
Sistema d'autolocalització per a robots mòbils mitjançant tècniques de visió per computador

Joan Rodas Cusidó
02-10-2016

Treball final de grau en eng. informàtica
Tecnologies de la informació

Lliurable 2 - GEP



Facultat d'Informàtica de Barcelona
Universitat Politècnica de Catalunya
Director: Joan Climent Vilaró

Índex

1	Introducció	4
1.1	Context	4
1.2	Descripció del problema	4
1.3	Actors implicats	4
1.4	Estat de l'art	5
1.4.1	Visió per computador	5
1.4.2	Robòtica	6
2	Definició de l'abast	8
2.1	Objectius	8
2.2	Requeriments	8
2.3	Obstacles	9
2.3.1	Noves eines	9
2.3.2	Calendari	9
2.3.3	Error de programació	9
2.3.4	Condicions variables en les imatges	9
2.4	Ampliacions	10
2.5	Metodologia	10
2.6	Eines de desenvolupament	10
2.6.1	OpenCV	11
2.7	Eines de seguiment	11
2.7.1	LibreOffice Calc	11
2.7.2	Gantt Project	11
2.7.3	Git + Github	11
2.8	Mètode de validació	12
3	Planificació	13
3.1	Planificació temporal	13
3.1.1	Bloc 0: Preparació de l'entorn	13
3.1.2	Bloc 1: Curs de GEP	14
3.1.3	Bloc 2: Desenvolupament del projecte	14
3.1.4	Bloc 3: Preparació de la defensa	15

3.1.5	Diagrames	15
3.2	Recursos	16
3.2.1	Recursos humans	16
3.2.2	Recursos de maquinari	16
3.2.3	Recursos de programari	16
3.3	Desviacions i pla d'actuació	17
4	Gestió econòmica	18
4.1	Costos directes	18
4.1.1	Recursos humans	18
4.1.2	Recursos de maquinari	18
4.1.3	Recursos de programari	19
4.2	Costos indirectes	19
4.3	Desviacions	19
4.4	Costos totals	20
5	Sostenibilitat	21
5.1	Sostenibilitat econòmica	21
5.2	Sostenibilitat social	21
5.3	Sostenibilitat ambiental	21
	Apèndix	22
	A Diagrames	23
	Índex de taules	26
	Índex de figures	27

1. Introducció

Els avenços tecnològics dels últims anys, han millorat la capacitat de les màquines per extreure informació i resoldre problemes de manera autònoma, imitant cada vegada millor el comportament humà.

En aquest treball, es treballarà la visió per computador aplicada a un problema de robòtica.

1.1 Context

Aquest projecte es desenvolupa com a treball final de grau dels estudis de grau en enginyeria informàtica, de l'especialitat en tecnologies de la informació. Es tracta d'un projecte de modalitat A, realitzat a la Facultat d'Informàtica de Barcelona (Universitat Politècnica de Catalunya) i proposat pel director Joan Climent, del departament d'ESAI (Enginyeria de Sistemes, Automàtica i Informàtica Industrial).

1.2 Descripció del problema

El treball pretén resoldre un problema d'autolocalització de robots mòbils en un entorn variable, de tal manera que el robot sigui capaç de desplaçar-se d'un punt inicial a un punt final escollit per l'usuari. Per fer això, s'utilitzaran diverses tècniques de visió per ordinador.

1.3 Actors implicats

En aquesta secció es descriuen els actors implicats del projecte, és a dir, totes aquelles persones que es veuran beneficiades directa o indirectament amb la realització d'aquest.

- **Autor/Desenvolupador:** És el màxim responsable del projecte. En tractar-se d'un treball final de grau, l'autor del projecte serà també el màxim beneficiari, ja que la realització d'aquest li permetrà acabar la carrera d'enginyeria informàtica.
- **Usuaris:** Qualsevol persona qui ho desitgi, tindrà accés a tots els codis desenvolupats durant el projecte, ja que es llançaran sota una llicència de programari lliure que permetrà veure i adaptar el codi a les necessitats d'altres usuaris.
- **Altres beneficiaris:** Qualsevol empresa o institució interessada podrà utilitzar el sistema desenvolupat i adaptar-lo a les seves necessitats, com podria ser per exemple un sistema de transport d'equipatge basat en robots.

1.4 Estat de l'art

1.4.1 Visió per computador

La visió per computador[Szeliski] és una ciència que té com a objectiu dotar les màquines o ordinadors de la capacitat de “veure”. Es basa en l'extracció i anàlisi de dades obtingudes a partir d'imatges.

Algunes de les aplicacions de la visió per computador són:

- Vehicles autònoms
- Realitat augmentada
- Reconeixement facial
- Restauració d'imatges
- Inspecció industrial
- Robòtica

En aquest treball, ens interessa utilitzar la visió per computador en el camp de la robòtica, per aconseguir guiar a un robot mòbil cap a un objectiu determinat basant-se en la detecció d'un punt o regió en una imatge.

Nous algorismes

En els darrers anys, han aparegut nous algorismes d'obtenció de punts i extracció de característiques que suposen una alternativa als clàssics SIFT[**SIFT**] (Scale Invariant Feature Transform) i SURF[**SURF**] (Speeded-Up Robust Features). Alguns d'aquests algorismes són BinBoost[**Trzcinski13a**] o un dels més recents: LATCH[**LeviHassner2016LATCH**]

En aquest projecte s'analitzarà si es adequat emprar algun d'aquests algorismes en la implementació del sistema d'autolocalització.

1.4.2 Robòtica

La robòtica és un camp de la tecnologia que estudia el disseny i la construcció de robots.

Que és, doncs, un robot? Al llarg de la història, s'han donat diverses definicions del concepte de robot, sense existir encara una definició exacta acceptada per tothom. I a mesura que passa el temps, cada vegada resulta més complicat determinar si una màquina és o no un robot. Per no complicar-nos massa, entendrem com a robot una màquina programable capaç de realitzar una sèrie de tasques concretes interactuant amb l'entorn, ja sigui de manera automàtica o dirigida.

Existeixen diversos tipus de robots, podent fer una classificació senzilla segons la seva arquitectura: robots mòbils, poliarticulats (industrials, mèdics, etc.), humanoides, zoomòrfics¹ i híbrids.

Els robots mòbils, que són els que ens interessen per aquest projecte, acostumen a tenir una sèrie de sensors i dispositius per permetre'n el desplaçament, la localització, esquivar obstacles i realitzar tasques concretes. Alguns exemples de sensors utilitzats per robots mòbils són:

- Odometria: S'utilitza la informació obtinguda amb sensors de moviment (*encoders* a les rodes, per exemple) per estimar la posició del robot respecte a la inicial.
- GPS (Global Positioning System): Es determina la ubicació del robot amb la xarxa de satèl·lits.
- Sensors de contacte: Permeten detectar si el robot està en contacte amb un altre objecte.

¹**Robots zoomòrfics:** Robots que imiten característiques pròpies de determinats animals.

- Sensors d'ultrasons: Detecten objectes mitjançant ones ultrasòniques.
- Acceleròmetre: Determina l'acceleració del robot quan es mou.
- Càmera: Permet capturar imatges de l'entorn.

En el nostre cas, només ens interessaran les dades obtingudes a través d'una càmera, és a dir, les imatges. El treball no se centrarà per tant en la part robòtica del sistema, i no es tindran en compte els sensors i algorismes necessaris per poder moure el robot.

En cas d'aplicar el sistema desenvolupat en robots en un futur, aleshores s'hauran de tenir en compte altres sensors per permetre el moviment de la màquina i arribar a la destinació evitant obstacles.

2. Definició de l'abast

2.1 Objectius

L'objectiu principal del projecte consisteix a dissenyar i desenvolupar un sistema d'autolocalització per a robots mòbils.

Aquest sistema estarà basat en tècniques de visió per computador i consistirà, bàsicament, a comparar dues imatges (una global i una altra capturada pel robot) i localitzar un punt o regió seleccionat per l'usuari.

Per arribar a aquest objectiu, es dividirà el treball en diverses fases:

- Estudi dels diferents algorismes de visió existents
- Obtenció de *keypoints* en una imatge
- Extracció de característiques
- *Matching* de dues imatges

2.2 Requeriments

El sistema d'autolocalització implementat ha de complir amb una sèrie de requeriments mínims presentats a continuació:

- L'usuari ha de poder seleccionar un punt o regió d'interès en una imatge donada.
- El sistema ha de ser capaç d'adaptar-se mínimament a diverses condicions de l'entorn (canvis de lluminositat, perspectiva, etc.).

2.3 Obstacles

Durant la planificació i realització del treball, s'hauran de tenir en compte els possibles obstacles que es trobaran. A continuació es detallen alguns dels problemes que es podran trobar.

2.3.1 Noves eines

Un dels principals obstacles serà el fet de treballar amb noves eines i algorismes. Per tal d'evitar problemes en aquest aspecte, caldrà fer una planificació acurada i documentar-se apropiadament. També serà important mantenir una bona comunicació amb el tutor en tot moment, per poder resoldre possibles dubtes referents als algorismes.

2.3.2 Calendari

Un altre obstacle important serà la falta de temps, ja que està previst realitzar el projecte en el transcurs d'un quadrimestre. Gestionar correctament el temps serà clau per aconseguir finalitzar el projecte sense problemes. Per tant, s'haurà de fer una planificació el més realista possible i escollir una metodologia de treball adequada i flexible.

2.3.3 Errors de programació

Com a qualsevol projecte on s'ha de programar, el codi serà una font important d'errors. Per això, caldrà realitzar diverses proves cada vegada que es realitzi una modificació en el codi o s'implementi una nova funcionalitat.

2.3.4 Condicions variables en les imatges

Les imatges capturades a través d'una càmera no presentaran sempre les mateixes condicions. La lluminositat, perspectiva o resolució de la imatge influiran a l'hora de processar les imatges i comparar-les.

Per intentar minimitzar aquests efectes, s'analitzaran diversos algorismes d'obtenció de punts i extracció de característiques. També s'estudiarà si és necessari realitzar un preprocessament o filtratge de les imatges abans d'aplicar els algorismes.

2.4 Ampliacions

Encara que el calendari és força estricte i no hi ha gaire marge d'ampliació, es podria estendre el projecte amb les següents ampliacions:

- Anàlisi del rendiment d'algorismes alternatius per l'obtenció de punts i característiques de les imatges.
- Creació d'una aplicació d'Android que permeti seleccionar un punt o regió d'una imatge.
- Execució del codi del sistema via servidor web, utilitzant les dades enviades per l'aplicació d'Android.

2.5 Metodologia

Per aquest projecte, s'utilitzarà una metodologia de treball àgil amb cicles de desenvolupament curts. Com que només hi ha un desenvolupador, no s'utilitzaran exactament les metodologies Scrum o XP[**Pxp**] (Extreme Programming), però sí que s'aplicaran moltes de les pràctiques pròpies d'aquestes dues metodologies (proves, simplicitat, refacció de codi, etc.). Això ens donarà més flexibilitat a l'hora de fer canvis i adaptar-nos a una nova planificació.

Es començarà treballant amb imatges de prova (casos senzills) i algorismes coneguts com ara Harris[**Harris**] i SIFT. Més endavant, s'aniran introduint modificacions en el codi per intentar aconseguir un sistema capaç de funcionar amb fotografies “reals” i es provaran altres algorismes de visió per computador.

Per altra banda, s'utilitzarà el mètode en cascada per la realització del curs de GEP.

2.6 Eines de desenvolupament

El codi del projecte es desenvoluparà amb python i s'utilitzaran, sempre que sigui possible, eines de programari lliure o de codi obert.

En cas de crear una aplicació per a dispositius Android, es realitzarà mitjançant Android Studio (Java).

2.6.1 OpenCV

Per tal d'utilitzar algorismes de visió per computador en el codi amb relativa facilitat, s'utilitzarà la biblioteca de codi obert OpenCV[**OpenCV**] (Open Source Computer Vision Library), disponible per a python. La versió emprada serà la 3.1.

En concret, hi haurà tres passos indispensables que faran ús d'aquesta biblioteca:

- Obtenció de punts en una imatge
- Extracció de característiques
- *Matching* de dues imatges

2.7 Eines de seguiment

A continuació es detallen les eines de programari usades per fer el seguiment del treball final de grau:

2.7.1 LibreOffice Calc

Per fer un seguiment de les hores dedicades al projecte, es crearà un full de càlcul amb les hores diàries dedicades a cada tasca. S'utilitzarà LibreOffice Calc, inclòs en la *suite* ofimàtica LibreOffice.

2.7.2 Gantt Project

Per tal d'organitzar totes les tasques a realitzar i mirar si hi ha desviacions respecte el pla inicial, s'utilitzarà l'eina de *software* lliure Gantt Project[**GanttProject**]. Aquesta eina ens permetrà realitzar tant un diagrama de Gantt com un diagrama de PERT.

2.7.3 Git + Github

Tot i que no es tracta d'un projecte col·laboratiu (només hi ha un desenvolupador), s'ha decidit utilitzar el sistema de control de versions Git juntament amb la pàgina web Github. D'aquesta manera es facilitarà treballar amb diverses màquines i portar un control dels canvis realitzats. A més, permetrà compartir el codi amb el director amb facilitat.

2.8 Mètode de validació

Es faran validacions parcials durant la realització del projecte, fent proves del sistema amb diverses imatges.

Contacte amb el director

Hi haurà reunions presencials amb el director, així com comunicació via correu electrònic, per tal de resoldre dubtes i validar la feina realitzada.

3. Planificació

3.1 Planificació temporal

El treball té una duració aproximada de 4 mesos i mig, des de setembre fins a principis de gener. La càrrega total serà d'unes 450 hores, corresponents a 18 crèdits ECTS. La dedicació setmanal estimada serà d'unes 25 hores.

Es dividirà el projecte en quatre blocs, descrits a continuació:

Bloc	Descripció	Metodologia	Hores
Bloc 0	Preparació de l'entorn	-	5h
Bloc 1	Curs de GEP	Cascada	75h
Bloc 2	Desenvolupament del projecte	Àgil	340h
Bloc 3	Preparació de la defensa	-	30h

Taula 3.1: Blocs del projecte

3.1.1 Bloc 0: Preparació de l'entorn

Inicialment, s'instal·larà tot el programari necessari per començar a desenvolupar el projecte i es faran algunes proves bàsiques per familiaritzar-se amb el nou entorn de treball. Aquest primer bloc tindrà una durada aproximada de 5 hores.

Per poder començar a treballar en el projecte, caldrà instal·lar:

- **Desenvolupament:** Python, OpenCV, Geany i Git.
- **Curs de GEP:** Gantt Project.
- **Documentació:** L^AT_EX i Zathura.

3.1.2 Bloc 1: Curs de GEP

Aquest bloc correspon a la realització del curs de GEP, amb inici el dia 19/09/2016 i finalització el 24/10/2016 (amb una presentació final entre el 7 i l'11 de novembre). Té com a dependència el bloc 0.

Durant el curs s'entregaran 6 lliurables, detallats a continuació:

Descripció	Inici	Finalització	Durada	Hores
Introducció i abast	19/09/2016	27/09/2016	9 dies	16h
Planificació temporal	28/09/2016	03/10/2016	6 dies	9h
Gestió econòmica i sostenibilitat	04/10/2016	10/10/2016	7 dies	10h
Presentació preliminar en vídeo	11/10/2016	17/10/2016	7 dies	11h
Plec de condicions	18/10/2016	24/10/2016	7 dies	13h
Document final + presentació	18/10/2016	24/10/2016	7 dies	16h

Taula 3.2: Lliurables de GEP

3.1.3 Bloc 2: Desenvolupament del projecte

El bloc principal consistirà en el desenvolupament del projecte en si mateix: buscar informació, implementar el codi, redactar la memòria, etc.

Aquest bloc té com a dependència el bloc 0 i es dividirà en quatre tasques.

Tasca	Inici	Finalització	Durada	Hores
Implementació i proves	13/09/2016	19/12/2016	98 dies	240h
Conclusions i resultats	20/12/2016	31/12/2016	12 dies	30h
Ampliacions (opcional)	01/01/2017	09/01/2017	9 dies	30h
Redacció de la memòria	10/10/2016	09/01/2016	92 dies	40h

Taula 3.3: Tasques desenvolupament

Recerca d'informació, implementació i proves

Durant tot el projecte, se cercarà informació contínuament i s'aniran fent proves a mesura que s'implementa el codi.

Conclusions i resultats

Un cop enllestida la implementació, es procedirà a elaborar les conclusions d'acord amb els resultats obtinguts. Es compararan els resultats obtinguts amb diferents algorismes de visió i es faran proves del sistema amb diverses imatges.

Ampliacions (opcional)

Un cop realitzades les conclusions, en cas de disposar de més temps, es podran fer ampliacions i millores.

Redacció de la memòria

La memòria s'anirà redactant a mesura que es realitza el projecte. No hi ha per tant cap dependència, encara que es dedicarà més temps en l'etapa final del treball.

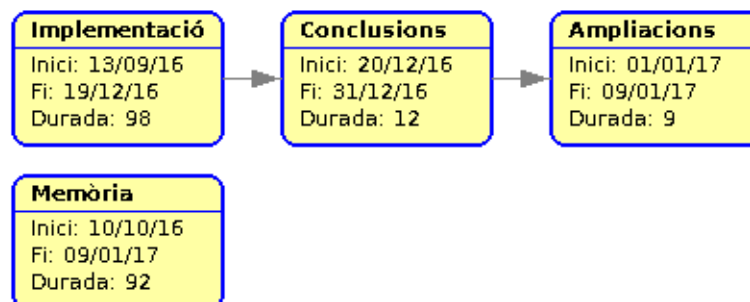


Figura 3.1: PERT - tasques desenvolupament

3.1.4 Bloc 3: Preparació de la defensa

En aquest bloc final es revisarà la memòria del projecte i es prepararà la presentació. Està previst dedicar unes 30 hores al bloc, que començarà el dia 10 de gener i acabarà el 22. La defensa del projecte es durà a terme entre els dies 23 i 27 de gener.

3.1.5 Diagrames

Podeu trobar tots els diagrames de Gantt i PERT a l'apèndix A del treball.

3.2 Recursos

En aquesta secció s'analitzen els recursos necessaris per a la realització del projecte.

3.2.1 Recursos humans

El projecte el realitzarà una sola persona, que haurà d'assumir els rols de cap de projecte, analista, dissenyador, programador i *tester*. També es comptarà amb l'ajuda del director del projecte, que assumirà el paper de consultor/supervisor.

3.2.2 Recursos de maquinari

Per la realització del projecte no serà necessari adquirir cap mena de maquinari específic. Es podrà utilitzar un ordinador personal per treballar a casa i els ordinadors disponibles a la FIB per treballar des de la universitat.

Es treballarà principalment amb un ordinador equipat amb un processador AMD FX 6300 hexa-core 3.5GHz, 4GB de RAM i 250GB de disc dur SSD. També s'utilitzarà una càmera o smartphone qualsevol.

3.2.3 Recursos de programari

Durant la realització del projecte i el curs de GEP, s'utilitzaran diverses eines de programari, detallades a continuació:

Nom	Tipus	Ús
Arch Linux	Eina de desenvolupament	Execució del programari
Python	Eina de desenvolupament	Programació
OpenCV	Eina de desenvolupament	Algorismes de VC
Geany	Eina de desenvolupament	Programació del codi
L ^A T _E X	Eina de desenvolupament	Redacció de la memòria
Zathura	Eina de desenvolupament	Visualització de pdf
Gantt Project	Eina de gestió	Creació diagrames de Gantt
LibreOffice Calc	Eina de gestió	Control de les hores
Git + Github	Desenvolupament i gestió	Control de versions

Taula 3.4: Recursos de programari

3.3 Desviacions i pla d'actuació

Mala planificació [Impacte: baix]

Hi haurà reunions amb el director i s'usaran eines de planificació per mirar de corregir la planificació i acabar el projecte a temps. També es reserven unes hores a l'ampliació del treball, que es podrien utilitzar en cas que una tasca s'allargués més del previst. Si fos necessari, es podria incrementar una mica la càrrega de treball setmanal.

Fallades de maquinari [Impacte: baix]

En cas de fallades en l'ordinador principal, no hi hauria cap problema en utilitzar-ne un altre. No hi ha dependències de hardware i es disposa d'altres ordinadors (a casa i a la FIB). Tampoc hi hauria una pèrdua de dades important, ja que es treballa amb Github i una còpia local.

4. Gestió econòmica

4.1 Costos directes

4.1.1 Recursos humans

Rol	Hores	Cost/hora	Cost total
Cap de projecte	h	€/h	€
Programador	h	€/h	€
<i>Tester</i>	h	€/h	€
Analista	h	€/h	€

Taula 4.1: Recursos de programari (costos)

4.1.2 Recursos de maquinari

Producte	Unitats	Preu unitari	Vida útil	Amortització	Cost
Ordinador personal	1	500€	5 anys	-	-
Càmera	1	€	10 anys	-	-

Taula 4.2: Recursos de maquinari (costos)

4.1.3 Recursos de programari

Producte	Llicència	Preu	Vida útil	Amortització	Cost
Arch Linux	ll	0€	-	-	0€
Arch Linux	ll	0€	-	-	0€
Arch Linux	ll	0€	-	-	0€
Arch Linux	ll	0€	-	-	0€
Arch Linux	ll	0€	-	-	0€
Arch Linux	ll	0€	-	-	0€
Arch Linux	ll	0€	-	-	0€
Arch Linux	ll	0€	-	-	0€
Arch Linux	ll	0€	-	-	0€
Arch Linux	ll	0€	-	-	0€

Taula 4.3: Recursos de programari (costos)

4.2 Costos indirectes

Tipus	Hores	Cost/hora	Cost total
Electricitat	450h	€/h	€
Internet	450h	€/h	€

Taula 4.4: Recursos humans (costos)

4.3 Desviacions

asdasdasdasdasdasdasda

4.4 Costos totals

Tipus	Cost
Costos directes	€
Costos indirectes	€
Costos totals	€

Taula 4.5: Recursos humans (costos)

5. Sostenibilitat

[MATRIU]

5.1 Sostenibilitat econòmica

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque quis pellentesque odio. Sed hendrerit nisi ut metus malesuada placerat. Aenean a turpis tempus, auctor elit nec, elementum ante. Pellentesque faucibus malesuada pulvinar. Duis tristique est nibh, in dapibus est iaculis ut. Vestibulum gravida congue mi, a sodales est imperdiet at. Aliquam varius condimentum nisi, et pulvinar orci egestas eget. Sed pharetra metus nec enim fermentum commodo. Aenean auctor purus sit amet felis imperdiet semper. Ut scelerisque, ex eu tristique pretium, tellus purus vehicula magna, sed ornare dolor sem ut sapien. Morbi sit amet odio odio. Maecenas sodales nulla nec velit tempor, eu aliquam sapien cursus. Sed volutpat libero a nunc luctus, sagittis commodo ipsum ultrices.

5.2 Sostenibilitat social

Phasellus consectetur tincidunt nunc, quis ultrices sapien rutrum non. Sed posuere lacinia mattis. Nunc fringilla ultricies dolor a ullamcorper. Praesent eu odio et dui aliquam aliquam eget a nibh. Nullam metus eros, convallis nec rhoncus eu, fermentum nec libero. Mauris finibus nunc in dapibus rhoncus. Cras ac lacus at nunc elementum tristique nec sed arcu. Aliquam ex neque, finibus sed laoreet ut, fringilla at nulla.

5.3 Sostenibilitat ambiental

Sed mattis turpis a nisi sodales, nec sodales nisi finibus. Mauris et nisi sit amet velit gravida facilisis et in lectus. Suspendisse sodales erat non semper fermentum. Mauris nec lacus non sapien fermentum posuere. Sed hendrerit nisl ac lectus consectetur sagittis. Cras vehicula lacus eget augue feugiat tincidunt. Vivamus vulputate enim velit, eget volutpat dolor dignissim at.

Apèndix

A. Diagrames

En aquest apèndix trobareu els diagrames de Gantt i PERT realitzats durant el curs de GEP.

A.1 Projecte

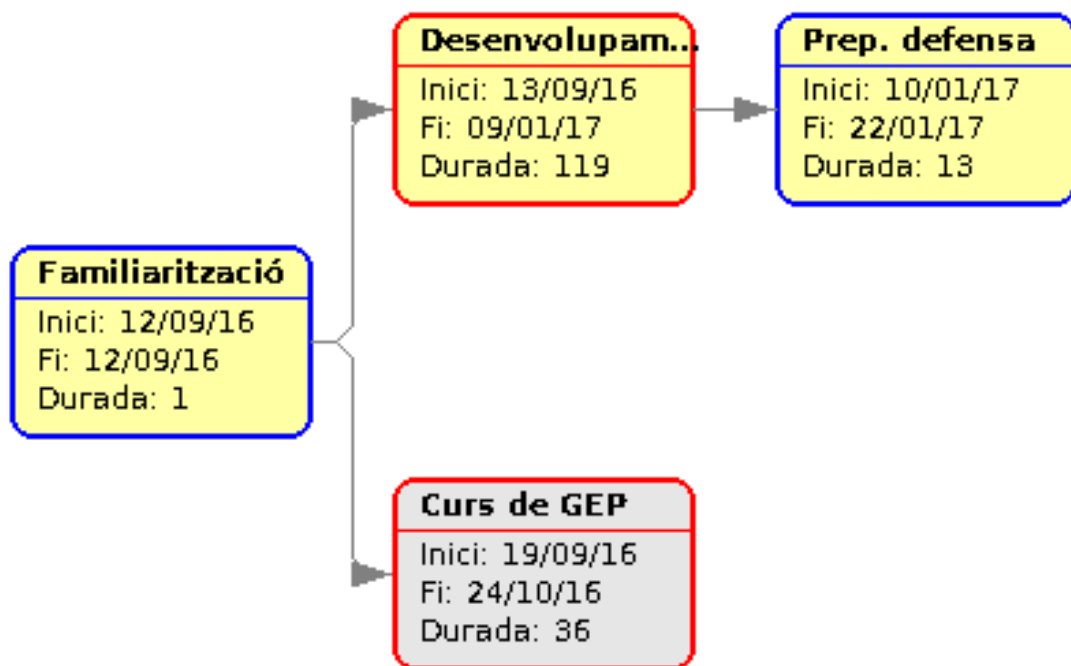


Figura A.1: PERT del projecte

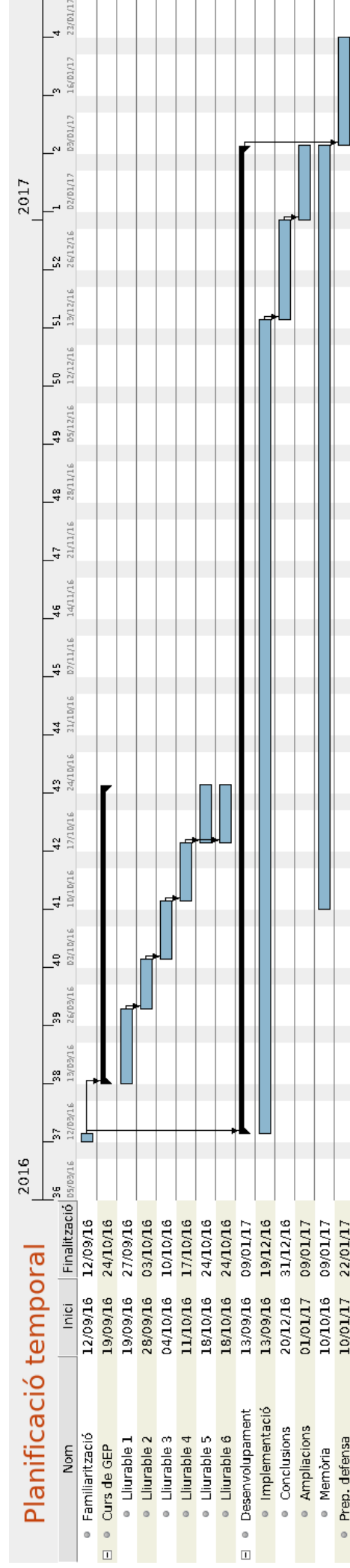


Figura A.2: Gantt del projecte

A.2 Curs de GEP

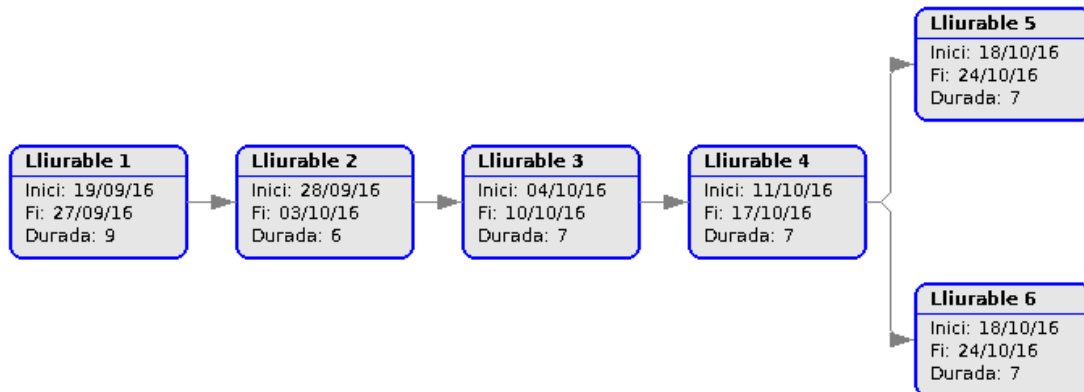


Figura A.3: PERT - curs de GEP

A.3 Desenvolupament

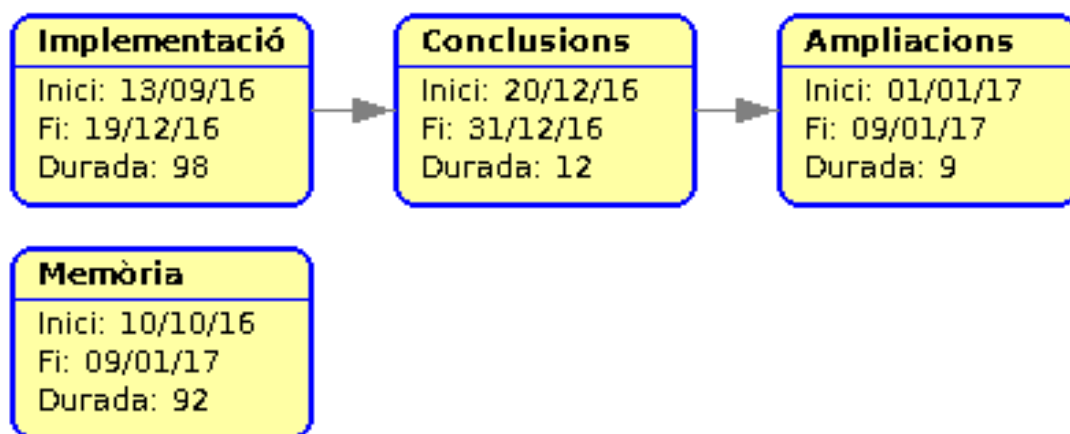


Figura A.4: PERT - tasques desenvolupament

Índex de taules

3.1	Blocs del projecte	13
3.2	Lliurables de GEP	14
3.3	Tasques desenvolupament	14
3.4	Recursos de programari	16
4.1	Recursos de programari (costos)	18
4.2	Recursos de maquinari (costos)	18
4.3	Recursos de programari (costos)	19
4.4	Recursos humans (costos)	19
4.5	Recursos humans (costos)	20

Índex de figures

3.1	PERT - tasques desenvolupament	15
A.1	PERT del projecte	23
A.2	Gantt del projecte	24
A.3	PERT - curs de GEP	25
A.4	PERT - tasques desenvolupament	25