bibliografia.

**Pac1** (03/10/2017 - 16/10/2017)

## Pla de treball

1. Descripció del PFC
2. Objectius generals i específics
3. Planificació amb fites i temporalització
4. Contingut de les PACs i del lliurament final
5. Estructura de la memòria del TFG

**Pac2** (17/10/2017 - 20/11/2017)

1. Desenvolupament del treball - Fase 1
2. Informació de seguiment.
  - 2.1. Descripció de l'avanç del projecte.
    - 2.1.1. Grau de compliment dels objectius i resultats previstos en el pla de treball.
    - 2.1.2. Justificació dels canvis en cas necessari.
  - 2.2. Relació de les activitats realitzades.
    - 2.2.1 Activitats previstes en el pla de treball.
    - 2.2.2 Activitats no previstes i realitzades o programes.
  - 2.3. Relació de les desviacions en la temporització i accions de mitigació si escau i actualització del cronograma si escau.
  - 2.4. Llistat dels resultats parcials obtinguts fins al moment .
  - 2.5. Comentaris del vostre director particular si ho considereu necessari.

**Pac3** (21/11/2017 - 18/12/2017)

1. Desenvolupament del treball - Fase 2
2. Informació de seguiment.
  - 2.1. Descripció de l'avanç del projecte.

- 2.1.1. Grau de compliment dels objectius i resultats previstos en el pla de treball.
- 2.1.2. Justificació dels canvis en cas necessari.
- 2.2. Relació de les activitats realitzades.
  - 2.2.1 Activitats previstes en el pla de treball.
  - 2.2.2 Activitats no previstes i realitzades o programes.
- 2.3. Relació de les desviacions en la temporització i accions de mitigació si escau i actualització del cronograma si escau.
- 2.4. Llistat dels resultats parcials obtinguts fins al moment .
- 2.5. Comentaris del vostre director particular si ho considereu necessari.

**Pac4** (19/12/2017 - 02/01/2018)

Redacció de la memòria.

**Pac5a** (03/01/2018 - 10/01/2018)

Elaboració de la presentació.

**Pac5b** (03/01/2018 - 10/01/2018)

Defensa pública.

**Lliurament Final**

1. Memòria
2. Presentació
3. Vídeo de defensa
4. Fitxa d'autoavaluació

## 2. Definicions de conceptes

---

Per continuar amb el desenvolupament del projecte cal definir: Què és la intel·ligència artificial, quin és l'abast dels projectes actuals i com s'ha de d'enfocar; d'altra banda, és interessant saber com aprenen els humans, que tipus de memòries existeixen i com les podem simular; finalment, què és Python, per què és aconsellable la seva aplicació en qualsevol aplicació d'aquest tipus, i és imprescindible fer comparatives amb altres paquets de programari, des del punt de vista de l'estudi de costos.

### 2.1. Què es la Intel·ligència artificial?

---

Col·loquialment, el terme intel·ligència artificial s'aplica quan una màquina imita les funcions «cognitives» que els humans s'associen amb altres humans, com, per exemple: "aprendre" i "resoldre problemes". Per tant, per definició, la intel·ligència artificial (IA), també anomenada intel·ligència computacional, és la intel·ligència que simulen les màquines, que implica desenvolupament o aprenentatge interactiu. Un sistema d'intel·ligència artificial hauria de resoldre problemes pel que és fonamental en el seu disseny la delimitació dels tipus de problemes que resoldrà i les estratègies i algoritmes que utilitzarà per trobar la solució.

Per una altra banda, en els humans, la capacitat de resoldre problemes té dos aspectes: els aspectes innats i els aspectes apresos. Els aspectes innats permeten, per exemple, emmagatzemar i recuperar informació en la memòria, mentre que en els aspectes apresos resideix el saber resoldre un problema matemàtic mitjançant l'algoritme adequat. De la mateixa manera que un humà ha de disposar d'eines que li permetin solucionar certs problemes, els sistemes artificials han de ser programats de manera tal que puguin arribar a resoldre'ls.

Un dels majors problemes en sistemes d'intel·ligència artificial és la comunicació amb l'usuari. Aquest obstacle és causa de l'ambigüitat del llenguatge. La capacitat dels humans per comunicar-se entre sí que implica el coneixement del llenguatge que utilitza l'interlocutor. Perquè un humà pugui comunicar-se amb un sistema intel·ligent hi ha dues opcions: o bé que l'humà aprengui el llenguatge del sistema com si aprengués a parlar qualsevol altre idioma diferent al natiu, o bé que el sistema tingui la capacitat d'interpretar el missatge de l'usuari en la llengua que l'usuari utilitza.

Així mateix, Un sistema d'intel·ligència artificial s'enfronta amb el mateix problema que un humà. Aquest últim, durant tota la seva vida, aprèn el vocabulari de la seva llengua nativa o materna, sent capaç d'interpretar els missatges utilitzant el context. No obstant això, ha de conèixer els diferents significats per poder interpretar certs llenguatges especialitzats i tècnics que són coneguts només per experts en les respectives disciplines.

Com a conclusió, en aquest projecte no es pretén crear un sistema d'IA perfecte, que serveixi per a tot tipus d'usuari de manera independent dels seus coneixements i formació; sinó tot el contrari, la idea és que el sistema aprengui tot allò que l'usuari individual, dins del seva imperfecció, desitgi o necessiti. Aquest procés d'aprenentatge deu ser gradual en el temps i constant durant tots els dies. Igual que moltes persones intenten completar la seva formació acadèmica complerts els 50 anys d'edat.

## 2.2. Com aprenen els humans?

---

El procés d'aprenentatge que desenvolupem els éssers humans varia d'acord a les característiques de l'ambient que l'envolta, els estímuls rebuts al llarg de la seva vida i la disposició que té cap al mateix.

D'acord amb el que han descobert els estudiosos en el camp de la neurociència, els processos d'aprenentatge tenen el seu inici en les xarxes de neurones interconnectades, on cadascuna de les seves parts compleix una

funció específica al llarg de la vida de l'individu i que actua de diferent manera segons l'edat del mateix.

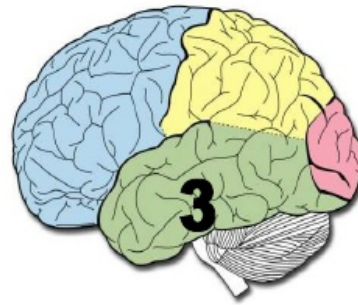
És precisament, en el cervell on es duen a terme totes les activitats que controlen i modifiquen el comportament de l'individu a través de la gran quantitat de neurones i cèl·lules que estan interconnectades entre si, en on cadascuna d'elles té una funció particular i va a promoure un procés d'aprenentatge en les diferents etapes de vida de la persona.

Un aspecte fonamental per desenvolupar processos d'ensenyament adequats, és tenir molt clar que els ambients que envolten a l'individu són essencials per al seu aprenentatge, la quotidianitat i conegut són eines que es poden utilitzar per reafirmar conceptes, idees o promoure el desenvolupament de noves idees i condicions.

Com a resum direm que, hi ha tres formes bàsiques en què aprenem els éssers humans i alguns principis que regeixen l'aprenentatge:

- Aprenem per les conseqüències que es deriven de la nostra conducta. Quan ens enfrontem a un problema nou, vam començar a provar conductes de tal manera que aquelles que ens permeten solucionar el problema les incorporem en el nostre repertori conductual per usar-les en el futur, mentre que aquelles que no ens serveixen per solucionar el problema les descartem.
- Aprenem per imitació. En observar com es comporten certes persones i com aconsegueixen el que volen emetent cert tipus de conductes podem aprendre que imitant aquests comportaments nosaltres també podem aconseguir aquestes mateixes conseqüències
- Aprenem seguint instruccions. Quan alguna cosa ho aprenem per instruccions, ja siguin orals (una explicació a classe) o escrites (seguint un llibre, etc.) aquest aprenentatge sol ser més ràpid. No obstant això, la possibilitat d'oblit és també més ràpida i més gran.

**Su función es la memoria. El lóbulo temporal dominante recuerda las palabras y nombres de los objetos. El lóbulo temporal no dominante, contiene la memoria visual (caras, imágenes,...).**



Il·lustració 4. Del lòbul frontal (3), on es centralitza la memòria<sup>4</sup>.

En conclusió, l'aprenentatge és el procés mitjançant el qual s'adquireix una determinada habilitat, s'assimila una informació o s'adopta una nova estratègia de coneixement i es realitza una acció. Partint d'aquestes bases, aquest projecte es basarà en un aprenentatge on s'associarà una imatge a un so i un text. D'aquesta manera, cada individu (usuari) serà l'encarregat (administrador) de formar el sistema, partint dels seus coneixements, del seu idioma, de les seves imperfeccions, i els seus àmbits socials; en definitiva, el sistema aprendrà el que ens interessi.

### 2.3. Per què Python?

Per definició, Python<sup>5</sup> és un llenguatge de programació “multi paradigma”. Això significa que més que forçar als programadors a adoptar un estil particular de programació, permet diversos estils: programació orientada a objectes<sup>6</sup>, programació imperativa<sup>7</sup> i programació funcional<sup>8</sup>. Python usa tipat dinàmic<sup>9</sup> i conte de referències<sup>10</sup> per a l'administració de memòria.

<sup>4</sup> <https://es.slideshare.net/ELIZABETHMARIASALAZAR/cmo-aprenden-las-personas-desde-la-neurociencia>

<sup>5</sup> <http://docs.python.org.ar/tutorial/pdfs/TutorialPython3.pdf>

<sup>6</sup> La programació orientada a objectes (POO, o OOP en anglès) és un paradigma de programació que ve a innovar la forma d'obtenir resultats. Els objectes manipulen les dades d'entrada per a l'obtenció de dades de sortida específics, on cada objecte ofereix una funcionalitat especial.

<sup>7</sup> La programació imperativa és un dels paradigmes de programació d'ordinadors més generals, que descriu la programació en termes de l'estat del programa i sentències que canvien aquest estat. Els programes imperatius són un

Una característica important de Python és la resolució dinàmica de noms; és a dir, el que connecta un mètode i un nom de variable durant l'execució del programa (també anomenat enllaç dinàmic de mètodes). Un altre objectiu del disseny del llenguatge és la facilitat d'extensió. Es poden escriure nous modificacions fàcilment en C o C++. Python pot incloure's en aplicacions que necessiten una interfície programable. Encara que la programació en Python podria considerar-se en algunes situacions hostils a la programació funcional tradicional del Lisp, existeixen bastants analogies entre Python i les llengües minimalistes de la família Lisp com es pot ser 'scheme'.

A continuació, es necessita la comparació amb Java per respondre a la pregunta: per què Python?

- La sintaxi de Python està dissenyada per ser intuïtiva i la seva simplicitat relativa permet als principiants començar ràpidament a escriure codi per a diverses aplicacions. Mentre que Java té una corba d'aprenentatge més marcada, és extremadament útil per desenvolupar aplicacions que s'executaran en qualsevol plataforma.
- Una de les diferències més grans entre Python i Java és la forma en què cada llengua maneja les variables. Java obliga a indicar el tipus d'una variable quan declara la primera vegada i no li permetrà canviar el tipus més tard en el programa. Això es coneix com tipat estàtic. Per contra, Python usa el teclat dinàmic, que li permet canviar el tipus d'una variable, reemplaçant un entero per una cadena.

Python	
Codi	Resultat
n=11 print (n) n='Hola'	11 Hola 1.5

conjunt d'instruccions que li indiquen al computador com realitzar una tasca, de la mateixa manera que la manera imperatiu en els llenguatges naturals humans li diu què fer a l'interlocutor.

<sup>8</sup> En ciències de la computació, la programació funcional és un paradigma de programació declarativa basat en l'ús de funcions matemàtiques, en contrast amb la programació imperativa, que emfatitza els canvis d'estat mitjançant la mutació de variables.

<sup>9</sup> Un llenguatge de programació és dinàmicament tipat si una mateixa variable pot prendre valors de diferent tipus en diferents moments. La majoria de llenguatges de tipus dinàmics són llenguatges interpretats, com Python o Ruby. Un llenguatge que no és dinàmicament tipat es diu que és de tipat estàtic, o estàticament tipat.

<sup>10</sup> Conte de referències, en anglès "Reference counting", és una tècnica per comptabilitzar les vegades que un determinat recurs està sent referit.

<pre>print (n) n=1.5 print (n)</pre>	
--------------------------------------	--

Taula 4. Codi en Python que mostra que fa servir tipat estàtic.

- Python és un poc estrany quan es fa servir '*indentació*'<sup>11</sup> per separar el codi en blocs. Java, com la majoria d'altres idiomes, usa les claus per definir el principi i el final de cada funció o definició de classe. L'avantatge d'usar '*indentació*' és que obliga a construir el seu programa de forma que sigui fàcil de llegir, i no hi haurà cap error derivat de que falti una clau.
- La gran avantatge de Java és que es pot utilitzar per crear aplicacions independents de la plataforma. Qualsevol ordinador o dispositiu mòbil que pugui executar una màquina virtual de Java pot executar una aplicació Java, mentre que per executar programes, Python necessita un compilador que pot convertir el codi en Python a un codi que el seu operador en particular pugui entendre. El inconvenient d'executar-se dins d'una màquina virtual és que el programa Java s'executa més lentament que els programes en Python.

Taula resum comparativa	
Python	Java
<ul style="list-style-type: none"> <li>- Tipificació dinàmica.</li> <li>- Utilitza els espais en blanc per transmetre el principi i el final dels blocs de codi.</li> <li>- Els programes són petits i per tant corren molt més ràpid.</li> <li>- Necessita menys codi per crear un programa.</li> <li>- Aquest llenguatge és lent en l'execució.</li> <li>- Es pot compilar un codi de bytes natiu.</li> <li>- Podeu assignar una cadena de caràcters a una variable que una vegada va tenir un total.</li> </ul>	<ul style="list-style-type: none"> <li>- Tipus estètics.</li> <li>- Usa claudàtors per indicar el començament i el final de les funcions.</li> <li>- Els programes són més grans.</li> <li>- No es compila un codi natiu.</li> <li>- Es pot executar en qualsevol sistema operatiu que pugui executar la màquina virtual de Java.</li> <li>- No es poden canviar els tipus de dades de variables.</li> <li>- La programació orientada a objectes és obligatòria.</li> </ul>

<sup>11</sup> *Indentació*: és un anglicisme (de la paraula anglesa *indentation*). Aquest terme significa moure un bloc de text a la dreta inserint espais o tabuladors, per així separar-lo del marge esquerre i millor distingir-lo del text adjacent.



<ul style="list-style-type: none"> <li>- Més fàcil de llegir i entendre amb relació a Java.</li> <li>- No es suporta a través d'una àmplia varietat de plataformes.</li> <li>- La programació orientada a objectes és opcional.</li> </ul>	
--	--

Taula 5. Comparativa de les principals característiques diferencials entre Java i Python.

Però hi ha tres punts que s'han de destacar en la implementació d'aquest projecte:

- El primer, en Python la programació és més productiva. Exemple: Si definim una aplicació buida que no faci res, en Python només tenim amb definir una funció; en canvi, en Java, hem de definir la classe, realitzar un mètode d'iniciació buit i després definir els "setters" i "getters". Per tant, des del punt de vista dels costos, Python resulta força més interessant.

Python	Java
<pre>def mi_funcion (nombre, apellido):</pre>	<pre>public class MiClase {     //Atributos de la clase     private String atributo1;     private int atributo 2;     private float atributo 3;      //Constructor con el mismo nombre de la clase     public MiClase(){}</pre> <pre>    //Métodos de la clase     public void metodo1()     {         //Método vacio     }      public String metodo2()     {         return "metodo2";     } }</pre>

Taula 6. Comparativa des de el punt de vista dels costos.

- El segon, Python es poden utilitzar paràmetres arbitraris. és possible definir una funció que esperi rebre un nombre arbitrari (desconegut) d'arguments.

Parámetros arbitrarios
<pre>def recorrer_parametros_arbitrarios(parametro_fijo, *arbitrarios):     print parametro_fijo      # Los parámetros arbitrarios se corren como tuplas     for argumento in arbitrarios:         print argumento</pre>

Taula 7. Exemple de paràmetres arbitraris en Python.

- L'últim, és la recursivitat. Són aquelles funcions que, si algoritme, fan referència si mateixa.

Crides recursives
<pre>def jugar(intento=1):     respuesta = raw_input("¿De qué color es una naranja? ")     if respuesta != "naranja":         if intento &lt; 3:             print "\nFallaste! Inténtalo de nuevo"             intento += 1             jugar(intento) # Llamada recursiva         else:             print "\nPerdiste!"     else:         print "\nGanaste!" jugar()</pre>

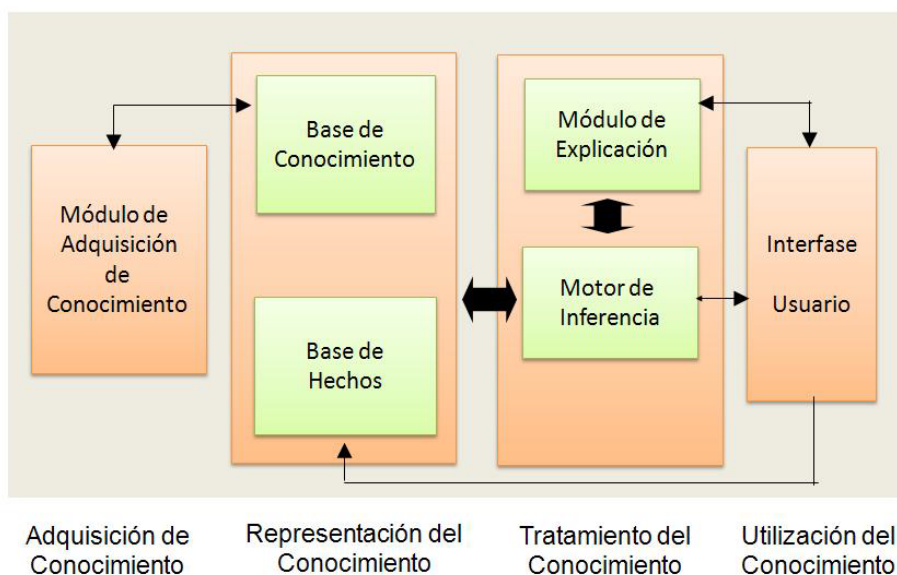
Taula 8. Exemple de crides amb recursivitat en Python.

En conclusió, Python és un llenguatge de programació poderós i fàcil d'aprendre. Compta amb estructures de dades eficients i d'alt nivell i un enfocament simple però efectiu a la programació orientada a objectes. L'elegant sintaxi de Python i el seu tipatge dinàmic, juntament amb la seva naturalesa interpretada, fan d'aquest un llenguatge ideal per "scripting" i desenvolupament ràpid d'aplicacions en diverses àrees i sobre la majoria de les plataformes. Però, sobretot, cal tenir en compte les seves virtuts, Python és millor per a manejar gran quantitat de dades (en anglès "Big Data") i en intel·ligència artificial; en canvi Java és idoni per a aplicacions web.

### 3. Breu descripció de sistemes de IA

La intel·ligència artificial (IA), també anomenada intel·ligència computacional, és la intel·ligència exhibida per màquines. En ciències de la computació, una màquina "intel·ligent" ideal és un agent racional flexible que percep el seu entorn i porta a terme accions que maximitzin les seves possibilitats d'èxit en algun objectiu o tasca. Col·loquialment, el terme intel·ligència artificial s'aplica quan una màquina imita les funcions "cognitives" que els humans s'associen amb altres ments humanes, com per exemple: "aprendre" i "resoldre problemes".

A si mateix, els sistemes experts utilitzats en intel·ligència artificial són programari que emula el comportament d'un expert humà en la solució d'un problema. Els sistemes experts funcionen de manera que emmagatzemen coneixements concrets per a un camp determinat i solucionen els problemes, utilitzant aquests coneixements, mitjançant deducció lògica de conclusions. Amb ells es busca una millora en qualitat i rapidesa de respostes donant així lloc a una millora de la productivitat de l'expert.



Il·lustració 5. Arquitectura i funcionament dels sistemes experts<sup>12</sup>.

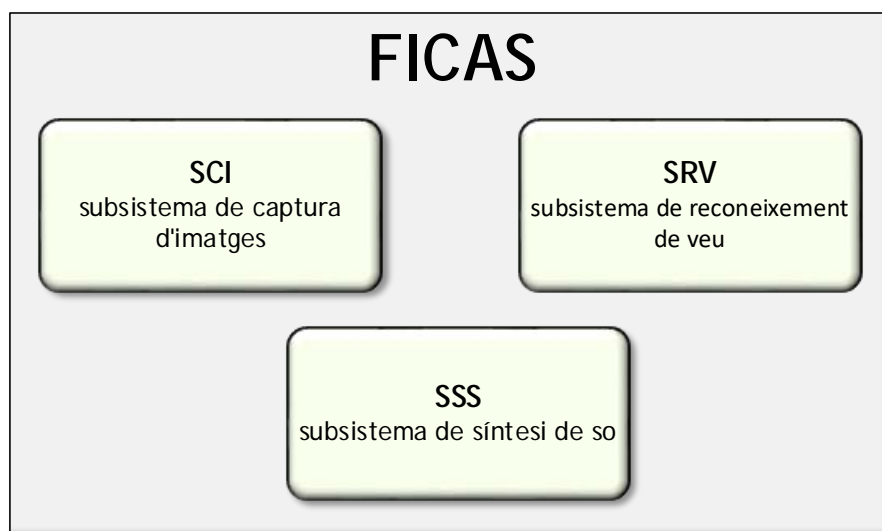
<sup>12</sup> <https://maricelamaldonado.wordpress.com/2011/03/23/arquitectura-de-un-sistema-experto/>

També, les xarxes neuronals són un paradigma d'aprenentatge i processament automàtic inspirat en la forma en què actua el sistema nerviós dels animals. La unitat bàsica de processament de les xarxes neuronals està inspirada en la cèl·lula fonamental del sistema nerviós humà, la neurona. En aquest sentit, les xarxes neuronals no són més que una altra forma d'emular certes característiques pròpies dels humans.

Un altre camp d'aplicació és, la robòtica, la ciència o branca de la ciència que s'ocupa de l'estudi, desenvolupament i aplicacions dels robots. Els robots són màquines controlades per ordinador i programades de manera que siguin capaços de moure, manipular objectes i realitzar treballs al mateix temps que interactuen amb l'entorn. Els robots solen ser programats per a estalviar temps i substituir els humans en treballs costosos, pesats, complicats ... i repetitius, per això, són capaços de realitzar tasques repetitives de forma més ràpida, barata i precisa que els éssers humans.

Finalment, La cibernètica és la ciència que estudia els sistemes de control i comunicació de les persones i les màquines. D'aquesta manera s'aprofiten tots els aspectes i mecanismes comuns.

Con a conclusió, per al desenvolupament d'aquest projecte s'estudiarà la viabilitat de integració dels sistemes de: captura de fotos, síntesi de so i reconeixement de veu, per fer una simulació de com aprenen els humans.



II·lustració 6. Integració de sistemes

## 3.1 Sistema de captura d'imatges

---

En aquest projecte el sistema de captura de fotos passarà a nomenar se com subsistema de captura d'imatges, amb les sigles SCI, per simplificar la introducció de la informació en els posterior apartats aquest.

### 3.1.1 Definició

---

Dels cinc sentits que posseeix l'home, probablement la visió sigui el més important per la utilitat i satisfacció que proporciona: ens aporta immediatament un marc de referència de la posició que ocupem en l'espai, i ens permet avaluar tot el que ens envolta d'acord a la seva forma, mida i color; a més, com està constituïda per un sensor remot com és l'ull humà i pel cervell, tal avaluació es pot fer a distància, el que significa que les decisions i accions que executa un ésser humà, fins i tot estant en moviment, estan relacionades amb el medi que l'envolta; és a dir, poden ser fetes sobre la marxa o desenvolupament d'una tasca. L'emulació d'això per una màquina per mitjà d'un sensor optoelectrònic i un ordinador degudament programada constitueix un bon exemple de la intel·ligència artificial que, plasmada en la robòtica.

Per simular com aprenen els humans aquest projecte es basarà en la captura, i la posterior comparació, d'una sèrie d'imatges, que es guardaran, un cop processades, amb un nom (veure Il·lustració 6). Per a això, és molt important partir d'una bona captura de fotos, que en un principi es realitzarà amb una càmera (web) amb format RGB; però amb aquest format si comparem dues fotos, píxel a píxel, per saber si són idèntiques, obtindrem un gran percentatge de fallades, pel fet que un petit moviment en la captura de la imatge la posició de cada píxel té valors diferents.



II·lustració 7. Comparativa de fotos<sup>13</sup> en format RGB, amb un petit desplaçament en el quadri.

### Codi amb Java

```
private int Comparar_Fotos(BufferedImage i, BufferedImage f) {
    //Crea una copia del mismo tamaño que la imagen
    //System.out.println("\tComparador
"+i.getColorModel()+"\t"+f.getColorModel());
    int cont=0;
    for (int x=0;x < i.getWidth();x++){
        for (int y=0;y < i.getHeight();y++){
            if (i.getRGB(x, y) != f.getRGB(x, y)){
                cont++;
            }
        }
    }
    porcentaje=(100*cont)/76800;
    System.out.println("\tFotos comparadas con un porcentaje de:
"+porcentaje+" % - de: "+cont);
    return porcentaje;
}
```

Taula 9. Codi amb Java per comparar dues fotos píxel a píxel.

Per obtenir un millor resultat en la comparativa de les fotos, les passarem a escala de grisos per aconseguir un menor nombre d'errors quan comparem les imatges píxel a píxel.

<sup>13</sup> Fotògraf: Francisco Cortés Sánchez



Il·lustració 8. Imatge convertida a escala de grisos.

**Codi amb Java**

```
public BufferedImage RGB_Gris(BufferedImage f){
    //Crea una copia del mismo tamaño que la imagen
    BufferedImage fDestino = new
    BufferedImage(f.getWidth(),f.getHeight(), BufferedImage.OPAQUE);
    for (int x=0;x < f.getWidth();x++){
        for (int y=0;y < f.getHeight();y++){
            //Obtiene el color
            Color c1=new Color(f.getRGB(x, y));
            //Calcula la media de tonalidades
            int med=(c1.getRed()+c1.getGreen()+c1.getBlue())/3;
            //Almacena el color en la imagen destino
            fDestino.setRGB(x, y, new Color(med,med,med).getRGB());
        }
    }
    return fDestino;
}
```

Taula 10. Codi Java per fer la conversió d'una imatge en RGB a escala de grisos.

El següent pas que podem fer és convertir la imatge en blanc i negre, amb el que s'obté una imatge en el mapa de bits, amb valors zero o un (0,1), que comporta un millor resultat a l'hora de realitzar una comparativa d'imatges.



II-lustració 9. Imatge convertida a blanc i negre.

**Codi amb Java**

```
public BufferedImage set_Blanco_y_Negro_con_Umbra1(BufferedImage f){
    BufferedImage fDestino = new
    BufferedImage(f.getWidth(),f.getHeight(), BufferedImage.TYPE_BYTE_BINARY);
    //se traspasan los colores Pixel a Pixel
    for(int i=0;i<f.getWidth();i++){
        for(int j=0;j<f.getHeight();j++){
            color = new Color(f.getRGB(i, j));
            //se extraen los valores RGB
            int R = color.getRed();
            int G = color.getGreen();
            int B = color.getBlue();
            //dependiendo del valor del umbral, se van separando los
            // valores RGB a 0 y 255
            R=(R>umbral)? 255: 0;
            G =(G>umbral)? 255: 0;
            B =(B>umbral)? 255: 0;
            fDestino.setRGB(i, j, new Color(R,G,B).getRGB());
        }
    }
    return fDestino;
}
```

Taula 11. Codi Java per fer la conversió d'una imatge en RGB a blanc i negre.

Analitzant tot l'anterior, dona la sensació que falta alguna cosa perquè aquest subsistema funcioni adequadament. Si es defineix les imatges en tons



grisos com matrius de 2 dimensions ( $2D^{14}$ ), i les imatges en RGB com a matrius de 3 dimensions ( $3D^{15}$ ); a continuació, si comparem aquestes matrius, dos a dos (2D amb 2D y 3D amb 3D) obtindrem una zona, concreta i, comú a les dues imatges que és completament diferent; i si això mateix ho repetim diverses vegades aquesta zona en concret anirà canviant la seva posició, amb això s'aconsegueix delimitar una part molt concreta d'una successió d'imatges que ens servirà per reconèixer cares o objectes. Per tant, el que ens interessa és, que a partir d'una entra vídeo, aplicar un algoritme de comparació de matrius per poder extreure la zona que està en moviment dins d'una escena.

Això s'aconseguirà afegint la llibreria OpenCV<sup>16</sup> a Python per obtenir el següent resultat.



Il·lustració 10. Imatge<sup>17</sup> de simulació de visió artificial

### 3.1.2 Funcionament

El SCI funcionarà com la visió dels humans: percebem una seqüència d'imatges, captades per l'ull, comprovem que les reconeixem i les guardem en la nostra memòria. Per tant, el SCI el dividirà en dues parts molt significatives: el subsistema de formació (captació) d'imatges (o objectes)

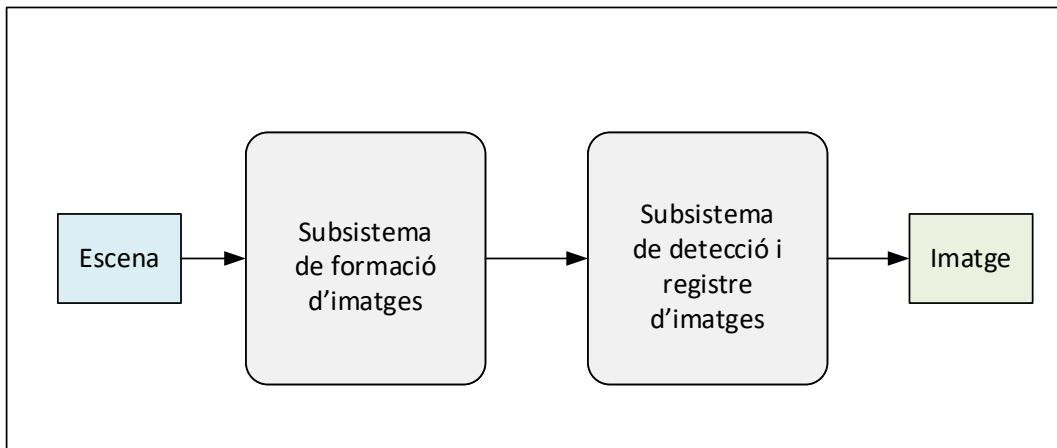
<sup>14</sup> *Imatges en 2D.* Per a cada coordenada  $(x, y)$  li correspon un to de gris  $(0,255)$ .

<sup>15</sup> *Imatges en 3D.* Per a cada coordenada  $(x, y)$  li correspon un valor de  $R(0,255) G(0,255) B(0,255)$ .

<sup>16</sup> Per realitzar la instal·lació veure l'annex A5.

<sup>17</sup> <https://es.wikipedia.org/wiki/OpenCV>

capturarà una foto de la part mòbil d'una seqüència d'imatges (Escena); la passarà al subsistema de detecció i registre d'imatges, que tindrà la funció de verificar, comparar la imatge amb les fotos guardades al registre d'imatges; y retornarà tota la informació assignada a aquesta foto y en cas contrari la guardarà en el registre.



II-lustració 11. Diagrama de blocs que mostra el procés de captura d'una imatge.

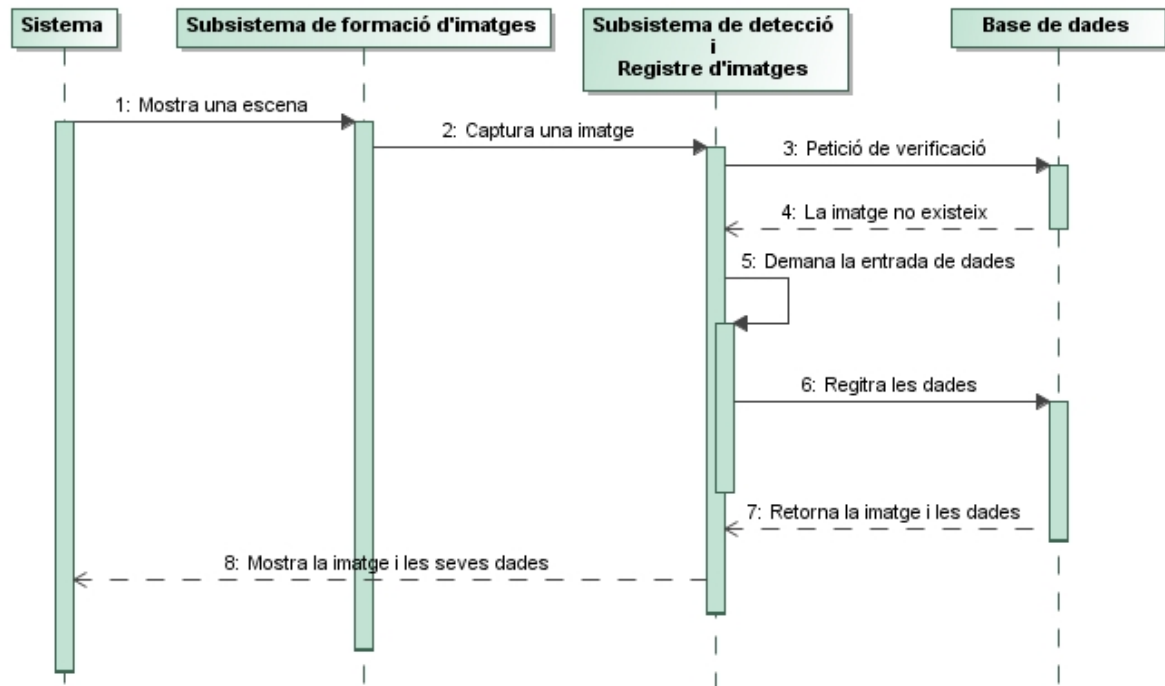
### 3.1.3 Cas d'ús i diagrama de seqüència

#### Cas d'us Gestió SCI

<i>Resum de funcionalitat:</i>	Permet registrar, verificar i comparar imatges al sistema.
<i>Actors:</i>	Admin. (privilegis de creació, eliminació, modificació i consulta), Usuari. (privilegis de consulta)
<i>Casos d'ús relacionats:</i>	Gestió de reconeixement de veu i Gestió síntesis de so.
<i>Precondició:</i>	L'usuari es vàlid i s'ha identificat correctament en el sistema
<i>Postcondició:</i>	Les accions s'ha realitzat correctament, en el cas de ser l'usuari Admin les modificacions/altes/baixes han quedat enregistrades en la base de dades.
<i>Procés normal principal:</i>	1. El sistema presenta una pantalla, on es veu una finestra amb una escena capturada per un dispositiu de captura de vídeo

	<p>(WebCam)</p> <ol style="list-style-type: none"> <li>2. L'usuari, és Admin, i quan veu que la foto es la correcta confirma la captura       <ol style="list-style-type: none"> <li>2.1. El sistema mostra la foto capturada i espera confirmació</li> <li>2.2. L'usuari és Admin, si es correcta prem el botó de "Return"</li> </ol> </li> <li>3. El sistema compara i verifica si la foto ja està guardada en el registre       <ol style="list-style-type: none"> <li>3.1. En cas que sigui negatiu, el sistema demanarà totes les dades referent a la imatge per enregistrar-les al sistema           <ol style="list-style-type: none"> <li>3.1.1. Introdueix el nom de la imatge</li> <li>3.1.2. Introdueix totes les dades dels paràmetres arbitraris</li> <li>3.1.3. Demana confirmació de les dades</li> </ol> </li> </ol> </li> <li>4. L'usuari, és Admin, confirmarà que les dades introduïdes siguin correctes i donarà la conformitat.</li> <li>5. El sistema guardarà totes les dades al registre i tornarà al punt 1</li> </ol>
<p><i>Alternatives de procés i excepcions:</i></p>	<ol style="list-style-type: none"> <li>2.2.1 En cas contrari el sistema torna al punt 1</li> <li>3.2. En cas que sigui afirmatiu, el sistema mostrarà totes les dades enregistrades al sistema referent a la imatge</li> <li>4.1 En cas contrari el sistema torna al punt 1</li> </ol>

Taula 12. Descripció del cas d'us del subsistema de captura d'imatges (SCI).



II·lustració 12. Diagrama de seqüència del subsistema SCI

## 3.2 Sistema de reconeixement de veu

També, aquest sistema de reconeixement de veu, passarà a nomenar se com subsistema reconeixement de veu, amb les sigles SRV, per fer més fàcil la introducció de la informació en els posterior apartats.

### 3.2.1 Definició

El reconeixement automàtic de la parla (RAH) o reconeixement automàtic de veu és una disciplina de la intel·ligència artificial que té com a objectiu permetre la comunicació parlada entre éssers humans i computadores. Un sistema de reconeixement de veu és una eina computacional capaç de processar el senyal de veu emesa per l'ésser humà i reconèixer la informació continguda en aquesta, convertint-la en text o emetent ordres que actuen sobre un procés. Per dissenyar un sistema de RAH s'ha d'enfocar segons el tipus d'aprenentatge, que poden ser de dos tipus:

- **Aprenentatge deductiu:** Les tècniques d'Aprenentatge Deductiu es basen en la transferència dels coneixements que un expert humà posseeix a un sistema informàtic.
- **Aprenentatge inductiu:** Les tècniques d'Aprenentatge Inductiu es basen en el fet que el sistema pugui, automàticament, aconseguir els coneixements necessaris a partir d'exemples reals sobre la tasca que es vol modelitzar.

### 3.2.2 Funcionament

---

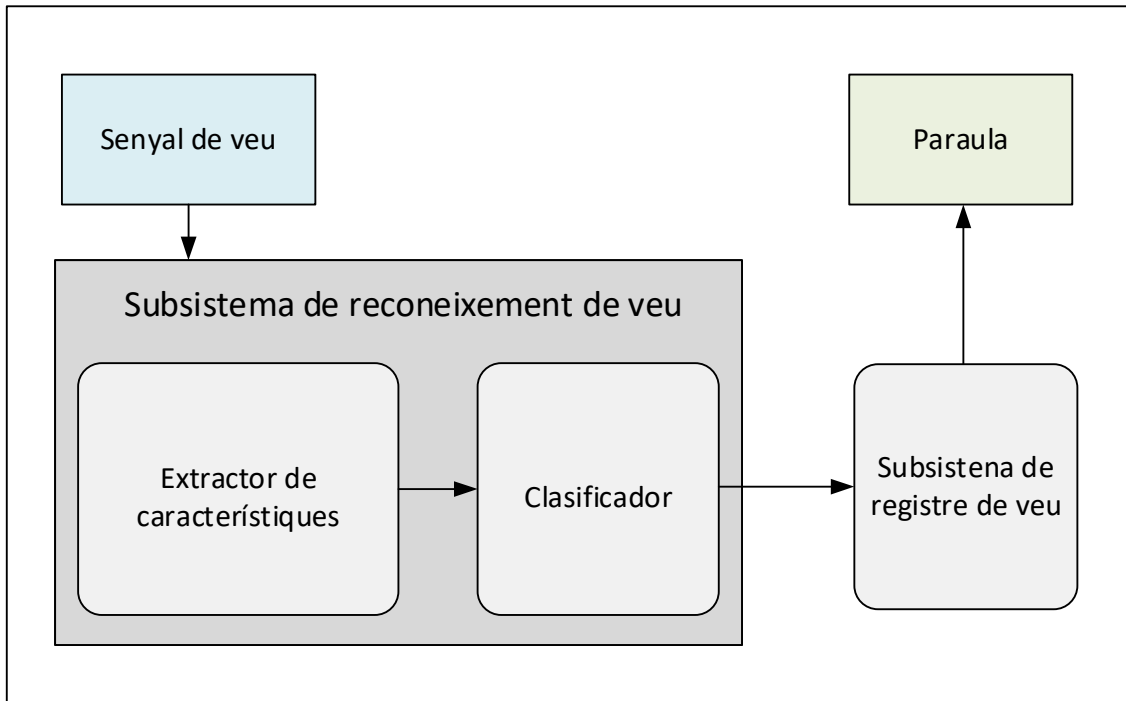
Per entendre el funcionament d'un sistema de reconeixement de veu és necessari conèixer els seus principals components: l'extractor de característiques i el classificador. Quan es rep el senyal de veu, aquesta passa per un reconeixedor el qual dona com a resultat la paraula que reconeix. Després hi ha un processament del llenguatge natural, una representació semàntica i finalment es realitza una acció. En l'arquitectura d'un sistema de reconeixement de veu es compta amb dos processos importants en la fase de reconeixement, aquests són els següents:

- **Extracció de característiques:**
  1. El senyal es divideix en una col·lecció de segments.
  2. S'aplica alguna tècnica de processament de senyals per obtenir una representació de les característiques acústiques més distintives de segment.
  3. Amb base a les característiques obtingudes, es construeix un conjunt de vectors que constitueixen l'entrada al següent mòdul.
- **Classificador probabilístic:**
  1. Es crea un model probabilístic basat en xarxes neuronals com a models ocults de Markov<sup>18</sup>, etc.

---

<sup>18</sup> Els models ocults de Markov són especialment aplicats a reconeixement de formes temporals, com a reconeixement de la parla, d'escriptura manual, de gestos, etiquetatge gramatical o en bioinformàtica.

2. Amb les probabilitats obtingudes es realitza una recerca per trobar la seqüència de segments amb major probabilitat de ser reconeguts.



II-lustració 13. Diagrama de blocs que mostra el procés del subsistema de reconeixement de veu.

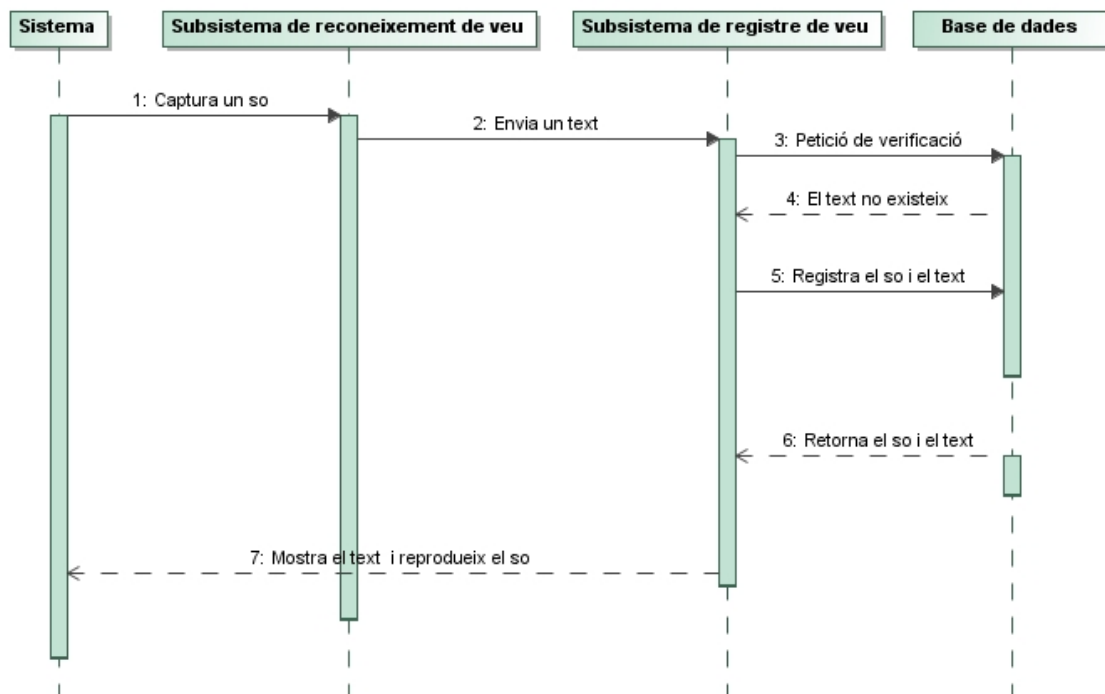
### 3.2.3 Cas d'ús i diagrama de seqüència

#### Cas d'ús Gestió SRV

<i>Resum de funcionalitat:</i>	Permet obtenir un text a partir d'una entrada de veu.
<i>Actors:</i>	Admin. (privilegis de creació, eliminació, modificació i consulta), Usuari. (privilegis de consulta)
<i>Casos d'ús relacionats:</i>	Gestió de reconeixement d'imatges i Gestió síntesis de so.
<i>Precondició:</i>	L'usuari es vàlid i s'ha identificat correctament en el sistema
<i>Postcondició:</i>	Les accions s'ha realitzat correctament, en el cas de ser l'usuari Admin les modificacions/altes/baixes han quedat enregistrades

	en la base de dades.
<i>Procés normal principal:</i>	<ol style="list-style-type: none"> <li>1. El sistema presenta una pantalla, on es veu una finestra amb una escena que representa una captura de veu</li> <li>2. L'usuari, és Admin, i quan veu que el sistema està a la espera, pronuncia una paraula.             <ol style="list-style-type: none"> <li>2.1. El sistema mostra la paraula capturada, la reproduueix i espera confirmació</li> <li>2.2. L'usuari és Admin, si es correcta ho confirma</li> </ol> </li> <li>3. El sistema compara i verifica si la paraula ja està guardada en el registre</li> <li>4. L'usuari, és Admin, confirmarà que les dades introduïdes siguin correctes i donarà la conformitat.</li> <li>5. El sistema guardarà totes les dades al registre i tornarà al punt 1</li> </ol>
<i>Alternatives de procés i excepcions:</i>	2.2.1 En cas contrari el sistema torna al punt 1

Taula 13. Descripció del cas d'us del subsistema de reconeixement de veu (SRV).



II·lustració 14. Diagrama de seqüència del subsistema SRV

## 3.3 Sistema de síntesis de so

---

Així mateix, aquest sistema, de síntesi de so, passarà a nomenar-se com subsistema síntesi de so, amb les sigles SSS, per facilitar la introducció de la informació en els posterior apartats.

### 3.3.1 Definició

---

Per definició, la síntesi de parla és la producció artificial de la parla. El sistema computat que és usat amb aquest propòsit és anomenat ordinador de parla o sintetitzador de veu i pot ser implementat en productes software i hardware. Un sistema text-to-speech<sup>19</sup> (TTS) converteix el llenguatge de text normal en parla; altres sistemes recreen la representació simbòlica lingüística com transcripcions fonètiques en parla. Així mateix, la parla sintetitzada pot ser creada a través de la concatenació de fragments de parla gravats que són emmagatzemats en una base de dades.

### 3.3.2 Funcionament

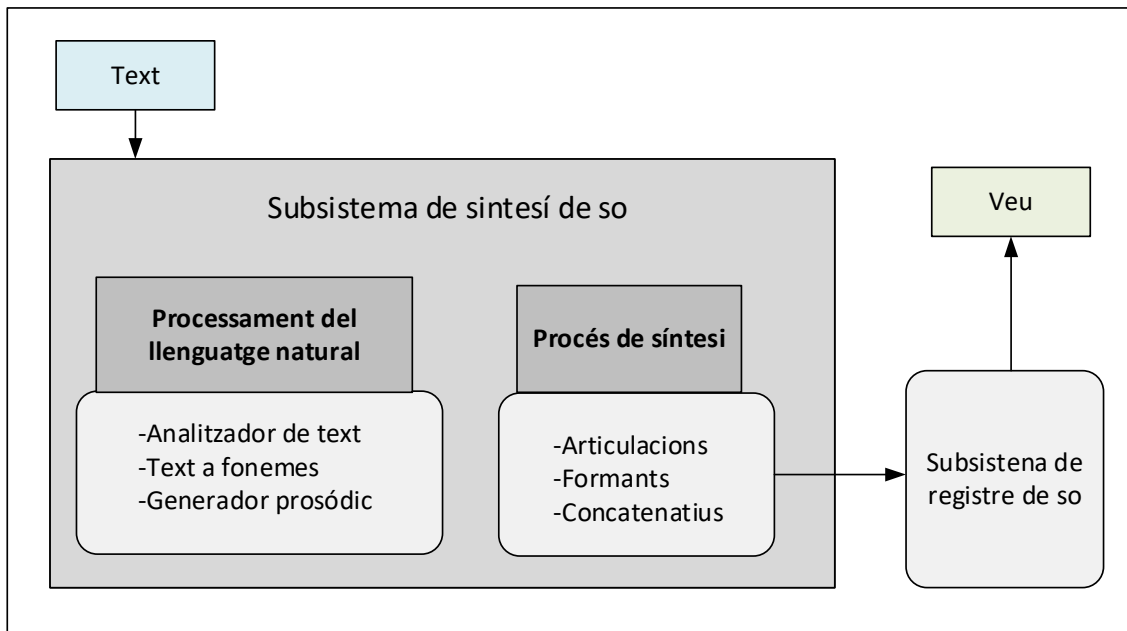
---

Per entendre el funcionament d'un sistema de síntesi de veu és necessari conèixer els seus principals components: Processament del llenguatge natural i procés de síntesi.

---

<sup>19</sup> La conversió text-veu és la generació per mitjans automàtics d'una veu artificial que genera el so produït per una persona en llegir un text qualsevol en veu alta o una veu artificial. És a dir, són sistemes que permeten la conversió de textos en veu sintètica. Els convertidors de text-veu són coneguts també amb les sigles CTV o per les sigles en anglès TTS (Text To Speech)





Il·lustració 15. Diagrama de blocs que mostra el procés del subsistema de síntesi de so.

- **Processament del llenguatge natural.** Aquest sub procés s'encarrega de produir una transcripció fonètica del text llegit, a més de l'entonació i el ritme desitjat per a la veu de sortida. El text d'entrada consisteix en una col·lecció de símbols que s'han d'interpretar com a paraules per tal de tenir la idea del que s'ha escrit. El NLP està dividit en les següents parts:
  - *Analitzador de text.* La seva funció principal és prendre com a entrada qualsevol text i donar-li el format d'una seqüència de paraules lliures de soroll i pauses entre frases.
  - *Convertidor de text a fonemes.* Per convertir-lo en fonemes cal assignar la seva pronunciació per a poder generar el senyal de veu.
  - *Generador prosòdic.* S'encarrega d'assignar la durada i l'entonació correcta a cada un dels fonemes.
- **Procés de síntesi.** Aquest sub procés s'encarrega de transformar la informació simbòlica que rep del NLP en una veu de sortida. Aquest

text d'entrada consisteix en una col·lecció de símbols que s'han d'interpretar com a paraules, hi ha diferents tipus:

- *Sintetitzadors articuladoris*. Són models físics basats en descripcions detallades de mecanismes fisiològics de producció de veu i generació de sons.
- *Sintetitzadors paramètrics*. Generen la veu variant paràmetres que s'apliquen senyals harmòniques.
- *Síntesi concatenada*. Forma la veu sintetitzada per mitjà de la concatenació de segments de veu prèviament emmagatzemats en una base de dades.

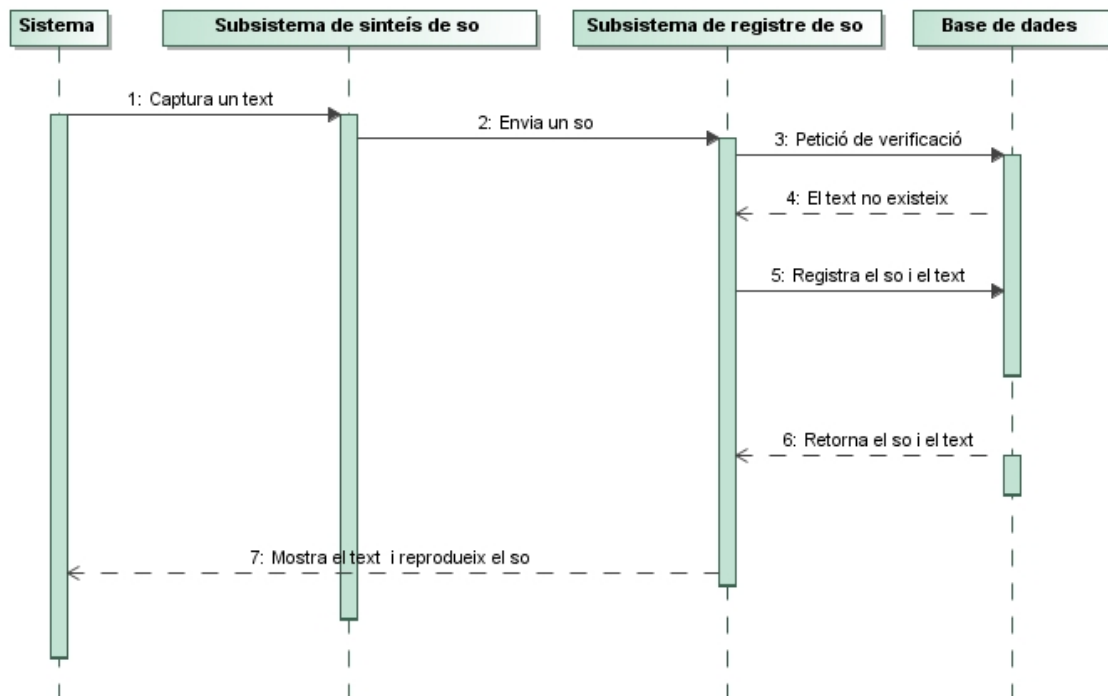
### 3.3.3 Cas d'ús i diagrama de seqüència

#### Cas d'us Gestió de SSS

<i>Resum de funcionalitat:</i>	Permet obtenir una veu a partir d'una entrada de text
<i>Actors:</i>	Admin. (privilegis de creació, eliminació, modificació i consulta), Usuari. (privilegis de consulta)
<i>Casos d'ús relacionats:</i>	Gestió de reconeixement d'imatges i Gestió síntesis de so.
<i>Precondició:</i>	L'usuari es vàlid i s'ha identificat correctament en el sistema
<i>Postcondició:</i>	Les accions s'ha realitzat correctament, en el cas de ser l'usuari Admin les modificacions/altes/baixes han quedat enregistrades en la base de dades.
<i>Procés normal principal:</i>	<ol style="list-style-type: none"> <li>1. El sistema espera una entrada de text</li> <li>2. L'usuari, és Admin, i quan veu que el sistema està a la espera, escriu la paraula. <ol style="list-style-type: none"> <li>2.1. El sistema mostra la paraula capturada, la reproduueix i</li> </ol> </li> </ol>

	<p>espera confirmació</p> <p>2.2. L'usuari és Admin, si es correcta ho confirma</p> <p>3. El sistema compara i verifica si la paraula ja està guardada en el registre</p> <p>4. El sistema guardarà totes les dades al registre i tornarà al punt 1</p>
<p><i>Alternatives de procés i excepcions:</i></p>	<p>2.2.1 En cas contrari el sistema torna al punt 1</p>

Taula 14. Descripció del cas d'us del subsistema de síntesis de so (SSS).



II-lustració 16. Diagrama de seqüència del subsistema SSS

## 4. Integració de sistemes

---

FICAS tindrà com a objectiu principal la simulació de l'ensenyament dels humans per mitjà de la integració dels sistemes d'intel·ligència artificial, com: captura d'imatges (SCI), reconeixement de la veu (SRV) i síntesi de so (SSS); per mitjà, de l'aplicació de recursivitat i l'ús de classes amb dades arbitràries. Per a això, en cada acció que es realitzi es substituirà l'entrada de text, en el teclat, per una entrada al SRV; i cada sortida de text, per pantalla, sota canviarà per una sortida del SSS.

L'accés complet als es definiran dos tipus de persones, que s'hauran de registrar mitjançant un reconeixement facial i la introducció de dades personals, que són: els usuaris i els administradors. Per a cada usuari el sistema registrarà: el tipus d'usuari, el nom, la seva foto i un registre de veu (com a paraula de pas). Aquests últims, tindran tots els privilegis necessaris per a realitzar: la definició dels continguts de les classes, la introducció, modificació i eliminació de dades; en canvi els usuaris, només podran interactuar amb el sistema introduint o traient informació i fer converses.

El sistema pretén que l'usuari li faciliti informació per aprendre d'allò que l'administrador ha configurat prèviament. Per tant, el sistema proporcionarà dos mòduls (grups d'informació) d'aprenentatge: els catàlegs, que són grups de dades on es registraran fotos amb dades de text i so (catàleg de fotos animals, insectes, plantes, minerals, lectura, etc.) ; i els diccionaris, el que s'associaran frases amb frase, paraules amb paraules de text (Preguntes i respostes, sinònims, traduccions, etc.)

El sistema gestionarà cada mòdul per separat; en els catàlegs, es demanarà el nom del grup, i les seves dades: que són el nom de la foto, i el nom de cada un dels camps arbitraris que es desitgi i les dades d'aquests camps. En canvi, en els diccionaris, només es demanarà el nom del grup i l'entrada de dades de dos camps fixos.

### Catàlegs

<b>Nom del grup: Animals Vertebrats</b>		<b>Nom del grup: Animals Invertebrats</b>	
<b>Nom de la Foto</b>	León	<b>Nom de la Foto</b>	Caragol
<b>Nom del camp 1:</b> Classe	Mamífers	<b>Nom del camp 1:</b> Classe	Mol·luscos
<b>Nom del camp 2:</b> Subclasse	Carnívors	<b>Nom del camp 2:</b> Subclasse	Terrestres
<b>Nom del camp 3:</b> Alimentació	Depredadors	<b>Nom del camp 3:</b> Classificació	Gasteròpodes

II-lustració 17. Exemple de classificació<sup>20</sup> del catàleg (“animals”)

### Diccionaris

<b>Nom del grup: Conversació</b>	<b>Nom del grup: Sinònims</b>
----------------------------------	-------------------------------

<sup>20</sup> <http://cmapspublic3.ihmc.us/rid=1MQLVPDC9-1SXHZY8-1512/CLASIFICACION%20ANIMALES.cmap>

Text1 - Text2	Text1 - Text2
---------------	---------------

Taula 15. Exemple de descripció dels camps arbitraris

## Paràmetres arbitraris en Python

Per definir arguments arbitraris en una funció, s'arriba al paràmetre un asterisc (\*)

És possible també, obtenir paràmetres arbitraris com parells de clau = valor.

En aquests casos, al nom del paràmetre han de precedir 2 asteriscos (\*\*)

```
def recorrer_parametros_arbitrarios(parametro_fijo, *arbitrarios, **kwargs):
    print parametro_fijo
    for argumento in arbitrarios:
        print argumento

    # Los argumentos arbitrarios tipo clave, se recorren como los diccionarios
    for clave in kwargs:
        print "El valor de", clave, "es", kwargs[clave]

recorrer_parametros_arbitrarios("Fixed", "arbitrario 1", "arbitrario 2",
"arbitrario 3", clave1="valor uno", clave2="valor dos")
```

Taula 16. Exemple de definició de paràmetres arbitraris en Python

El sistema haurà paraules clau que es guardaran amb la classe 'macros', s'introduiran amb el SRV i, serviran per realitzar accions, que seran: Conversar, Catàleg, Diccionari, Crear, Desar, Cancel·lar, Confirmar i Esborrar.

El sistema, quan se li d'una ordre (SRV), buscarà la paraula el bloc dels diccionaris i si la troba retornarà la seva paraula emparellada; és a dir, que si vam crear un diccionari de traduccions en anglès i introduim els emparellaments ('correct' - confirmar : 'ok' - confirmar) podrem donar-li la ordres en anglès, o en qualsevol idioma que ens interessi (Italià: 'confermare' - confirmar), per confirmar una acció.

Les converses son cerques de informació amb el 'tex1' dels diferents diccionaris con a preguntes, i com a respostes son la informació del 'text2' del mateix diccionari. Exemples: diccionari sinònims (text1="generós" : text2="esplèndid"); diccionari anglès (text1="Hola, com estàs?" : text2="Hello, how are you?"); diccionari converses (text1="Hoy va a llover" :

text2="creo que si, ya hace falta"); es a dir, si li preguntem al sistema "generós", el sistema respondrà "esplèndid"; etc. Si el sistema troba dos entrades iguals, respondrà aleatòriament una d'elles. A mes a mes, si es fa que el sistema es el que pregunta al usuari, i es canvia l'ordre d'intervenció, de forma aleatòria, simulem una conversa.

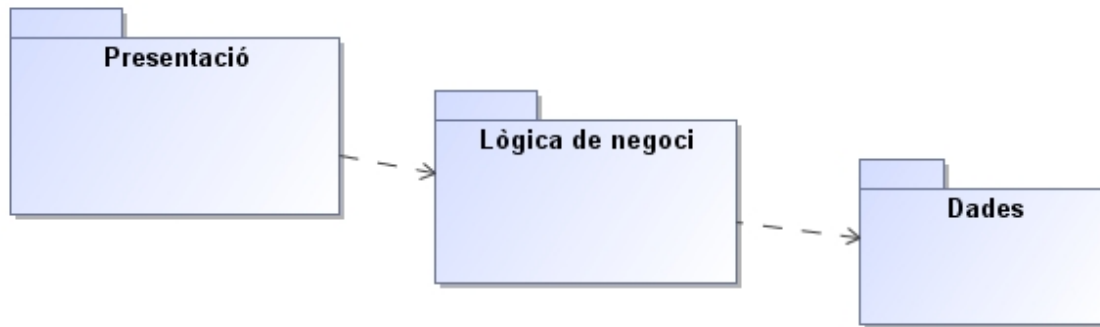
### Exemple d'un catàleg de lectura

Per simular lectura s'haurà de educar el sistema de la mateixa manera que els humans aprenem, aprenem associant fotos a un so i després aprenem a escriure-ho en format text. De manera anàloga, si al sistema li vam mostrar en un paper un símbol egipci o una síl·laba com "PA", capturarà una foto i ens demanarà un so que serà el seu nom; en tornar a ensenyar un paper de nou, si és el mateix el sistema reproduirà el seu nom, en el cas contrari, demanarà un nou so. Si tot això ho repetim diverses vegades aconseguim que el sistema aprengui a llegir el que ens interessi. En aquest procés es basarà en recursivitat i finalitzarà davant d'una ordre com "stop, sortir", o si el temps d'espera de sistema sobrepassa el límit preestablert.

## 4.1.Descripció de funcionament

---

FICAS es basarà en la programació per capes. Aquesta es refereix a un estil de programació que té com a objectiu separar la lògica de disseny de la lògica de negocis. Un dels avantatges que podem destacar sobre aquest estil és que el desenvolupament del programari es pot dur a terme en diversos tipus de nivells, així, quan succeeixi algun canvi només ens anirem sobre el nivell requerit.



II-lustració 18. Representació UML del model-vista-controlador.

Per definició, la programació per capes és una tècnica de l'enginyeria del programari pròpia de la programació a objectes, que es divideix en 3 capes: la capa de presentació o frontera, la capa de lògica de negoci i finalment la capa de dades.

#### - Capa de presentació

Es refereix a la presentació del programa davant l'usuari, aquesta presentació ha de complir el seu propòsit amb l'usuari final, una presentació fàcil d'usar i amigable. També les interfícies han de ser consistents amb la informació dins del programari (Per exemple, en els formularis no ha d'haver més que el necessari), prendre en compte els requeriments de l'usuari, la capa de presentació va de la mà amb capa de la lògica de negoci.

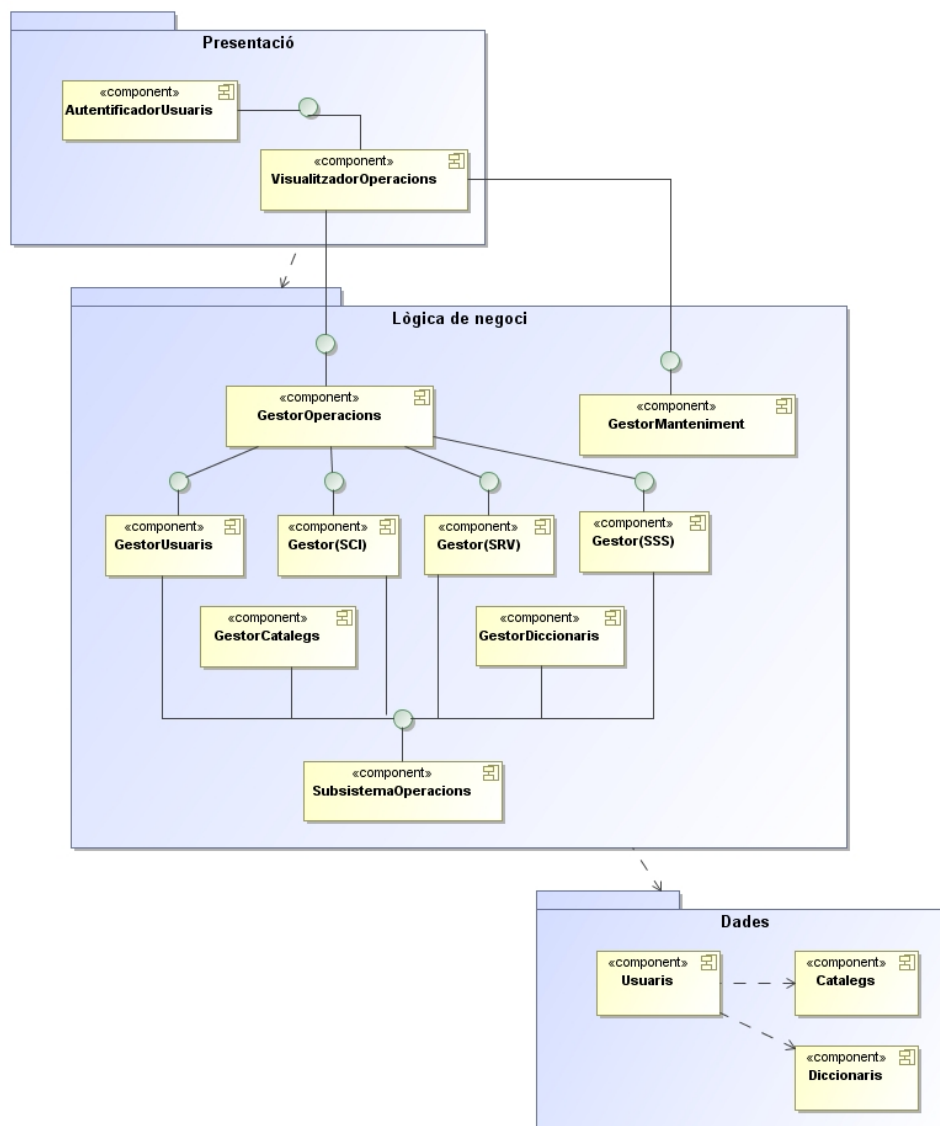
#### - Capa de lògica de negoci

En aquesta capa és on es troben els programes que són executats, rep les peticions de l'usuari i posteriorment envia les respostes després del procés. Aquesta capa és molt importants ja que és on s'estableixen totes aquelles regles que s'hauran de complir, deia anteriorment que la capa de presentació es comunica amb la capa de lògica de negoci ja que s'han de comunicar per rebre les sol·licituds i presentar els resultats.



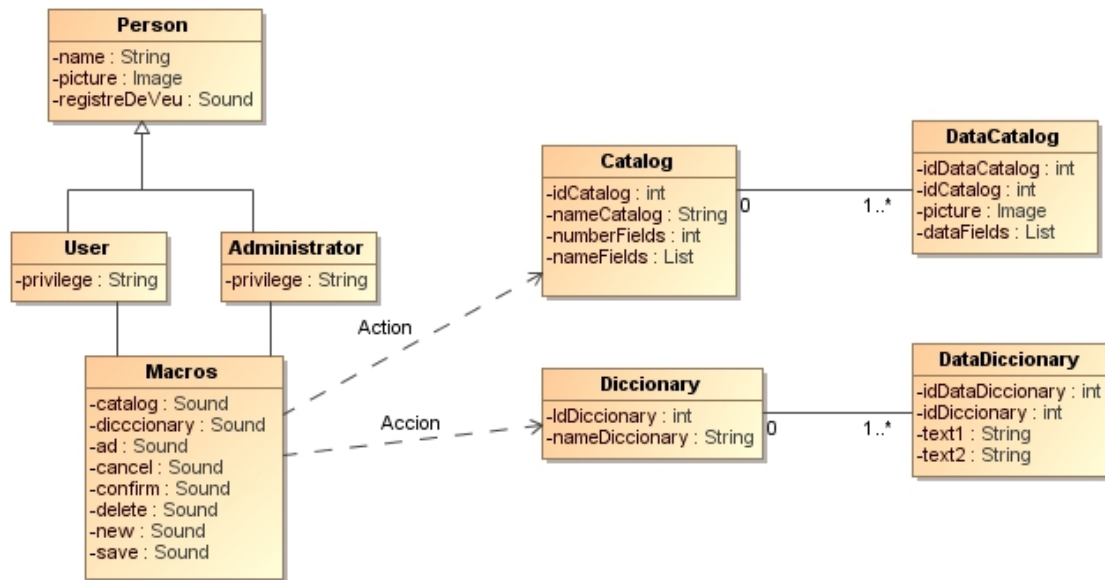
- Capa de dades

Aquesta capa és la que s'encarrega de fer les transaccions amb la base de dades i amb altres sistemes per descarregar o inserir informació al sistema. La consistència en les dades és summament important, és a dir, les dades que s'ingressen o s'insereixen han de ser precisos i conscients. Aquí definim les consultes que realitzarem a la base de dades, o consultes. La comunicació d'aquesta capa amb la capa de lògica de negoci es refereix al fet que la capa de dades és la que li enviés informació a la capa de negoci perquè sigui processada i ingressada en objectes segons sigui necessari (encapsulament).



II-lustració 19. Representació UML de la programació per capes i les respectives funcions.

### 4.1.1. Diagrama de classes



II-lustració 20. Diagrama de classes del projecte FICAS

### 4.1.2. Prototip d'Interface gràfica

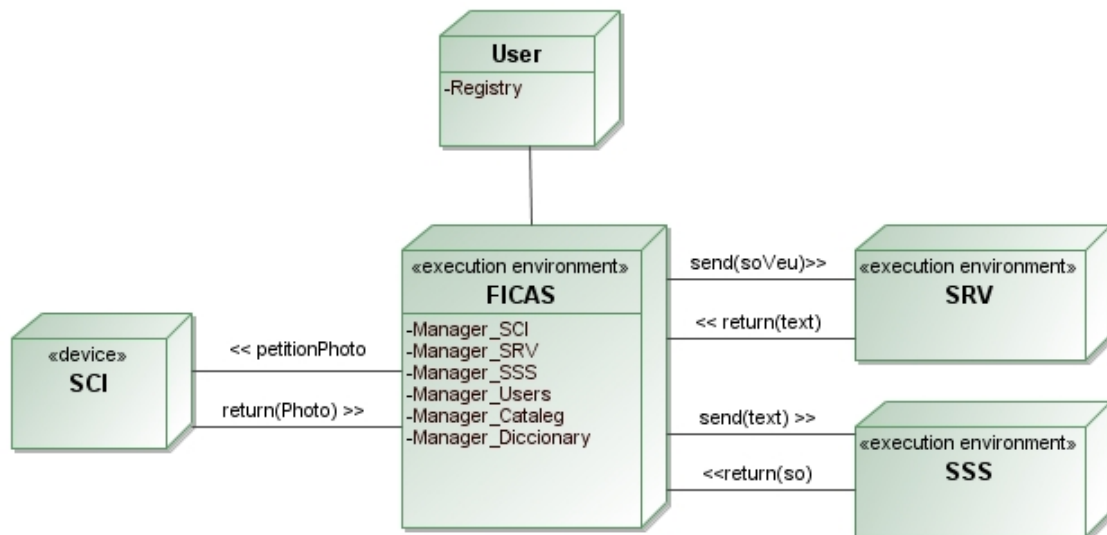
En aquesta pantalla apareixeran i desapareixeran, superposades, noves finestres d'entrada de vídeo i captura d'imatges, amb els corresponents missatges de confirmació.



II-lustració 21. Pantalla de presentació del projecte FICAS

### 4.1.3.Descripció d'integració

Per gestionar el catàlegs i els diccionaris, el sistema, farà crides als tres subsistemes, en qualsevol moment, per interactuar amb l'usuari; es a dir, que les sortides de text per pantalla es substituiran per sortides de sons de veu, i les entrades de text per pantalla es canviaran per entrades de veu.



Il·lustració 22. Esquema d'integració de sistemes del projecte FICAS

### 4.2.Casos d'ús

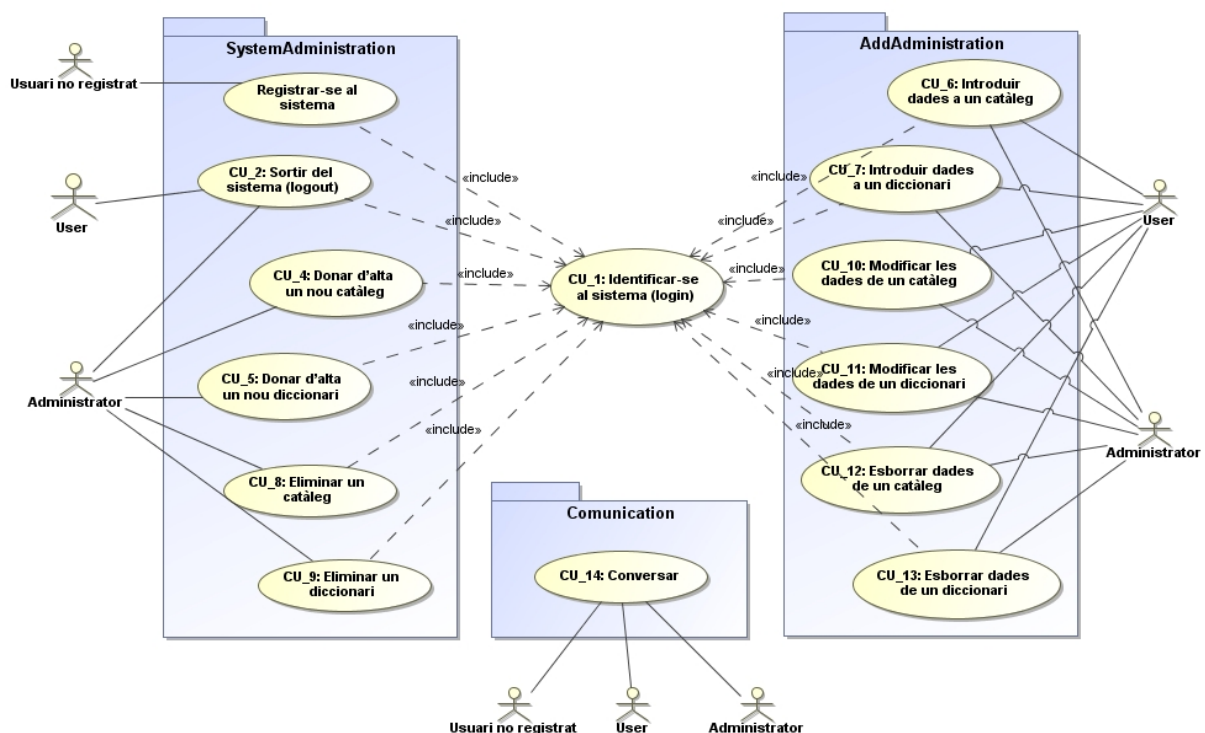
La capa de presentació s'ha d'implementar les diferents interfícies per a oferir les funcionalitats descrites anteriorment.

La capa de negoci s'ha d'implementar les funcionalitats descrites amb una arquitectura suficientment flexible perquè en un futur es puguin integrar noves funcionalitats i estendre les actuals.

La capa de dades gestionarà la informació de l'aplicació. S'ha de gestionar tota la informació necessària per a poder proporcionar les funcionalitats descrites anteriorment.

Per a cadascuna de les funcionalitats s'especifiquen els usuaris que fan ús d'elles: Usuari (U), Administrador (A) i Usuari no registrat.

- CU\_1: Identificar-se al sistema (login) (U, A)
- CU\_2: Sortir del sistema (logout) (U, A)
- CU\_3: Registrar-se al sistema (Usuari no registrat)
- CU\_4: Donar d'alta un nou catàleg (A)
- CU\_5: Donar d'alta un nou diccionari (A)
- CU\_6: Introduir dades a un catàleg (U, A)
- CU\_7: Introduir dades a un diccionari (U, A)
- CU\_8: Eliminar un catàleg (A)
- CU\_9: Eliminar un diccionari (A)
- CU\_10: Modificar les dades de un catàleg (U, A)
- CU\_11: Modificar les dades de un diccionari (U, A)
- CU\_12: Esborrar dades de un catàleg (U, A)
- CU\_13: Esborrar dades de un diccionari (U, A)
- CU\_14: Conversar (U, A, Usuari no registrat)



Il·lustració 23. Diagrama de casos d'ús de les funcionalitats del projecte.

#### Notes:

- L'agrupació proposada només és una de les infinites maneres d'agrupar les funcionalitats.
- S'han posat els actors diverses vegades al diagrama per llegibilitat.

A continuació, es descriu els casos d'ús, més rellevants, referents a la creació d'un nou catàleg, d'un nou diccionari i de com es realitza una conversa.

### Cas d'us nou catàleg

<i>Resum de funcionalitat:</i>	Permet definir i registrar, els catàleg al sistema.
<i>Actors:</i>	Admin. (privilegis de creació, eliminació, modificació i consulta)
<i>Casos d'ús relacionats:</i>	Introduir de dades a un catàleg, Gestió de reconeixement de veu i Gestió síntesis de so
<i>Precondició:</i>	L'usuari es vàlid i s'ha identificat correctament en el sistema
<i>Postcondició:</i>	Les accions s'ha realitzat correctament, en el cas de ser l'usuari Admin les modificacions/altes/baixes han quedat enregistrades en la base de dades.
<i>Procés normal principal:</i>	<ol style="list-style-type: none"> <li>1. El sistema espera que l'usuari doni una ordre de veu</li> <li>2. L'usuari, és Admin, i pronuncia a paraula "catàleg" <ol style="list-style-type: none"> <li>2.1. El sistema li dema el nom del catàleg (amb veu)</li> <li>2.2. L'usuari és Admin, pronuncia el nom del catàleg</li> </ol> </li> <li>3. El sistema verifica si el nom està guardat al registre <ol style="list-style-type: none"> <li>3.1. En cas que sigui negatiu, el sistema demanarà el nombre de camp que tindrà el nou catàleg (amb veu)</li> <li>3.2. L'usuari pronunciarà el número de camps.</li> <li>3.3. El sistema, per cada camp, demanarà el seu nom (amb veu)</li> <li>3.4. L'usuari, per cada camp, pronunciarà el nom del camp</li> </ol> </li> <li>4. Un cop finalitzat, el sistema mostrarà les dades i esperarà conformitat</li> <li>5. L'usuari, és Admin, confirmarà que les dades introduïdes siguin correctes i donarà la conformitat (amb veu)</li> <li>6. El sistema guardarà totes les dades al registre i tornarà al</li> </ol>

	punt 1
<i>Alternatives de procés i excepcions:</i>	<p>3.1 En cas que sigui afirmatiu, el sistema passarà al cas d'ús d'introduir de dades a un catàleg</p> <p>5.2 En cas contrari el sistema torna al punt 1</p>

Taula 17. Descripció del cas d'ús de crear un nou catàleg.

### Cas d'us nou diccionari

<i>Resum de funcionalitat:</i>	Permet definir i registrar, els diccionaris al sistema.
<i>Actors:</i>	Admin. (privilegis de creació, eliminació, modificació i consulta)
<i>Casos d'ús relacionats:</i>	Introduir de dades a un diccionari, Gestió de reconeixement de veu i Gestió síntesis de so
<i>Precondició:</i>	L'usuari es vàlid i s'ha identificat correctament en el sistema
<i>Postcondició:</i>	Les accions s'ha realitzat correctament, en el cas de ser l'usuari Admin les modificacions/altes/baixes han quedat enregistrades en la base de dades.
<i>Procés normal principal:</i>	<ol style="list-style-type: none"> <li>1. El sistema espera que l'usuari doni una ordre de veu</li> <li>2. L'usuari, és Admin, i pronuncia a paraula "diccionari" <ol style="list-style-type: none"> <li>2.1. El sistema li dema el nom del diccionari (amb veu)</li> <li>2.2. L'usuari és Admin, pronuncia el nom del diccionari</li> </ol> </li> <li>3. El sistema verifica si el nom està guardat al registre <ol style="list-style-type: none"> <li>3.1. En cas que sigui negatiu, el sistema demanarà el nom de camp número 1 (amb veu)</li> <li>3.2. L'usuari pronunciarà el nom del camp 1</li> <li>3.3. El sistema demanarà el nom de camp número 2 (amb veu)</li> <li>3.4. L'usuari pronunciarà el nom del camp 2</li> </ol> </li> <li>4. Un cop finalitzat, el sistema mostrarà les dades i esperarà conformitat</li> <li>5. L'usuari, és Admin, confirmarà que les dades introduïdes</li> </ol>

	<p>siguin correctes i donarà la conformitat (amb veu)</p> <p>6. El sistema guardarà totes les dades al registre i tornarà al punt 1</p>
<i>Alternatives de procés i excepcions:</i>	<p>3.1.1 En cas que sigui afirmatiu, el sistema passarà al cas d'ús d'introduir de dades a un diccionari</p> <p>5.1 En cas contrari, el sistema torna al punt 1</p>

Taula 18. Descripció del cas d'ús de crear un nou diccionari

### Cas d'us conversar

<i>Resum de funcionalitat:</i>	Permet definir i registrar, els diccionaris al sistema.
<i>Actors:</i>	Admin, usuari registrats i usuaris no registrats
<i>Casos d'ús relacionats:</i>	Gestió de reconeixement de veu i Gestió síntesis de so
<i>Precondició:</i>	Tenen que haver dades registrades, con a mínim en un diccionari
<i>Postcondició:</i>	
<i>Procés normal principal:</i>	<ol style="list-style-type: none"> <li>1. El sistema espera que l'usuari faci una pregunta</li> <li>2. L'usuari pronuncia una frase</li> <li>3. El sistema cerca en els diferents diccionaris la frase <ol style="list-style-type: none"> <li>3.1. En cas que sigui negatiu, i el usuari es admin, el sistema demanarà la resposta (amb veu)</li> <li>3.2. L'usuari pronunciarà la resposta</li> </ol> </li> <li>4. Un cop finalitzat, el sistema mostrarà les dades i esperarà conformitat</li> <li>5. L'usuari, és Admin, confirmarà que les dades introduïdes siguin correctes i donarà la conformitat (amb veu)</li> <li>6. El sistema guardarà totes les dades al registre i tornarà al punt 1</li> </ol>
<i>Alternatives de procés i excepcions:</i>	3.1.1. En cas afirmatiu, el sistema retornarà la resposta (amb veu) i torna al punt 1

	<p>3.1.2. En cas negatiu, i el usuari no es admin, el sistema indicarà que no te la resposta, i torna al punt 1</p> <p>5.1 En cas que sigui afirmatiu, el sistema passarà al cas d'ús d'introduir de dades a un diccionari</p> <p>5.2 En cas contrari el sistema torna al punt 1</p>
--	--

Taula 19. Descripció del cas d'ús de conversar

### Cas d'us usuaris

<i>Resum de funcionalitat:</i>	Permet registrar els usuaris al sistema.
<i>Actors:</i>	Admin, usuari registrats i usuaris no registrats
<i>Casos d'ús relacionats:</i>	Gestió de reconeixement de veu i Gestió síntesis de so
<i>Precondició:</i>	
<i>Postcondició:</i>	Les accions s'ha realitzat correctament, en el cas de ser l'usuari Admin les modificacions/altes/baixes han quedat enregistrades en la base de dades
<i>Procés normal principal:</i>	<ol style="list-style-type: none"> <li>1. El sistema espera que l'usuari estigui davant del sistema</li> <li>2. L'usuari permet que el sistema capti la seva cara</li> <li>3. El sistema, quan detecta que hi ha una persona al davant, capta les faccions de la cara, i cerca si hi es al registre.             <ol style="list-style-type: none"> <li>3.1. En cas que sigui negatiu, el sistema demanarà si és vol registrar com a admin, user o no es vol registrar (amb veu)                 <ol style="list-style-type: none"> <li>3.1.1. Si l'usuari es vol registrar com a admin                     <ol style="list-style-type: none"> <li>3.1.1.1. El sistema demanarà el nom i una frase que serà la contrasenya (amb veu)</li> <li>3.1.1.2. L'usuari admin, confirmarà la captura de la foto, pronunciarà el seu nom i pronunciarà una frase.</li> </ol> </li> <li>3.1.2. Si l'usuari es vol registrar com a user                     <ol style="list-style-type: none"> <li>3.1.2.1. El sistema demanarà el nom (amb veu)</li> <li>3.1.2.2. L'usuari, no admin, confirmarà la captura</li> </ol> </li> </ol> </li> </ol> </li> </ol>

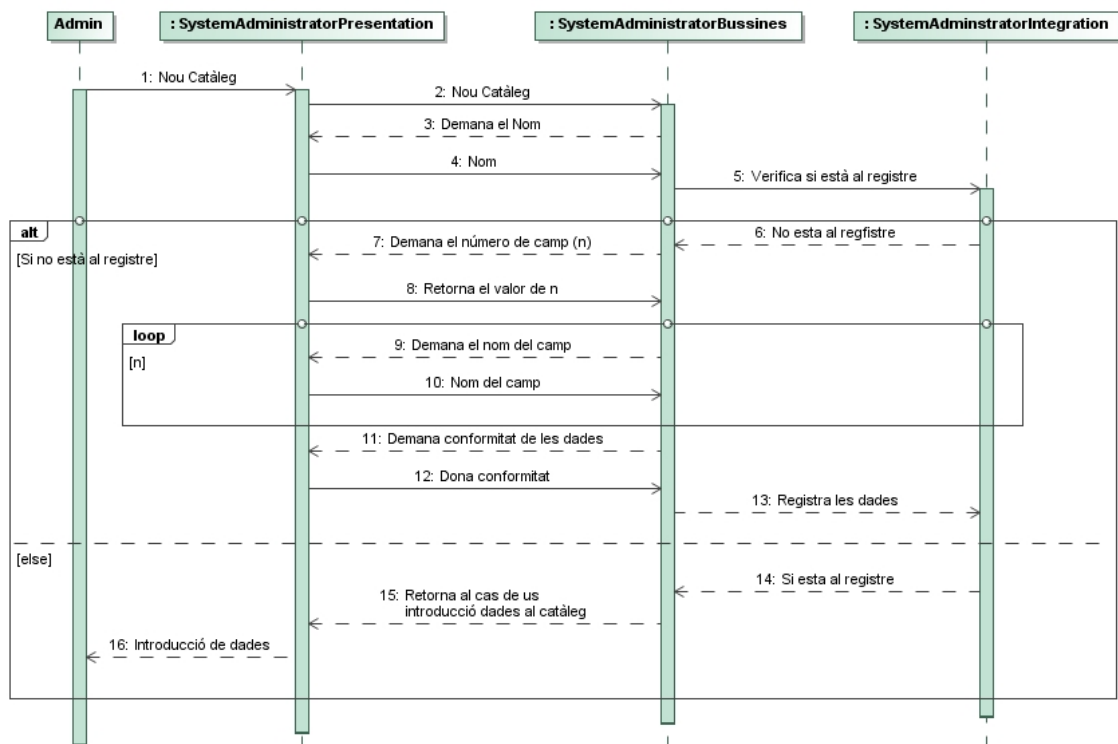


	<p>de la foto, pronunciarà el seu nom.</p> <p>3.1.2.3. El sistema demanarà conformitat al admin</p> <p>3.2. En el cas que estigui registrat, el sistema permetrà al usuari el accés sistema.</p> <p>4. Un cop finalitzat, el sistema mostrarà les dades i esperarà conformitat</p> <p>5. L'usuari, és Admin, confirmarà que les dades introduïdes siguin correctes i pronunciarà la contrasenya (amb veu)</p> <p>6. El sistema guardarà totes les dades al registre i permetrà l'accés al sistema</p>
<i>Alternatives de procés i excepcions:</i>	<p>3.2.1 En cas contrari el sistema passarà al cas d'ús conversar</p> <p>5.1. En cas contrari el sistema torna al punt 1</p>

Taula 20. Descripció del cas d'ús de registre d'usuaris

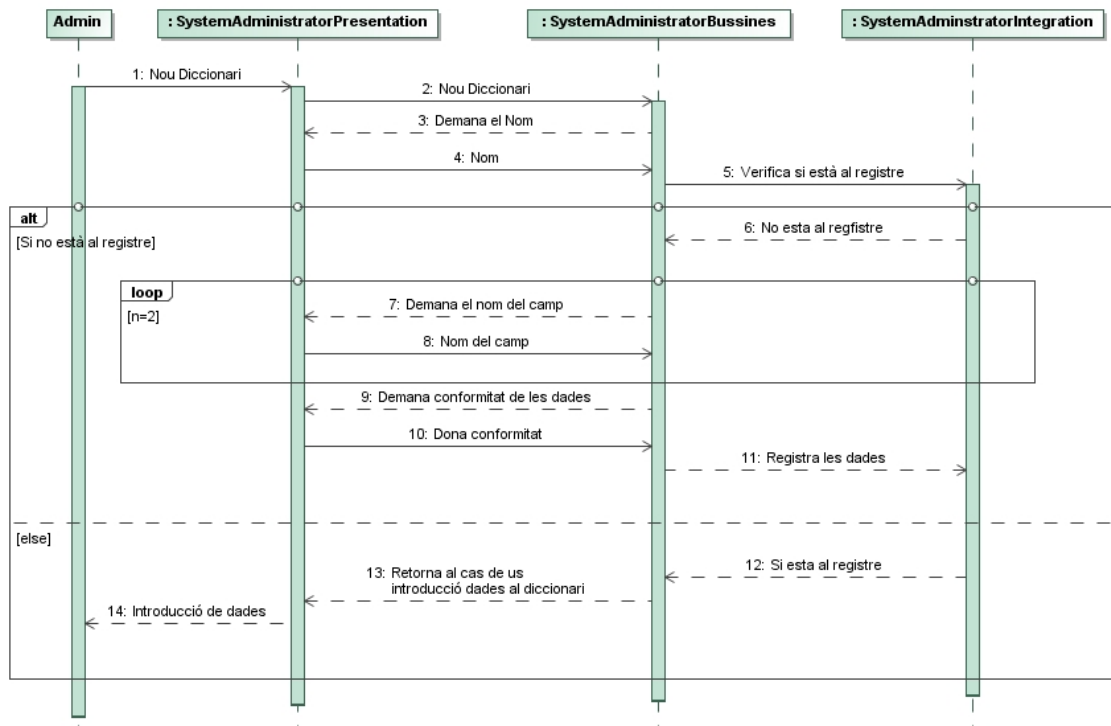
### 4.3. Diagrames de seqüència

#### Cas d'us nou catàleg



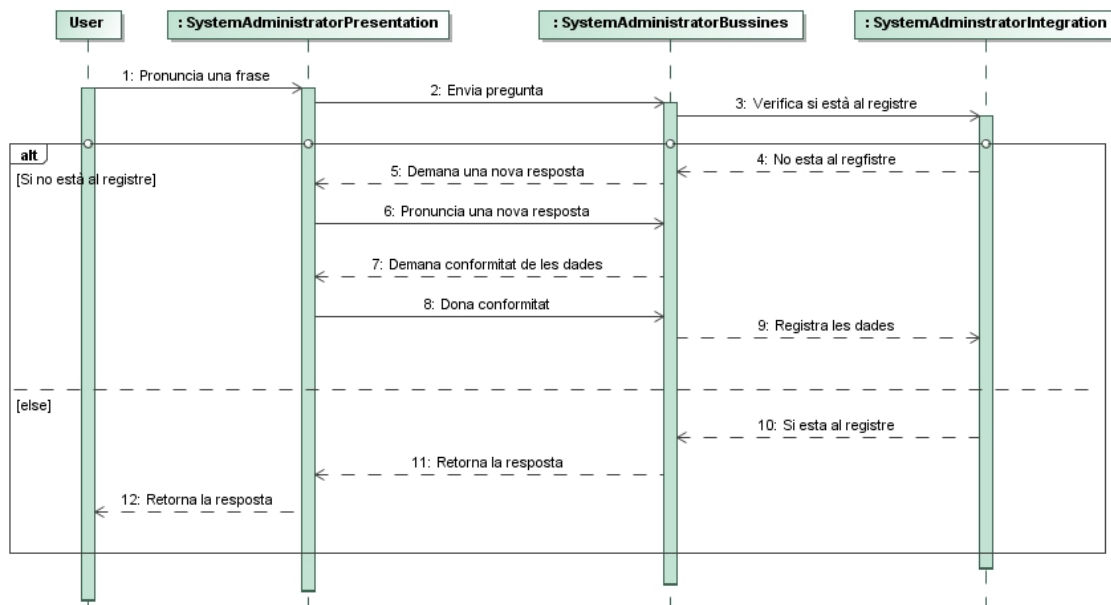
Il·lustració 24. Diagrama de seqüència per crear un nou catàleg.

### Cas d'us nou diccionari



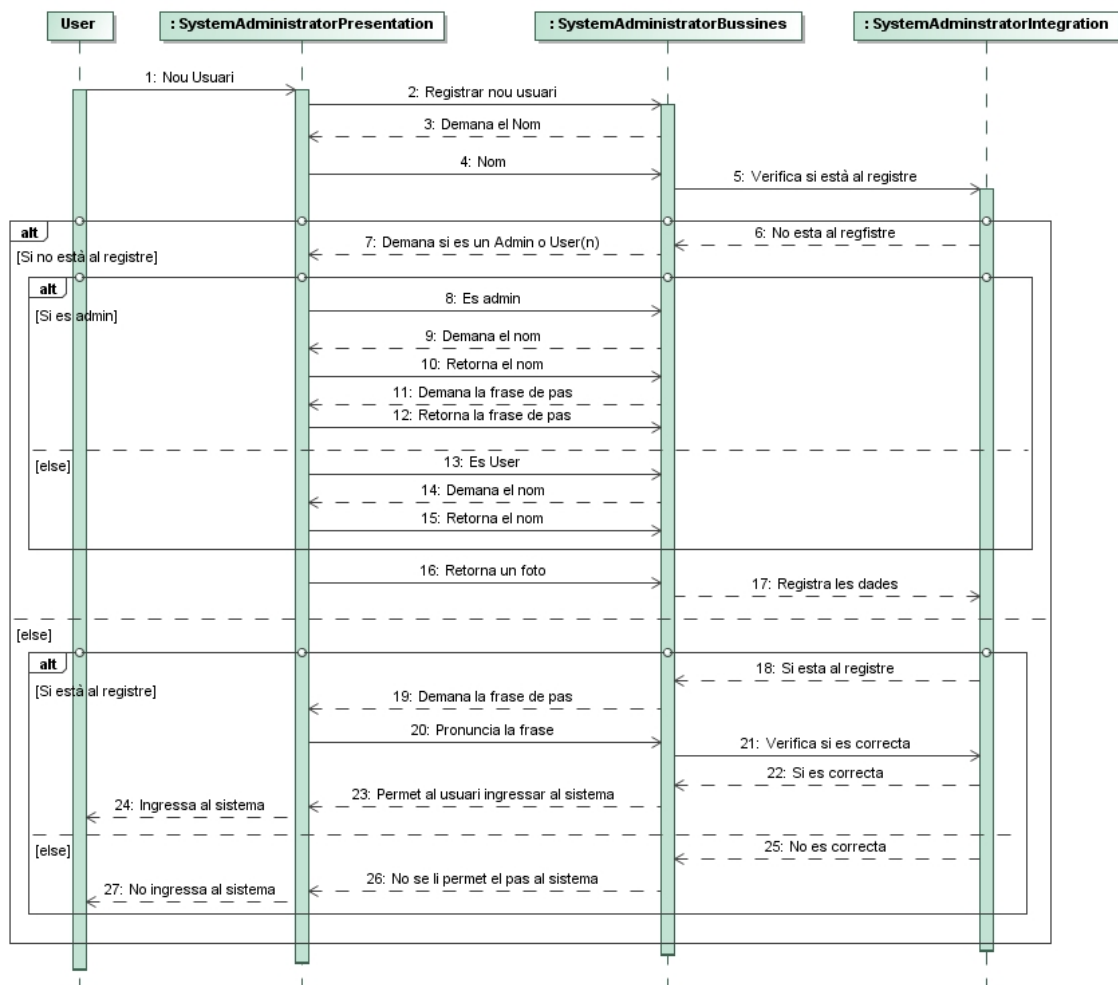
II-Il·lustració 25. Diagrama de seqüència per crear un nou diccionari

### Cas d'us conversar



II-Il·lustració 26. Diagrama de seqüència del cas d'ús conversar (admin i usuari registrat)

Cas d'ús usuaris



Il·lustració 27. Diagrama de seqüència del cas d'ús usuaris

## 5. Implementació

---

Per implementar FICAS, es pren com intèrpret, l'anterior esmentat, Python<sup>21</sup> per les seves qualitats que s'ha descrit al llarg d'aquesta memòria, i després d'un laboriós estudi d'integració de sistemes d'AI, serà necessària la utilització de diverses biblioteques que es defineixen a continuació:

- Per al sistema de captura d'imatges (SCI)
  - **OpenCV:** és una biblioteca lliure de visió artificialment originalment desenvolupada per. Des que va aparèixer la seva primera versió alfa en el mes de gener de 1999, s'ha utilitzat en infinitat d'aplicacions. Des de sistemes de seguretat amb detecció de moviment, fins a aplicacions de control de processos on es requereix reconeixement d'objectes. Això es deu a que la seva publicació es dona sota llicència BSD, que permet que sigui usada lliurement per a propòsits comercials i d'investigació amb les condicions que s'hi expressen.  
<https://sourceforge.net/projects/opencvlibrary/files/>
  - **Numpy:** És una extensió de Python, que agrega major suport per vectors i matrius, constituint una biblioteca de funcions matemàtiques de gran nivell per operar amb aquests vectors o matrius. L'ancestre de NumPy, va ser creat originalment per Jim Hugunin amb algunes contribucions d'altres desenvolupadors. El 2005, Travis Oliphant va crear NumPy incorporant característiques de Numarray en NumPy amb algunes modificacions. NumPy és de codi obert.  
<http://www.numpy.org/>
  - **Matplotlib:** és una biblioteca per a la generació de gràfics a partir de dades continguts en llistes o matrius en el llenguatge de

---

<sup>21</sup> <https://www.python.org/>

programació Python i la seva extensió matemàtica NumPy. Proporciona una API, pylab, dissenyada per recordar a la de MATLAB.

<http://matplotlib.org/>

- **Scipy:** és una biblioteca de codi obert de les eines i algorismes matemàtics per a Python que es va crear a partir de la col·lecció original de Travis Oliphant<sup>22</sup> que consistia en mòduls d'extensió per a Python, llançada en 1999 sota el nom de Multipack (cridada pels paquets netlib que reuneixen a ODEPACK, QUADPACK, i MINPACK).

<https://www.scipy.org/>

- Per al sistema de reconeixement de veu (SRV)
  - **Speech\_recognition:** Un sistema de reconeixement de veu és una eina computacional capaç de processar el senyal de veu emès pel ser humà i reconèixer la informació continguda en aquesta, convertint-la en text o emetent ordres que actuen sobre un procés. En el seu desenvolupament intervien diverses disciplines, com ara: la fisiologia, l'acústica, la lingüística, el processament de senyals, la intel·ligència artificial i la ciència de la computació.

<https://pypi.python.org/pypi/SpeechRecognition>

- Per al sistema de síntesi de so (SSS)
  - **Pyaudio:** proporciona enllaços de Python per PortAudio v19, la biblioteca d'E / S d'àudio multi plataforma. Amb PyAudio, es pot utilitzar Python fàcilment per reproduir i gravar seqüències d'àudio en una varietat de plataformes (per exemple, GNU / Linux, Microsoft Windows i Mac OS X).

<https://pypi.python.org/pypi/PyAudio/>

---

<sup>22</sup> Travis Oliphant és un científic i empresari americà de dades. És el CEO i fundador de l'inici de la tecnologia Continuum Analytics. A més, és el principal creador del NumPy i contribuent fundador als paquets SciPy en els llenguatges de programació Python.

El principal problema per realitzar l'estudi viabilitat d'integració de sistemes, ha estat que cadascuna, d'aquestes aplicacions, per separat està pensada per a una versió de Python diferent; per tant, després de diverses proves la millor versió, i més estable, que s'ha trobat és la 3.5.4, que està disponible a: <https://pypi.python.org/pypi/cppo/3.5.4>

## 5.1. Funcionalitat de FICAS

---

- Per començar FICAS verificarà que tots els sistemes funcionen, en cas contrari no podrà continuar. Ho farà de la següent manera:

**Verificació de sistemes**

```
image_path = "data/system/Ficas_inicial.jpg"
image = cv2.imread(image_path)
cv2.imshow("Projecte Ficas", image)
cv2.waitKey(10)

try:
    # Webcam
    video_capture = cv2.VideoCapture(webcam_id)
    ____, frame = video_capture.read()
    a = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    print("El sistema de captura de fotos funciona bien")
except:
    print("Lo siento! El sistema de captura de fotos no funciona bien")
    print('Ponte en contacto con el administrador del sistema')
    print('No puedo continuar hasta que no se repare la camara.')
    Pulsa_Tecla()
    sys.exit()

try:
    b = sr.Recognizer()
    c = sr.Microphone()
    SSS("Un momento de silencio, por favor...")
    with c as source: b.adjust_for_ambient_noise(source)
    SSS("Establecer el umbral mínimo de energía para el microfono:
".format(b.energy_threshold))
except:
    print("Lo siento! El microfono no funciona bien")
    print('Ponte en contacto con el administrador del sistema')
    print('No puedo continuar hasta que no se repare el microfono.')
    Pulsa_Tecla()
    sys.exit()

try:
    SSS("Todos los sistemas funcionan bien")
except:
    Habla = 'No'
    print("Lo siento!!!!!!!!!!!!!!!!!!!!!!!!!!!!")
    print("Hay problemas con el sistema de sintesis de voz")
    print("No se reconoce el lenguaje. Lo he desactivado")
    print("Debes de leer la pantalla")
```

Taula 21. Verificació inicial de funcionament de sistemes

- El funcionament FICAS es basarà en una verificació inicial dels usuaris registrats en el sistema, per mitjà d'un reconeixement facial: si s'està registrat en el sistema, es pots accedir a totes les funcions del sistema: gestionar catàlegs, usuaris, diccionaris i fotos; però, si no ets acceptat pel sistema, només pots realitzar converses. Es pot apreciar amb el següent codi.

Menú de verificació d'usuaris
<pre>def main():     if Administrador == 'admin' or Administrador == 'user':         SSS('Estás dentro del sistema Ficas. Un momento por favor! se están cargando los datos.')         Usuarios.loadUsuarios('')         Catalogs.loadCatalogs('')         Dictionarys.loadDictionary('')         SSS('Hola, que quieres hacer?')         SSS('Las opciones son: Usuarios, Fotos, Catálogos y Diccionarios')         menu()         main()      SSS('Lo siento!, no eres un usuario registrado')     SSS('Pero, si quieres podemos conversar.')     GC()</pre>

Taula 22. Menú de verificació d'usuaris per registrar-se en FICAS

- Un cop dins FICAS es pot accedir als diferents gestors (Catàlegs, diccionaris, fotos i usuaris) amb una ordre de veu, que es deu pronunciar en el moment que s'indiqui.

En el menú principal, el sistema demanarà una ordre de veu, que s'ha de comparar amb les entrades del fitxer de comandes; si hi es retornarà un clau, que accionarà l'entrada en el corresponent subsistema gestor; però, si la comanda no existeix i després de 5 intents el sistema acabarà i es sortirà del sistema, com es veu en la següent taula.



## Menú principal

```
def menu():
    show_foto("Projecte Ficas", "data/system", "Ficas_menu")
    cv2.waitKey(10)
    Intento=5
    while Intento > 0:
        clave = str(comando())
        #clave = 'SGC'
        Intento = Intento - 1
        print('Intento: '+str(Intento))
        if (clave == 'SGU'):
            SSS('Estás en el sistema de registro de usuarios')
            SGU()
            menu()
        if (clave == 'SGF'):
            SSS('Estás en el sistema de captura de fotos')
            SGF()
            menu()
        if (clave == 'SGC'):
            SSS('Estás en el sistema gestor de catalogos')
            SGC()
            menu()
        if (clave == 'SGD'):
            SSS('Estás en el sistema gestor de diccionarios')
            SGD()
            menu()
        if (clave == '_cancel_'):
            SSS('Adios, espero que te haya gustado')
            sys.exit()
    return
```

Taula 23. Menú principal de FICAS per accedir al diferents sistemes gestors.

- Les claus, del fitxer d'ordres, per interactuar amb el diferents subsistemes són els següents:

Fitxer de claus
<pre>!admin=SGU !user=SGU !diccionario=SGD !dictionary=SGD !catálogo=SGC !catálogos=SGC !cataleg=SGC !imagen=SGF !foto=SGF !hola=SGU !usuario=SGU !conversar=_conversar_ !nuevo=_new_ !guardar=_save_ !cancelar=_cancel_ !stop=_cancel_ !salir =_cancel_ !confirmar=_ok_ !okay=_ok_ !borrar=_delete_</pre>

Taula 24. Les ordres per interactuar amb els diferents subsistemes.

- Es pot apreciar que al pronunciar diferents paraules es converteix amb el mateix comando, per facilitar el accés al sistema; e inclòs es poden donar les ordres amb qualsevol llengua, simplement modificat les claus.

Eixample: L'ordre en espanyol: diccionario=SGD, es el mateix que en anglès: dictionary=SGD

- Un altre aspecte molt important, pel que fa a la funcionalitat, és la codificació dels arxius, perquè es puguin guardar i recuperar dades amb

accents i caràcters especials. El problema radica que 'open()' espera una cadena de tipus ASCII, de manera que s'ha d'utilitzar altres codificacions com UTF-8. En 'codecs.open()' s'afegeix un tercer argument en el qual pot especificar la codificació.

### Format d'escriptura amb fitxer

```
import codecs

f = codecs.open("archivo.txt", "w", "utf-8")
f.write("Esto es una línea con acentos.")
f.close()
```

Taula 25. Codi per escriure als fitxers amb codificació UTF-8

## 5.2. Joc de proves

---

Per verificar que FICAS funciona correctament s'ha fet el següent joc de proves:

```
Joc de proves

##### Usuarios #####
##### Text de pruebas #####

##### Usuarios #####
usuario = 'Paco'
tipoUsuario = 'admin'
passwordUsuario = 'boltaña'
Usuarios.addUsuarios('', usuario, tipoUsuario, passwordUsuario)

usuario = 'Eva'
tipoUsuario = 'user'
passwordUsuario = 'sieste'
Usuarios.addUsuarios('', usuario, tipoUsuario, passwordUsuario)

##### Catalogos #####
print('## Catalogos ')
nameCatalog = "Animales"
numFields = 3
nameFields = ('Clase', 'Subclase', 'Alimentacion')
Catalogs.addCatalogs('', nameCatalog, numFields, *nameFields)

##### DataCatalogo #####
print('## DataCatalogo')
pictureCatalog = "Tigre"
dataFiledS = ('Vertebrados', 'Mamiferos', 'Carnivoro')
DataCatalog.addDataCatalog('', nameCatalog, pictureCatalog, *dataFiledS)

pictureCatalog = "Elefante"
dataFiledS = ('Vertebrados', 'Mamiferos', 'Hervivoro')
DataCatalog.addDataCatalog('', nameCatalog, pictureCatalog, *dataFiledS)
```

```
##### Catalogos #####
print('## Catalogos')
nameCatalog = "Insectos"
numFields = 3
nameFields = ('Proteccion coporal', 'Clase', 'Subclase')
Catalogs.addCatalogs('', nameCatalog, numFields, *nameFields)
##### DataCatalogo #####
print('## DataCatalogo')
pictureCatalog = "Mosca"
dataFiledS = ('Con Proteccion', 'Atropodos', 'Insecto')
DataCatalog.addDataCatalog('', nameCatalog, pictureCatalog, *dataFiledS)

print('_---Lista de Usuarios---_')
Usuarios.loadUsuarios('')
Usuarios.listUsuarios('')
print('_---Lista de Catalogos---_')
Catalogs.loadCatalogs('')
Catalogs.listCatalogs('')
print('_---Lista de Datos del catalogo----')
DataCatalog.loadDataCatalog('', 'Animales')
DataCatalog.listDataCatalog('', 'Animales')

print('-----Busquedas-----')
Catalogs.SearchCatalogs('', 'Animales')

print('-----Busquedas-----')
DataCatalog.SearchDataCatalog('', 'Elefante')
print('-----Busquedas-----')
Usuarios.SearchUsuarios('', 'Paco')
Usuarios.SearchTipoUsuario('', 'user')

nameDictionary="Ingles"
Dictionarys.loadDictionary('')
Dictionarys.addDictionary('', nameDictionary)
Dictionarys.loadDataDictionary(nameDictionary)
```

Taula 26. Joc de proves per verificar la funcionalitat del projecte.

Com a resultat obtenim el següent llistat de dades, per la consola.

### Resultat del joc de proves

```
El sistema de captura de fotos funciona bien
Todos los sistemas funcionan bien
[('Paco', 'admin', 'boltaña')]
Se han introducidos los datos del usuario Paco
[('Paco', 'admin', 'boltaña'), ('Eva', 'user', 'sieste')]
Se han introducidos los datos del usuario Eva
## Catalogos
Se han introducidos los campos del catálogo Animales
[('Animales', 3, 'Clase', 'Subclase', 'Alimentacion')]
## DataCatalogo

El catálogo Animales, ya esta registrado en el sistema
[]
## Catalogos
Se han introducidos los campos del catálogo Insectos
[('Animales', 3, 'Clase', 'Subclase', 'Alimentacion'), ('Insectos', 3,
  'Proteccion coporal', 'Clase', 'Subclase')]
## DataCatalogo

El catálogo Insectos, ya esta registrado en el sistema
_---Lista de Usuarios---_
Se están cargando los datos de los usuarios
Nombre: Paco, Función: admin, Contraseña: jonas
Nombre: Eva, Función: user, Contraseña: sieste

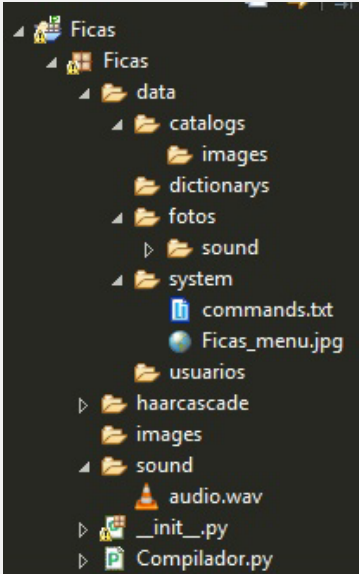
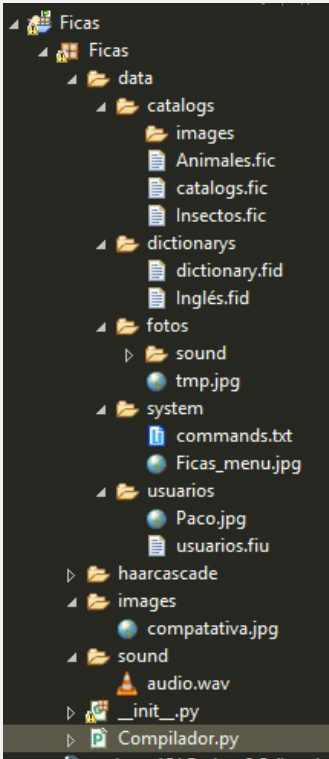
_---Lista de Catalogos---_
Se están cargando los catálogos
Nombre del catalogo: Animales, Número de campos: 3, Nombre de los
campos: Clase,Subclase,Alimentacion
Nombre del catalogo: Insectos, Número de campos: 3, Nombre de los
campos: Proteccion coporal,Clase,Subclase

_---Lista de Datos del catalogo----_
Se están cargando los datos del catálogo: Animales
Error, El archivo no existe.
```

```

-----Busquedas-----
Nombre del catalogo: Insectos, Número de campos: 3, Nombre de los
campos: Proteccion coporal,Clase,Subclase
-----Busquedas-----
-----Busquedas-----
Nombre: Paco, Función: admin, Contraseña: boltaña
Nombre: Eva, Función: user, Contraseña: sieste
    
```

Taula 27. Resultat del joc de proves del projecte.

Resultat després d'aplicar el joc de proves	
Abans	Després
	

Taula 28. Diferencia del directori després d'aplicar el joc de proves.

## 6. Conclusions

---

Al llarg del desenvolupament d'aquesta memòria, que té com a objectiu: l'estudi de viabilitat d'un prototip d'integració de sistemes d'intel·ligència artificial, s'ha pogut constatar que els objectius s'han complert, que el projecte és viable i que, per descomptat, té moltes possibles millores que es descriuen a continuació. És a dir, que el que en un principi es pensava era un plantejament molt ambiciós i que semblava un galimaties per la seva complexitat, perquè cada un d'aquests sistemes per separat ja podria ser un projecte individual, s'ha convertit en una realitat. A més, quan més s'aprofundeix en el seu desenvolupament, més t'atrau, i menys t'importa el temps invertit, perquè es descobreix un gran ventall de millores i possibilitats.

### 6.1. Objectius complits

---

L'objectiu inicial de FICAS és la interacció entre els tres sistemes d'IA (Captura d'imatges, reconeixement de la parla i síntesi de veu), per simular l'aprenentatge humà. Així mateix, en l'apartat 1.3 Objectius específics, es va definir un exemple de lectura com a principal idea de funcionament del projecte, que s'ha aconseguit amb el desenvolupament del que s'ha definit com a 'Capturar imatges'; però no només això, sinó que a més les podem agrupar en catàlegs, amb una longitud de camps que pot ser variable; també, s'ha creat un registre d'usuaris, per entrar-hi; al mateix temps, s'ha aconseguit la possibilitat de crear diccionaris; i el que és força interessant i que posseeix un gran interès, és la conversa, que de manera senzilla simularà com es comuniquen les persones amb la veu.



## 6.2. Viabilitat

---

El projecte FICAS, que partia com un estudi de viabilitat d'un prototip d'IA, ha deixat de ser un estudi i ha passat a ser una realitat, ha passat de ser un prototip a ser un projecte viable.

Però, la primera gran avantatge és la gran varietat de configuracions que es poden desenvolupar, depenent de les necessitats i del suport d'implementació: per a un sistema robòtic serà necessari el sistema de gestió de fotos (SGF) i el sistema de gestió de diccionaris (SGF) i la conversa; per domòtica, el sistema de reconeixement de la parla (SRV) i el sistema síntesi de veu (SSS) que executi ordres de veu com: obrir les finestres, connectar la climatització, posar música, despertar-nos per la matins; en un vehicle (perquè no, espacial) amb els sistemes SRV, SSS i potenciant un sistema de reconeixement d'objectes, perquè ens de les dades que necessitem, que ens informe dels paràmetres que sobrepassen límits preestablerts i que adverteixi de certs perills; i per descomptat, en qualsevol àmbit de la discapacitat que abasta la fundació ONCE, un exemple podria ser: que un venedor de l'ONCE, amb només apropar una butlleta a una càmera web, el sistema pronuncii els números i digui si està premiat.

El segon avantatge, és que el suport d'instal·lació pot ser molt variat: des d'una simple Raspberry Pi<sup>23</sup>, incorporant un micròfon i una sortida de so; fins a qualsevol tipus d'ordinador personal (Mac o Pc); passant per sistemes robòtics que necessitin acatar ordres per veu.

Amb tot l'anterior, es pot dir que FICAS és viable i molt versàtil, que el converteix en un projecte que pot tenir una gran cabuda en el mercat i una gran repercussió en la societat per aquesta petita revolució que suposa la interacció amb sistema informàtics per mitjà de la veu.

---

<sup>23</sup> *Raspberry Pi és un computador de placa reduïda, ordinador de placa única o computador de placa simple (SBC) de baix cost desenvolupat al Regne Unit per la Fundació Raspberry Pi, amb l'objectiu d'estimular l'ensenyament de ciències de la computació a les escoles.*

### 6.3. Futures millores

A simple vista, una de possibles millores és desenvolupar FICAS amb una base de dades, però la idea inicial, és que tots els diccionaris i tots els catàlegs es puguin intercanviar entre diferents sistemes informàtics, motiu pel qual, no seria necessari gestionar els dades amb una base de dades, ja que el convertiria en un sistema menys escalable. Tot l'anterior, obligaria a codificar aquests fitxers d'intercanvi amb petit algoritme de protecció intern i exclusiu dissenyat per FICAS.

Potenciar el sistema de captura de fotos perquè pugui reconèixer objectes, a partir de formes, mides i patrons preestablerts.

Millorar el sistema de comandaments de veu, que actualment és d'una sola paraula, amb diverses paraules en una mateixa ordre (es dirà Macro), per exemple: 'Eborrar tigre a catàleg animals'.

Ordre de veu				
Eborrar	tigre	a	Catàleg	animals
Es una acció	Es una clau d'entrada	No es útil	Pot ser Catàlegs o Diccionaris	El nom del catàleg que conté la clau 'tigre'

Taula 29. Sistema de funcionament de una Macro que te varies ordres.

D'aquesta manera s'aconsegueix un estalvi en temps ja que, per esborrar una entrada d'un fitxer, no ha de demanar cada dada per separat.

És aquest projecte, per al sistema de reconeixement de veu (SRV), s'ha fet servir, com s'ha esmentat anteriorment, una aplicació de Python que es basa en una 'aplicació' de Google, que funciona d'una manera molt simple: una petita aplicació de reconeixement lèxic, separa les síl·labes i es reproduïx el seu corresponent so prefixat. Per aquest motiu, no és molt complicat generar el nostre propi sistema SRV, com una millora substancial del sistema, d'aquesta manera no es dependria de tenir connexió a Internet.

## 7. Glossari

---

1. **Python.** Llenguatge de programació d'alt nivell de propòsit general. Guido van Rossum el va crear el 1991. Combina una potència remarcable amb una sintaxi clara i entenedora. Una característica important és la tipificació dinàmica i la capacitat per a interpretar el codi en temps d'execució.
2. **Prototip.** *m* Primer exemplar que es fabrica d'una figura, un invent o una altra cosa, i que serveix de model per fabricar altres iguals, o original amb el que es fabrica.
3. **Java.** *n* Llenguatge de programació orientat a objectes desenvolupat inicialment per Sun Microsystems i ara traspassat a Oracle (Oracle, 2010).
4. **Subsistema.** *f* Mòdul específic del projecte.
5. **Macro.** *n* és una instrucció complexa, formada per altres instruccions més senzilles. Això permet l'automatització de tasques repetitives.
6. **Plugin.** *n* és una aplicació (o programa informàtic) que es relaciona amb una altra per a agregar-li una funció nova i generalment molt específica.
7. **Diagrama de Gantt.** *f* és una eina gràfica amb l'objectiu d'exposar el temps de dedicació previst per a diferents tasques o activitats durant un determinat total de temps.
8. **Eclipse.** *n* és una plataforma de programari compost per un conjunt d'eines de programació de codi obert (multi plataforma) per desenvolupar el que el projecte anomena "Aplicacions de Client Enriquit".
9. **Sun Microsystems.** *n* La companyia va néixer amb la idea de fabricar i proporcionar sistemes complets, inicialment estacions de treball sota Unix BSD, al que es van anar unint servidors i eines d'administració de sistemes i xarxes, i programació.
10. **Unix.** *n* Com un dels sistemes operatius més importants en l'actualitat, el sistema UNIX és sens dubte una interessant aplicació que compta amb diverses utilitats i funcions.
11. **Cognitives.** *f* Les funcions cognitives ens permeten resoldre problemes, raonar i recordar experiències.
12. **Scheme.** *n* Scheme, com tots els dialectes de Lisp, té una sintaxi molt reduïda, comparat amb molts altres llenguatges. No necessita regles de precedència en la seva gramàtica, ja que fa servir notació prefixa per a totes les trucades a funció.
13. **LISP.** *n* L'acrònim LISP significa "LIST Processor" (Processament de llistes). Les llistes encadenades són una de les estructures de dades importants de Lisp, i el codi font de Lisp en si mateix està compost de llistes. Com a resultat, els programes de Lisp poden manipular el codi font com una estructura de dades, donant lloc als macro sistemes que permeten als programadors crear llenguatges de domini específic embeguts en Lisp.
14. **C++.** *n* és un llenguatge de programació dissenyat a mitjans dels anys 1980 per Bjarne Stroustrup. La intenció de la seva creació va ser el estendre al llenguatge de programació C mecanismes que permeten la manipulació d'objectes. En aquest sentit, des del punt de vista dels llenguatges orientats a objectes, el C ++ és un llenguatge híbrid.
15. **Ciència de la computació.** *f* Les ciències de la computació són aquelles que inclouen l'estudi de les bases teòriques de la informació i la computació i la seva aplicació en sistemes computacionals.

16. **Sistemes basats en regles.** *f* Aplicació basada en la representació dels coneixements que ofereix l'habilitat d'analitzar aquest coneixement usant regles de la forma IF-THEN.
17. **Multimèdia.** *f* És la presentació de la informació en un ordinador utilitzant àudio, vídeo, textos, animació i gràfics.
18. **Sistemes basats en procediments.** *f* Són tècniques de programació orientades a objectes que permeten l'enllaç de les característiques dels objectes en forma de codi. Quan un missatge és rebut pel codi, les seves característiques són executades.
19. **RGB.** *f* (sigla en anglés de *red, green, blue*, en espanyol «rojo, verde y azul»). RGB és un model de color basat en la síntesi additiva, amb el qual és possible representar un color mitjançant la barreja per addició dels tres colors de llum primaris.
20. **OpenCV.** *n* és una biblioteca lliure de visió artificialment originalment desenvolupada per Intel.
21. **MATLAB.** *n* (abreviatura de MATrix LABoratory, "laboratori de matrius") és una eina de programari matemàtic que ofereix un entorn de desenvolupament integrat (IDE) amb un llenguatge de programació pròpia (llengua M). Està disponible per a les plataformes Unix, Windows, Mac OS X i GNU / Linux.
22. **GNU.** *n* és un sistema operatiu de tipus Unix desenvolupat per i per al Projecte GNU, i auriat per la Free Software Foundation. Està format en la seva totalitat per programari lliure, majoritàriament sota termes de copyleft. GNU és l'acrònim recursiu de "GNU's Not Unix" (en espanyol: GNU no és Unix)
23. **Mac OS X.** *n* anteriorment denominada OS X i inicialment Mac OS X, és un entorn operatiu basat en Unix, desenvolupat, comercialitzat i venut per Apple Inc. Està inclòs en la seva gamma de computadores Macintosh des de l'any de 2002. OS X és el successor del Mac OS 9 (la versió final de Mac OS Classic), el sistema operatiu d'Apple des de 1984 està basat en BSD, i es va construir sobre les tecnologies desenvolupades en NeXT entre la segona meitat dels anys 80 i finals de 1996, quan Apple va adquirir aquesta companyia.
24. **Intel.** *n* és el major fabricant de circuits integrats del món, segons la seva xifra de negoci anual. La companyia nord-americana és la creadora de la sèrie de processadors x86, els processadors més comunament trobats en la majoria de les computadores personals. Intel va ser fundada el 18 de juliol de 1968 com Integrated Electronics Corporation pels pioners en semiconductors Robert Noyce i Gordon Moore, i moltes vegades associats amb la direcció executiva i la visió d'Andrew Grove.
25. **Google.** *n* Google LLC és una companyia principal subsidiària de la multinacional nord-americana Alphabet Inc, la especialització són els productes i serveis relacionats amb Internet, programari, dispositius electrònics i altres tecnologies. El principal producte de Google és el motor de cerca de contingut a Internet del mateix nom tot i que ofereix també altres productes i serveis com el correu electrònic anomenat Gmail, els seus serveis de mapes Google Maps, Google Street View i Google Earth.

## 8. Bibliografia

---

- [1] Wikipedia. "L'enciclopèdia de contingut lliure que tots poden editar". Robòtica [article en línia]. [Data de consulta: 17 d'Octubre del 2017]. <https://es.wikipedia.org/wiki/Rob%C3%B3tica>
- [2] Wikipedia. "L'enciclopèdia de contingut lliure que tots poden editar". Intel·ligència Artificial [article en línia]. [Data de consulta: 17 d'Octubre del 2017]. [https://es.wikipedia.org/wiki/Inteligencia\\_artificial](https://es.wikipedia.org/wiki/Inteligencia_artificial)
- [3] Wikipedia. "L'enciclopèdia de contingut lliure que tots poden editar". Reconocimiento del habla [article en línia]. [Data de consulta: 30 d'Octubre del 2017]. [https://es.wikipedia.org/wiki/Reconocimiento\\_del\\_habla](https://es.wikipedia.org/wiki/Reconocimiento_del_habla)
- [4] Vicenç Torra i Reventós (2012, 03 de novembre). Intel·ligència artificial [article en línia]. [Data de consulta: 17 d'Octubre del 2017]. <http://cataleg.uoc.edu/record=b1043450~S1%2Acat>
- [5] SearchDataCenter. Intel·ligència artificial [article en internet]. [Data de consulta: 17 d'Octubre del 2017]. <http://searchdatacenter.techtarget.com/es/definicion/Inteligencia-artificial-o-AI>
- [6] R. López-Cózar, A. J. Rubio, J. E. Díaz-Verdejo, J. M. López-Sole (2002, 28 de MAYO). "Validación de un Sistema de Diálogo Mediante el Uso de Diferentes Umbrales de Poda en el Proceso de Reconocimiento Automático de Voz". Procesamiento del lenguaje natural [article en línia]. [Data de consulta: 30 d'Octubre del 2017]. <http://www.sepln.org/revistaSEPLN/revista/29/29-Pag205.pdf>

- [8] e-Gnosis (2009, 01 de JULIO). "Revista Electrónica y Tecnológica". SISTEMA DE RECONOCIMIENTO AUTOMÁTICO DE OBJETOS [article en internet]. [Data de consulta: 23 d'Octubre del 2017].  
<https://doaj.org/article/b92dbebd9c8f4bf58c8699bc0e9452ef>
- [9] Santiago Salamanca Miño, Antonio Adán Oliver, Carlos Cerrada Somolinos, Miguel Adán, Pilar Merchán García (2007). "DIALNET". Reconocimiento de objetos de Forma Libre a Partir de los Datos de Rango de una Vista Parcial usando Cono Curvaturas Ponderadas [article en internet]. [Data de consulta: 24 d'Octubre del 2017].  
<https://dialnet.unirioja.es/servlet/articulo?codigo=2271726>
- [10] Departamento de Ingeniería Electrónica. UPM. (2009, 01 de JULIO). "Proyecto GEMINI". Realización de sistemas de diálogo en una plataforma compatible con VoiceXML [article en internet]. [Data de consulta: 06 de noviembre del 2017].  
<http://www.sepln.org/revistaSEPLN/revista/33/33-Pag103.pdf>
- [11] La Salle. (2002, 02 de JULIO). Un modelo híbrido orientado a la síntesis multimodal del habla [article en internet]. [Data de consulta: 31 d'Octubre del 2017].  
<http://www.sepln.org/revistaSEPLN/revista/29/29-Pag159.pdf>

## 9. Annexos

---

### A.1. Instal·lació del JDK

1. Fes doble clic en el programa de instal·lació del JDK: "jdk-8u121-windows-i586.exe".
2. Quan surti el quadre de llicència, polsa "Accept".
3. A la pantalla "Custom Setup", no canviis res i polsa "Next".
4. Quan acabi, polsa "Finish" i si et demana reiniciar el ordinador, fes-ho.

### A.2. Instal·lació de l'Eclipse

La instal·lació del Eclipse és bastant senzilla, heu de seguir els següents passos:

1. La versió que utilitzem es la Eclipse Neon 4.6 for Java EE Developers.

Agafeu l'Eclipse de <http://www.eclipse.org/downloads/>, concretament a l'adreça:

[http://www.eclipse.org/downloads/download.php?file=/technology/epp/downloads/release/neon/R/eclipse-jee-neon-R-win32.zip&mirror\\_id=17](http://www.eclipse.org/downloads/download.php?file=/technology/epp/downloads/release/neon/R/eclipse-jee-neon-R-win32.zip&mirror_id=17).

D'aquest enllaç obtindrem un ZIP que cal descomprimir-lo en el directori pare a on voleu tenir instal·lat l'Eclipse, per exemple c:\. Un cop fet això l'Eclipse ja està instal·lat. Per a obrir-lo només cal executar el fitxer eclipse.exe del directori d'instal·lació de l'Eclipse. Per a accedir-hi més còmodament podeu crear una icona d'accés directe a aquest fitxer en el mateix escriptori o allà on més us estimeu.

2. Executeu l'Eclipse. El primer cop que ho feu, us sortirà un missatge dient que s'està completant la instal·lació, i a continuació se us ha d'obrir una finestra semblant a la següent, que ens demana el directori a on guardar el projectes:



L'Eclipse és un entorn pensat per a poder treballar en qualsevol llenguatge de programació, i no només en Java. Per aquest motiu, haurem de configurar-lo per a treballar amb Java.

L'Eclipse permet treballar amb diferents perspectives, que ens seran útils per a diferents tasques. La perspectiva que s'obre per defecte al executar l'Eclipse per primer cop és la perspectiva "Resource". Haurem de tancar aquesta perspectiva i obrir les perspectives corresponents a Java.

### A.3. Instal·lació de Python

Per instal·lar Python necessitem baixar-nos la versió adequada al nostre sistema operatiu que podem trobar a <http://www.python.org/download>. Aquest tutorial utilitza la versió Python 3.5.4 per a Windows, i aquesta la podeu trobar baixant-vos el fitxer:

<https://www.python.org/ftp/python/3.5.4/python-3.5.4.msi>

Executem aquest fitxer, acceptant totes les opcions que venen per defecte:



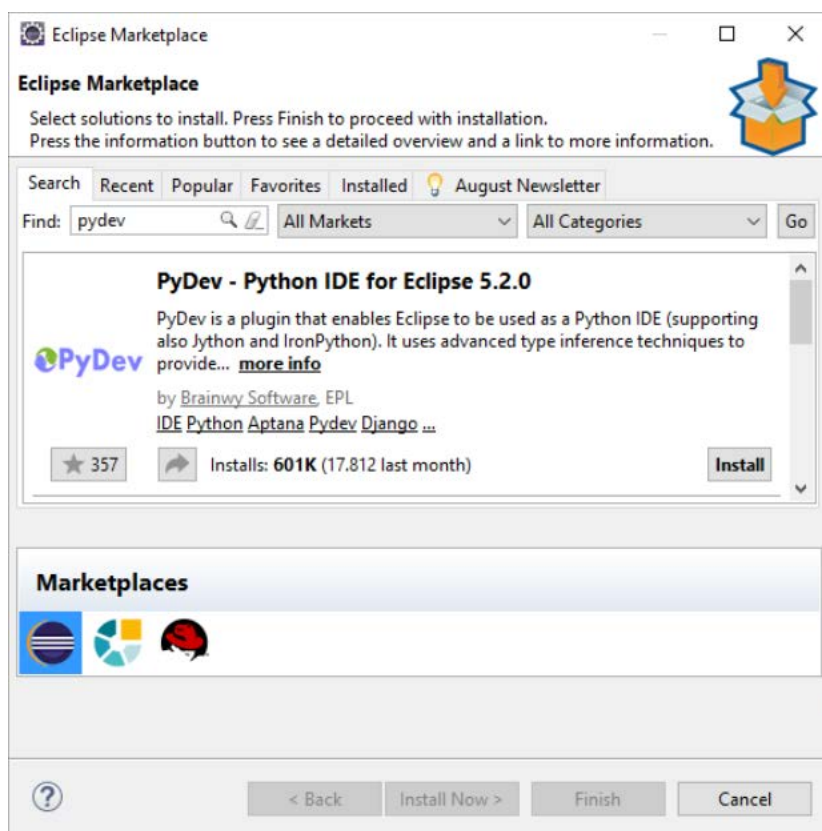


#### A.4. Instal·lació de PyDev

PyDev es un plugin per Eclipse, que ens permet utilitzar Eclipse com un IDE per a Python.

Per instal·lar PyDev cal seguir els passos:

- Help -> Eclipse Marketplace -> Find: PyDev -> Install -> Confirm.... -> Accept...

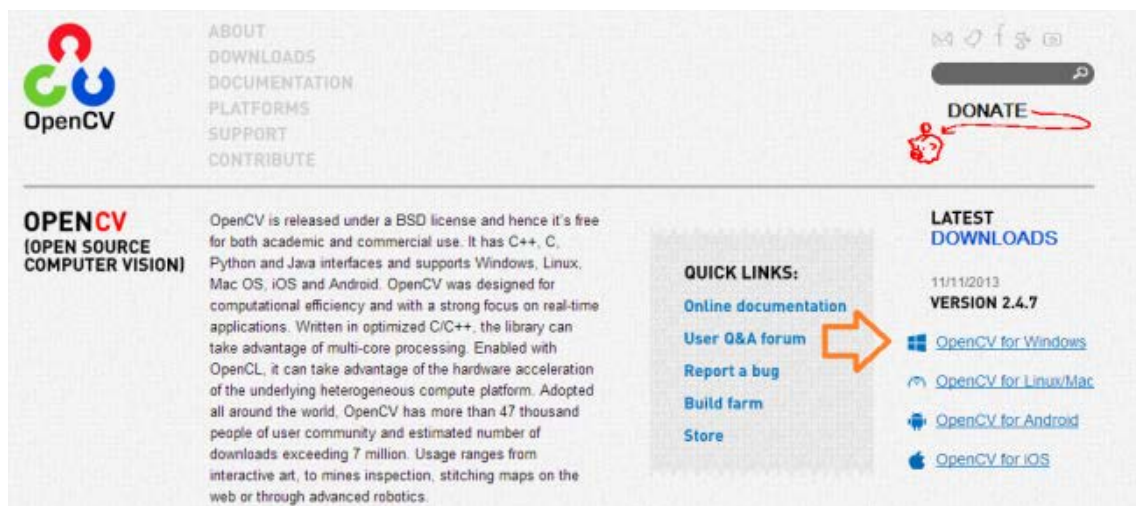


- Ara cal indicar al plugin PyDev quina versió de Python utilitzarem i a on es troba: Windows -> Preferences -> PyDev -> Interpreters -> Python Interpreter -> New

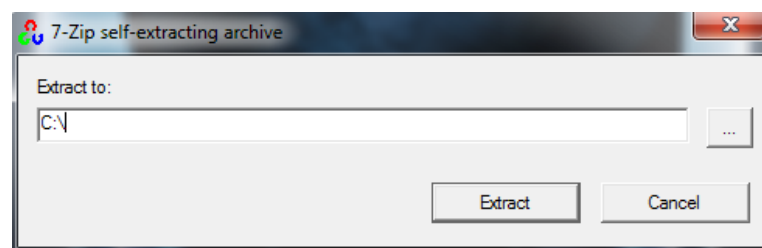
## A.5. Instal·lació de OpenCV

Seguiu els següents passos per utilitzar la biblioteca de visió per ordinador OpenCV en Python:

1.- Descarregar OpenCV en la seva versió més recent: <http://opencv.org/>



2.- Instal·lar OpenCV en C:\



3.- Descarregar Python 3.5.4 per a Windows (OpenCV és més estable amb la versió de 32 bits) desde: <http://www.python.org/>




## Python 2.7.6 Release

Python 2.7.6 was released on November 10, 2013. This is a 2.7 series bugfix release. Most importantly, it resolves an issue that caused the interactive prompt to crash on OS X 10.9. It also includes numerous bugfixes over 2.7.5.

### Download

This is a production release. Please [report any bugs](#) you encounter.

We currently support these formats for download:

- [Windows x86 MSI Installer \(2.7.6\) \(sig\)](#) 
- [Windows x86 MSI program database \(2.7.6\) \(sig\)](#)
- [Windows X86-64 MSI Installer \(2.7.6\) \[1\] \(sig\)](#)
- [Windows X86-64 MSI program database \(2.7.6\) \[1\] \(sig\)](#)
- [Windows help file \(sig\)](#)

4.- Instal·lar Python 3.5.4 donant la ruta per defecte, amb això Python s'instal·larà a C: / Python3.5.4

5.- Descarregar NumPy i matplotlib des dels següents links:

[http://sourceforge.net/projects/numpy/files/NumPy/1.8.0/numpy-1.8.0-win32-superpack-python2.7.exe/download?use\\_mirror=softlayer-ams](http://sourceforge.net/projects/numpy/files/NumPy/1.8.0/numpy-1.8.0-win32-superpack-python2.7.exe/download?use_mirror=softlayer-ams)

[http://sourceforge.net/projects/matplotlib/files/matplotlib/matplotlib-1.3.0/matplotlib-1.3.0.win32-py2.7.exe/download?use\\_mirror=softlayer-dal](http://sourceforge.net/projects/matplotlib/files/matplotlib/matplotlib-1.3.0/matplotlib-1.3.0.win32-py2.7.exe/download?use_mirror=softlayer-dal)

Nota: Provat amb èxit en la versió 2.4.8 de OpenCV, per a altres versions de OpenCV pot ser necessari una altra versió de NumPy. Un missatge a la consola de Python indicarà la versió de NumPy necessària si ocorre un error.

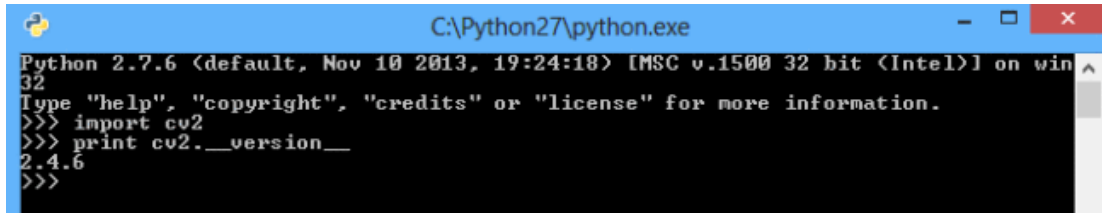
6.- Instal·lar NumPy i matplotlib amb els valors per defecte.

7.- Anar a `C:\opencv/build/python/3.5.4/x86` i copiar el arxiu `cv2.pyd` a `C:/Python3.5.4/lib/site-packages`

8.- Obrir Python (command line).

9.- Escriure a la consola que apareix el següent:

```
>>> import cv2
>>> print cv2.__version__
```



```
C:\Python27\python.exe
Python 2.7.6 <default, Nov 10 2013, 19:24:18> [MSC v.1500 32 bit <Intel>] on win
32
Type "help", "copyright", "credits" or "license" for more information.
>>> import cv2
>>> print cv2.__version__
2.4.6
>>>
```