

---

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

---

Joan Rodas Cusidó  
10-12-2016

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

Informe de seguiment



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

# Índex

<b>1</b>	<b>Introducció i abast</b>	<b>4</b>
1.1	Descripció del problema . . . . .	5
1.2	Motivació . . . . .	5
1.3	Actors implicats . . . . .	5
1.4	Estat de l'art . . . . .	6
1.4.1	Visió per computador . . . . .	6
1.4.2	Robòtica . . . . .	7
1.5	Objectius . . . . .	8
1.6	Requeriments . . . . .	8
1.7	Obstacles . . . . .	9
1.8	Ampliacions . . . . .	10
1.9	Metodologia . . . . .	10
1.10	Eines de desenvolupament . . . . .	10
1.10.1	OpenCV . . . . .	11
1.11	Eines de seguiment . . . . .	11
1.12	Mètode de validació . . . . .	12
<b>2</b>	<b>Planificació i recursos</b>	<b>13</b>
2.1	Planificació temporal . . . . .	13
2.1.1	Bloc 0: Preparació de l'entorn . . . . .	13
2.1.2	Bloc 1: Curs de GEP . . . . .	14
2.1.3	Bloc 2: Desenvolupament del projecte . . . . .	14
2.1.4	Bloc 3: Preparació de la defensa . . . . .	16
2.1.5	Diagrames . . . . .	16
2.2	Recursos . . . . .	18
2.2.1	Recursos humans . . . . .	18
2.2.2	Recursos de maquinari . . . . .	18
2.2.3	Recursos de programari . . . . .	18
2.3	Desviacions i pla d'actuació . . . . .	19

<b>3</b>	<b>Gestió econòmica</b>	<b>20</b>
3.1	Recursos de programari . . . . .	20
3.2	Recursos humans . . . . .	20
3.3	Recursos de maquinari . . . . .	21
3.4	Costos indirectes . . . . .	21
3.5	Imprevistos . . . . .	22
3.6	Contingència . . . . .	22
3.7	Costos totals . . . . .	22
3.8	Control de gestió . . . . .	22
<b>4</b>	<b>Informe de sostenibilitat</b>	<b>23</b>
4.1	Posada en producció . . . . .	24
4.1.1	Consum del disseny . . . . .	24
4.1.2	Factura . . . . .	24
4.1.3	Impacte personal . . . . .	24
4.2	Vida útil . . . . .	25
4.2.1	Petjada ecològica . . . . .	25
4.2.2	Pla de viabilitat . . . . .	25
4.2.3	Impacte social . . . . .	25
4.3	Riscos . . . . .	26
4.3.1	Riscos ambientals . . . . .	26
4.3.2	Riscos econòmics . . . . .	26
4.3.3	Riscos socials . . . . .	26
	<b>Bibliografia</b>	<b>27</b>
	<b>Índex de taules</b>	<b>28</b>
	<b>Índex de figures</b>	<b>29</b>

# 1. Introducció i abast

*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).*

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.

El primer capítol serveix com a introducció del projecte, on s'explica l'abast, objectiu, motivació i estat de l'art de les tecnologies a tractar. En el segon capítol es detallen els recursos utilitzats per realitzar el treball.

Els capítols 3 a 6 conformen el treball principal. Al tercer capítol s'explica el disseny i l'aquitectura del sistema desenvolupat. Al quart, les tècniques de visió utilitzades, amb la seva implementació detallada al cinqué capítol. Al capítol 6 s'explicaràn els experiments realitzats i els resultats obtinguts.

Al capítol 7 podeu trobar un anàlisi de la gestió econòmica del projecte, on es detallen els costos humans, de programari, de maquinari, indirectes i possibles imprevistos. I al vuité capítol es presenta l'informe de sostenibilitat. Per acabar, hi haurà les conclusions del projecte, on es valorarà l'aportació del projecte a nivell personal i si s'han aconseguit els objectius inicials proposats.

## 1.1 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.

## 1.2 Motivació

Visió per computador i Robòtica van ser sense cap mena de dubte dos de les assignatures més interessants que he cursat a la universitat, així que quan vaig veure la oferta del projecte vaig pensar que seria una bona idea per profunditzar els meus coneixements sobre la matèria.

## 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[1] 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[2] ( Scale Invariant Feature Transform) i SURF[3] (Speeded-Up Robust Features). Alguns d'aquests algorismes són BinBoost[4] o un dels més recents: LATCH[5]

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, zoomòrfics<sup>1</sup> 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.
- 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.

---

<sup>1</sup>**Robots zoomòrfics:** Robots que imiten característiques pròpies de determinats animals.

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.

## 1.5 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

## 1.6 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.).



## 1.7 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.

### **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.

### **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.

### **Error 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.

### **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.

## 1.8 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.

## 1.9 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[6] (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[7] 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.

## 1.10 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).

### 1.10.1 OpenCV

Per tal d'utilitzar algorismes de visió per computador en el codi amb relativa facilitat, s'utilitzarà la biblioteca de codi obert OpenCV[8] (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

## 1.11 Eines de seguiment

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

### 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.

### 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[9]. Aquesta eina ens permetrà realitzar tant un diagrama de Gantt com un diagrama de PERT.

### 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.

## 1.12 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. També es realitzarà una reunió de seguiment abans del 19 de desembre, per conèixer l'estat del projecte i poder escollir el torn de lectura.

## 2. Planificació i recursos

### 2.1 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 2.1: Blocs del projecte

#### 2.1.1 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<sup>A</sup>T<sub>E</sub>X i Zathura.

### 2.1.2 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 2.2: Lliurables de GEP

### 2.1.3 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 2.3: Tasques desenvolupament

## **Recerca d'informació, implementació i proves**

Una part molt important del projecte serà la cerca d'informació i l'estudi de les diverses eines i algorismes a utilitzar (com per exemple OpenCV i les seves funcions). Se cercarà informació contínuament i s'aniran fent proves a mesura que s'implementa el codi.

La fase d'implementació es dividirà en diverses tasques, que s'aniràn realitzant a mesura que avanci el projecte. Algunes d'aquestes tasques seràn:

- Obtenció de keypoints (Harris)
- Extracció de característiques (SIFT)
- Matching i homografia
- Altres algorismes (ORB, BRISK, BRIEF)
- Disseny de l'aplicació d'Android
- Creació de l'aplicació d'Android

## **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.

En cas de patir un retard en la planificació del projecte, s'utilitzarà aquest temps per acabar l'etapa de les conclusions.

## **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.

### 2.1.4 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.

### 2.1.5 Diagrames

Durant la fase de planificació del treball, s'han realitzat diversos diagrames. A continuació podeu trobar els diagrames de PERT i el Gantt del projecte.

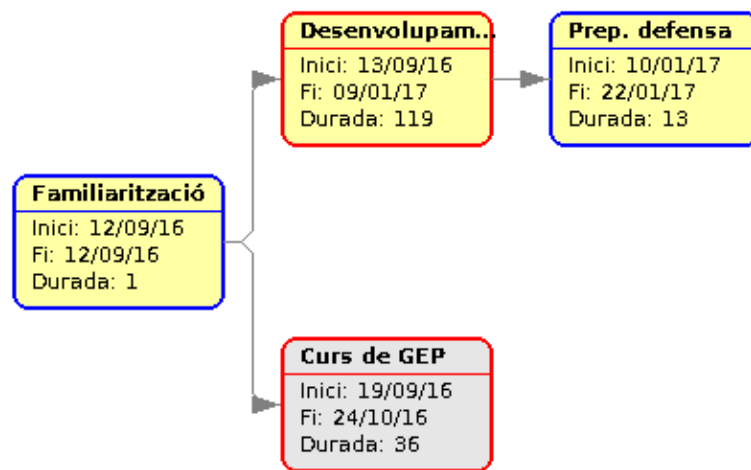


Figura 2.1: PERT del projecte

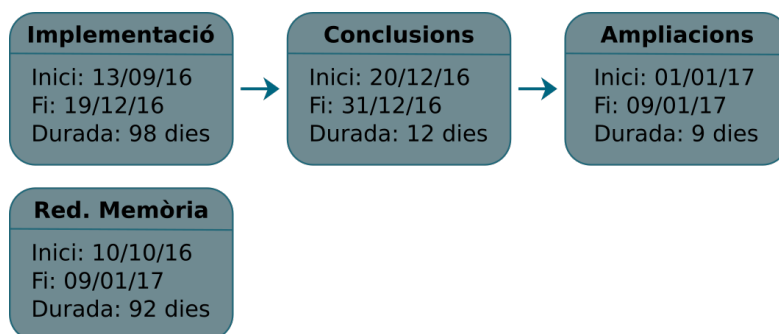


Figura 2.2: PERT - tasques desenvolupament



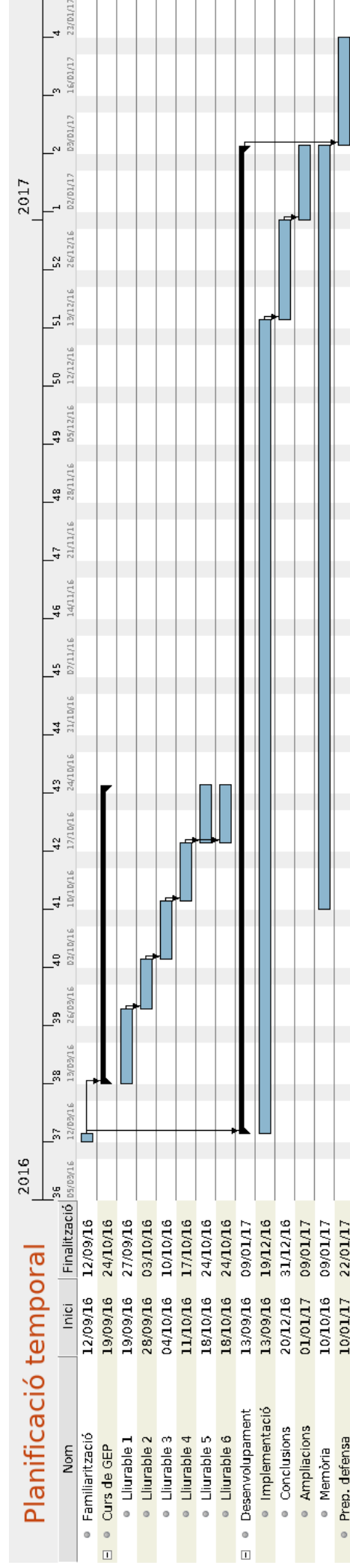


Figura 2.3: Gantt del projecte

## 2.2 Recursos

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

### 2.2.1 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.

### 2.2.2 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.

### 2.2.3 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
Android Studio	Eina de desenvolupament	Programació del codi
Gimp/Inkscape	Eina de desenvolupament	Retocs i creació d'imatges
L <sup>A</sup> T <sub>E</sub> 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 2.4: Recursos de programari

## 2.3 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.

## 3. Gestió econòmica

### 3.1 Recursos de programari

Tot el *software* utilitzat en aquest projecte és gratuït i de codi obert. Per tant, el programari no suposarà cap despesa. Podeu trobar el llistat del programari utilitzat a la taula 2.4 (Recursos de programari).

### 3.2 Recursos humans

El projecte el desenvoluparà una sola persona, que assumirà diversos rols durant la realització d'aquest. Tenint en compte les tasques descrites a la secció 2.1, les hores de treball queden repartides de la següent manera:

Tasca	Cap	Analista	Programador
Preparació de l'entorn	3h		2h
Curs de GEP	75h		
Implementació i proves		40h	200h
Conclusions i resultats		30h	
Ampliacions (opcional)			30h
Redacció memòria	40h		
Preparació defensa	30h		
Total	148h	70h	232h

Taula 3.1: Recursos humans (hores)

Suposem uns costos de 25€/h pel cap de projecte, 20€/h per l'analista i 15€/h pel programador/*tester*.

Rol	Hores	Cost/hora	Cost total
Cap de projecte	148h	25€/h	3700€
Analista	70h	20€/h	1400€
Programador	230h	15€/h	3450€
Total			8550€

Taula 3.2: Recursos humans (costos)

### 3.3 Recursos de maquinari

El *hardware* necessari per a la realització del treball serà només un ordinador (usat durant tot el projecte) i una càmera (per la fase d'implementació/proves).

Producte	Preu	Ús	Vida útil	Amortització
Ordinador personal	500€	5 mesos	5 anys	41,70€
Càmera	185€	1 mes	8 anys	1,93€
Total				43,60€

Taula 3.3: Recursos de maquinari (costos)

### 3.4 Costos indirectes

També es tindran en compte els costos indirectes més importants: la connexió a Internet i el consum elèctric. La connexió a Internet costarà 40€ al mes (considerem 240 hores) i l'electricitat 0,141033€/kWh (considerem la potència 0,2kW).

Tipus	Temps	Cost	Cost total
Electricitat	450h	0,028€/h	12,69€
Accès a Internet	450h	0,17€/h	75€
Total			87,69€

Taula 3.4: Costos indirectes

### 3.5 Imprevistos

Es podria donar el cas que el projecte ocupi més temps de l'esperat, pel que es considerarà un extra de 30 hores de treball, que es dividirien entre el programador i el *tester*. Això suposaria un increment de 600€ en el pressupost.

No es tindran en compte possibles fallades de maquinari, ja que l'ordinador principal amb què es treballa està en garantia i també es disposa d'altres ordinadors.

### 3.6 Contingència

Com a mesura de contingència, s'estableix un marge del 5%.

### 3.7 Costos totals

Tipus	Cost estimat
Recursos humans	8550€
Recursos de programari	0€
Recursos de maquinari	43,60€
Costos indirectes	87,69€
Imprevistos	600€
Contingència (5%)	464,06€
<b>Total</b>	<b>9745,35€</b>

Taula 3.5: Costos totals

### 3.8 Control de gestió

Després de cada tasca es farà una valoració del pressupost i es revisarà si és necessari. També es durà a terme un control de les hores de treball per cada rol mitjançant un full de càlcul, que s'anirà actualitzant cada dia de treball.

Es calcularà la desviació en mà d'obra, programari, maquinari i altres costos (cost estimat - cost real).

## 4. Informe de sostenibilitat

En aquest capítol es farà una anàlisi de la sostenibilitat del projecte, que es divideix en tres parts, identificades per les columnes de la matriu:

- El projecte posat en producció (PPP), que inclou la planificació, el desenvolupament i la implantació del projecte.
- La vida útil del projecte, que comença un cop implantat el sistema i finalitza amb el seu desmantellament.
- Els riscos inherents al propi projecte, considerant tota la construcció i la vida útil del mateix.

Cadascuna de les columnes s'analitzarà des dels punts de vista ambiental, econòmic i social, les tres dimensions de la sostenibilitat. A continuació podeu veure la matriu de sostenibilitat del projecte:

Sostenibilitat	PPP	Vida útil	Riscos
Ambiental	Consum del disseny <b>0</b> [0:10]	Petjada ecològica <b>0</b> [0:20]	Riscos ambientals <b>0</b> [-20:0]
Econòmica	Factura <b>0</b> [0:10]	Pla de viabilitat <b>0</b> [0:20]	Riscos econòmics <b>0</b> [-20:0]
Social	Impacte personal <b>0</b> [0:10]	Impacte social <b>0</b> [0:20]	Riscos socials <b>0</b> [-20:0]
Valoració total		<b>0</b> [-60:90]	

Taula 4.1: Matriu de sostenibilitat

## 4.1 Posada en producció

En aquesta secció es detalla la sostenibilitat del sistema desde la seva planificació fins a la possible implantació. Es tindrà en compte el consum del disseny, la factura i l'impacte personal que ha suposat la realització d'aquest treball.

### 4.1.1 Consum del disseny

Els recursos necessaris per al desenvolupament d'aquest projecte són mínims. No és necessari comprar cap tipus de maquinari addicional i tot el *hardware* utilitzat ha estat comprat a la unió europea, pagant una taxa pel correcte reciclatge dels residus. A més, el maquinari seguirà essent funcional un cop acabat el projecte. L'impacte ambiental del projecte serà mínim, ja que només es consumirà l'energia necessària per utilitzar un ordinador personal. No es generarà cap tipus de residu durant el desenvolupament o el desmantellament.

### 4.1.2 Factura

Tal com podeu veure a l'apartat "Gestió econòmica", per analitzar la viabilitat del projecte s'ha realitzat un pressupost tenint en compte els costos directes, indirectes i possibles imprevistos. Com es pot veure, els costos de *software* i *hardware* són mínims, pel que el projecte resulta econòmicament viable. L'única manera de reduir costos seria incrementant el nombre d'hores diàries (baixarien els costos de llum i Internet) o contractant a algú amb més experiència. No està prevista cap col·laboració amb altres projectes, però s'utilitzaran eines i algorismes existents que no caldrà programar de nou.

El cost inicial ha sigut aproximadament el mateix que el final, ja que no han hagut modificacions en els recursos de programari o maquinari.

### 4.1.3 Impacte personal

Aquest projecte m'ha permès profunditzar els meus coneixements sobre les tècniques de visió per computador actuals i la seva possible aplicació en sistemes robòtics.

Amb la realització d'aquest treball, també he hagut d'utilitzar una metodologia de treball àgil, que fins ara no havia utilitzat i he hagut de realitzar la planificació i la gestió dels recursos d'un projecte real, fet que estic segur que m'ajudarà en un futur a l'hora d'empendre altres projectes a nivell professional.



## 4.2 Vida útil

A continuació es descriu la sostenibilitat del projecte desde la implantació fins al desmantellament. Es tindrà en compte la petjada ecològica, la viabilitat i l'impacte del projecte en la societat.

### 4.2.1 Petjada ecològica

Es tracta d'un projecte de programari, que a més es publicarà sota una llicència de *software* lliure, de manera que qualsevol usuari se'n podrà beneficiar i podrà reutilitzar el codi per a futurs projectes. No suposaria un augment en la petjada ecològica ni tampoc una disminució, encara que amb la utilització del programari desenvolupat es necessitarien menys recursos materials pel control d'un robot.

### 4.2.2 Pla de viabilitat

En principi, no està prevista la implantació o la comercialització del sistema en un futur. El codi estarà disponible al repositori de Github i podrà ser utilitzat i adaptat lliurement. Per tant, seran els propis usuaris els que s'hauran de preocupar dels costos en cas d'utilitzar el sistema.

Si es volgués implantar el sistema, s'haurien de considerar els costos de la utilització i possible manteniment d'un servidor on allotjar el codi desenvolupat (ja sigui un servidor local o extern). I en cas de voler millorar o actualitzar el codi amb noves funcionalitats, s'haurien de tenir en compte els recursos humans necessaris.

### 4.2.3 Impacte social

El projecte no suposarà cap canvi important en la situació social o política del país ni té intenció de canviar substancialment la vida de les persones. Actualment, existeixen múltiples maneres de controlar un robot, ja sigui manualment o amb altres mètodes de localització, com podria ser utilitzant les coordenades GPS. També hi ha sistemes que utilitzen marques visuals (punts de referència) o que operen en un entorn conegut pel robot. El propòsit d'aquest projecte serà oferir una alternativa, un sistema d'autolocalització barat (no serà necessari dotar el robot de molts sensors) i disponible per a tothom.

Qualsevol usuari podrà beneficiar-se del sistema, ja que el codi serà publicat sota una llicència de *software* lliure en un repositori de Github. I evidentment, la realització d'aquest TFG no perjudicarà cap col·lectiu de cap manera.

## 4.3 Riscos

Finalment, s'analitzaran els riscos inherents del projecte en les tres dimensions de la sostenibilitat: ambientals, econòmics i socials.

### 4.3.1 Riscos ambientals

En principi la petjada ecològica del projecte té un marge molt limitat. Un cop desenvolupat el codi, s'allotjarà a Github. El que altres usuaris facin després ja no depen de la feina de l'autor.

### 4.3.2 Riscos econòmics

En cas d'implantar el sistema desenvolupat, l'únic risc econòmic serien les possibles fallades del sistema a nivell de servidor. Els costos del maquinari o del programari no podrien suposar cap problema, ja que el programari és gratuït i el sistema no depen d'un maquinari específic. Un cop implementat el codi, el sistema només depen d'un servidor que pugui executar Python i OpenCV.

### 4.3.3 Riscos socials

Aquest treball no perjudicarà en cap cas a algun sector de la població, ni en la posada en producció, ni durant la possible implantació o posterior desmantellament. El sistema desenvolupat és un programari inofensiu, que només podria ser perjudicial si algun usuari en fes un mal ús en un servidor extern, fet que ja no seria responsabilitat de l'autor.

La única dependència del programa principal és la biblioteca OpenCV, però tractant-se d'una biblioteca de *software* lliure no hauria de suposar cap problema. S'utilitzen els algorismes SIFT i SURF, que estan patentats i per tant no es poden utilitzar gratuïtament en productes comercials. Tot i que no està previst comercialitzar el sistema desenvolupat de cap manera, també s'utilitzen algorismes alternatius com ORB que no suposarien cap problema si algun usuari volgués comercialitzar un producte derivat d'aquest projecte.

# Bibliografia

- [1] Richard Szeliski. “Computer Vision: Algorithms and Applications”. 2010. URL: [http://szeliski.org/Book/drafts/SzeliskiBook\\_20100903\\_draft.pdf](http://szeliski.org/Book/drafts/SzeliskiBook_20100903_draft.pdf).
- [2] David G. Lowe. “Object recognition from local scale-invariant features”. A: *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*. Vol. 2. 1999, pàg. 1150-1157. DOI: 10.1109/ICCV.1999.790410. URL: <http://www.cs.ubc.ca/~lowe/papers/iccv99.pdf>.
- [3] Herbert Bay et al. “Speeded-Up Robust Features (SURF)”. A: *Comput. Vis. Image Underst.* 110.3 (juny de 2008), pàg. 346-359. ISSN: 1077-3142. DOI: 10.1016/j.cviu.2007.09.014. URL: <http://dx.doi.org/10.1016/j.cviu.2007.09.014>.
- [4] V. Lepetit T. Trzcinski M. Christoudias i P. Fua. “Boosting Binary Key-point Descriptors”. A: *Computer Vision and Pattern Recognition*. 2013. URL: <https://infoscience.epfl.ch/record/186246/files/top.pdf>.
- [5] Gil Levi i Tal Hassner. “LATCH: Learned Arrangements of Three Patch Codes”. A: *Winter Conference on Applications of Computer Vision (WACV)*. IEEE. 2016. URL: <http://www.openu.ac.il/home/hassner/projects/LATCH>.
- [6] Yani Dzhurov, Iva Krasteva i Sylvia Ilieva. “Personal Extreme Programming—An Agile Process for Autonomous Developers”. A: (2009). URL: <https://www.researchgate.net/publication/229046039>.
- [7] Chris Harris i Mike Stephens. “A combined corner and edge detector”. A: *In Proc. of Fourth Alvey Vision Conference*. 1988, pàg. 147-151. URL: [www.bmva.org/bmvc/1988/avc-88-023.pdf](http://www.bmva.org/bmvc/1988/avc-88-023.pdf).
- [8] Intel Corporation, Willow Garage i Itseez. *OpenCV*. 2000. URL: <http://opencv.org>.
- [9] Alexandre Thomas i Dmitry Barashev. *Gantt Project*. 2003. URL: <http://www.ganttproject.biz/>.

# Índex de taules

2.1	Blocs del projecte . . . . .	13
2.2	Lliurables de GEP . . . . .	14
2.3	Tasques desenvolupament . . . . .	14
2.4	Recursos de programari . . . . .	19
3.1	Recursos humans (hores) . . . . .	20
3.2	Recursos humans (costos) . . . . .	21
3.3	Recursos de maquinari (costos) . . . . .	21
3.4	Costos indirectes . . . . .	21
3.5	Costos totals . . . . .	22
4.1	Matriu de sostenibilitat . . . . .	23

# Índex de figures

2.1	PERT del projecte . . . . .	16
2.2	PERT - tasques desenvolupament . . . . .	16
2.3	Gantt del projecte . . . . .	17